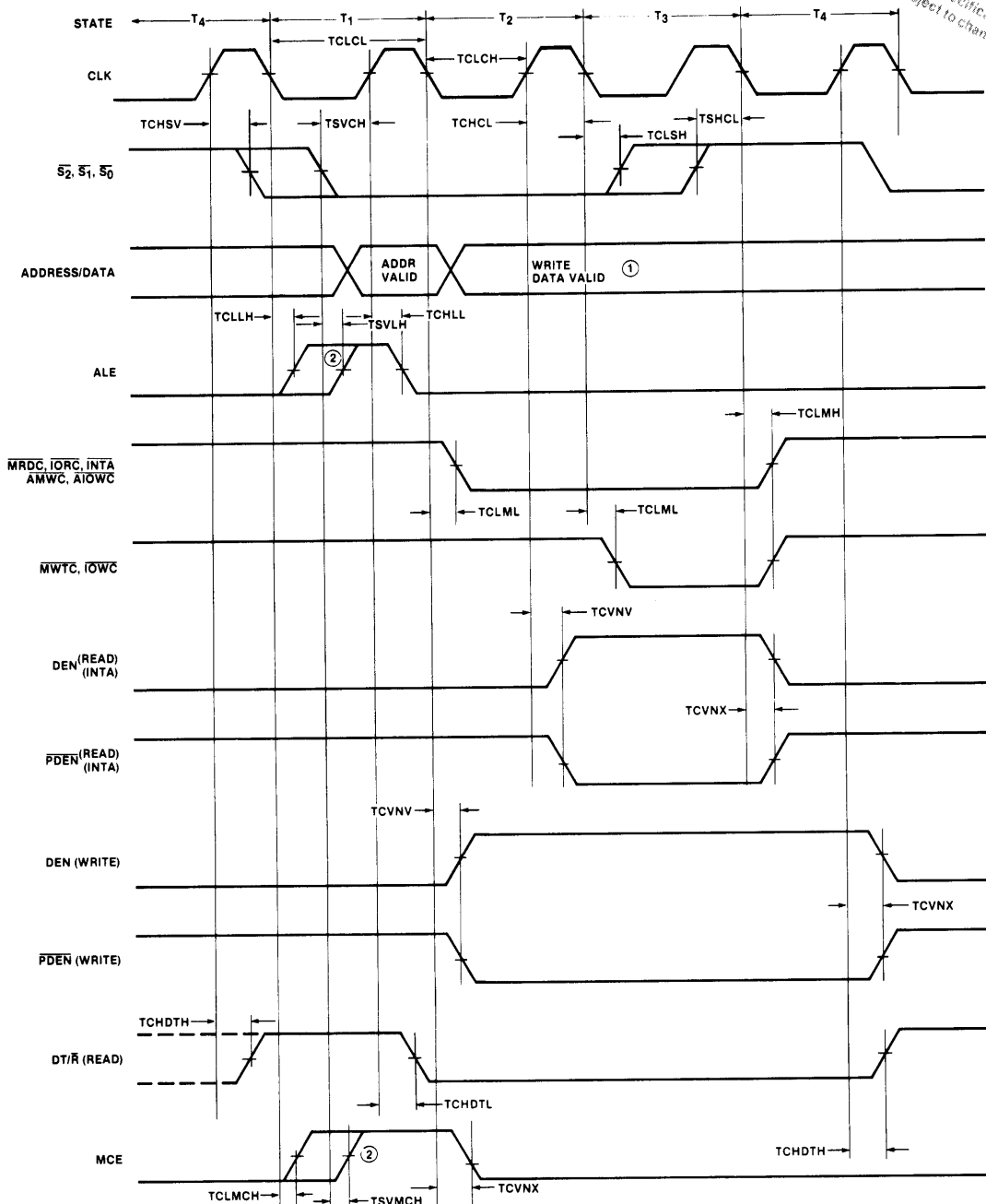


8288 TIMING DIAGRAM

PRELIMINARY
 Note: This is not a final specification. Some parametric limits are subject to change.



- NOTES:**
1. ADDRESS/DATA BUS IS SHOWN ONLY FOR REFERENCE PURPOSES.
 2. LEADING EDGE OF ALE AND MCE IS DETERMINED BY THE FALLING EDGE OF CLK OR STATUS GOING ACTIVE, WHICHEVER OCCURS LAST.
 3. ALL TIMING MEASUREMENTS ARE MADE AT 1.5V UNLESS SPECIFIED OTHERWISE.

8289 BUS ARBITER

- Provides Multi-Master System Bus Protocol
- Synchronizes 8086/8088 Processors With Multi-Master Bus
- Provides Simple Interface With 8288 Bus Controller
- Four Operating Modes For Flexible System Configuration
- Compatible with Intel Bus Standard MULTIBUS™
- Provides System Bus Arbitration For 8089 IOP In Remote Mode

The Intel 8289 Bus Arbiter is a 20-pin, 5-volt-only bipolar component for use with medium to large 8086/8088 multi-master/multiprocessing systems. The 8289 provides system bus arbitration for systems with multiple bus masters, such as an 8086 CPU with 8089 IOP in its REMOTE mode, while providing bipolar buffering and drive capability.

BLOCK DIAGRAM

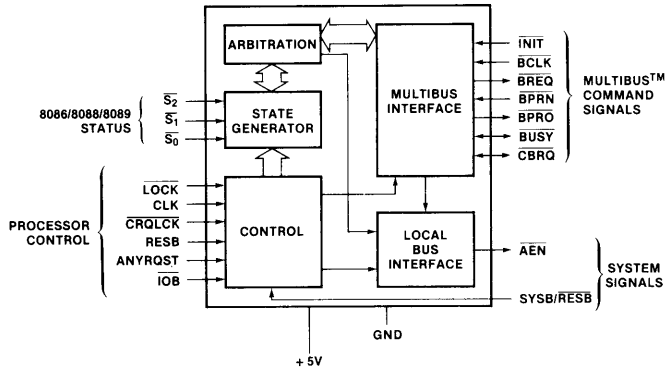


Figure 1. Block Diagram.

PIN DIAGRAM

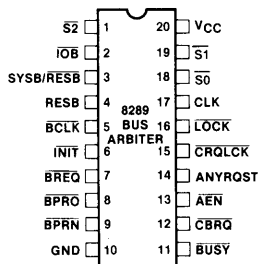


Figure 2. Pin Diagram.

FUNCTIONAL PINOUT

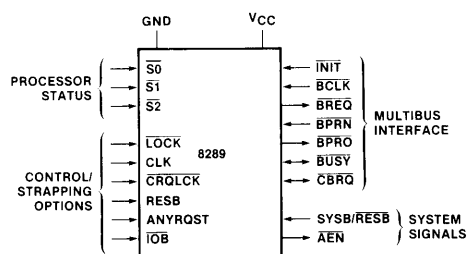


Figure 3. Functional Pinout.

FUNCTIONAL DESCRIPTION

The 8289 Bus Arbiter operates in conjunction with the 8288 Bus Controller to interface 8086/8088/8089 processors to a multi-master system bus (both the 8086 and 8088 are configured in their max mode). The processor is unaware of the arbiter's existence and issues commands as though it has exclusive use of the system bus. If the processor does not have the use of the multi-master system bus, the arbiter prevents the Bus Controller (8288), the data transceivers and the address latches from accessing the system bus (e.g. all bus driver outputs are forced into the high impedance state). Since the command sequence was not issued by the 8288, the system bus will appear as "Not Ready" and the processor will enter wait states. The processor will remain in Wait until the Bus Arbiter acquires the use of the multi-master system bus whereupon the arbiter will allow the bus controller, the data transceivers, and the address latches to access the system. Typically, once the command has been issued and a data transfer has taken place, a transfer acknowledge (XACK) is returned to the processor to indicate "READY" from the accessed slave device. The processor then completes its transfer cycle. Thus the arbiter serves to multiplex a processor (or bus master) onto a multi-master system bus and avoid contention problems between bus masters.

ARBITRATION BETWEEN BUS MASTERS

In general, higher priority masters obtain the bus when a lower priority master completes its present transfer cycle. Lower priority bus masters obtain the bus when a higher priority master is not accessing the system bus. A strapping option (ANYRQST) is provided to allow the arbiter to surrender the bus to a lower priority master as though it were a master of higher priority. If there are no other bus masters requesting the bus, the arbiter maintains the bus so long as its processor has not entered the HALT State. The arbiter will not voluntarily surrender the system bus and has to be forced off by another master's bus request, the HALT State being the only ex-

ception. Additional strapping options permit other modes of operation wherein the multi-master system bus is surrendered or requested under different sets of conditions.

PRIORITY RESOLVING TECHNIQUES

Since there can be many bus masters on a multi-master system bus, some means of resolving priority between bus masters simultaneously requesting the bus must be provided. The 8289 Bus Arbiter provides several resolving techniques. All the techniques are based on a priority concept that at a given time one bus master will have priority above all the rest. There are provisions for using parallel priority resolving techniques, serial priority resolving techniques, and rotating priority techniques.

Parallel Priority Resolving

The parallel priority resolving technique uses a separate bus request line (\overline{BREQ}) for each arbiter on the multi-master system bus, see Figure 4. Each \overline{BREQ} line enters into a priority encoder which generates the binary address of the highest priority \overline{BREQ} line which is active. The binary address is decoded by a decoder to select the corresponding \overline{BPRN} (Bus Priority In) line to be returned to the highest priority requesting arbiter. The arbiter receiving priority (\overline{BPRN} true) then allows its associated bus master onto the multi-master system bus as soon as it becomes available (i.e., the bus is no longer busy). When one bus arbiter gains priority over another arbiter it cannot immediately seize the bus, it must wait until the present bus transaction is complete. Upon completing its transaction the present bus occupant recognizes that it no longer has priority and surrenders the bus by releasing \overline{BUSY} . \overline{BUSY} is an active low "OR" tied signal line which goes to every bus arbiter on the system bus. When \overline{BUSY} goes inactive (high), the arbiter which presently has bus priority (\overline{BPRN} true) then seizes the bus and pulls \overline{BUSY} low to keep other arbiters off of the bus. See waveform timing diagram, Figure 5.

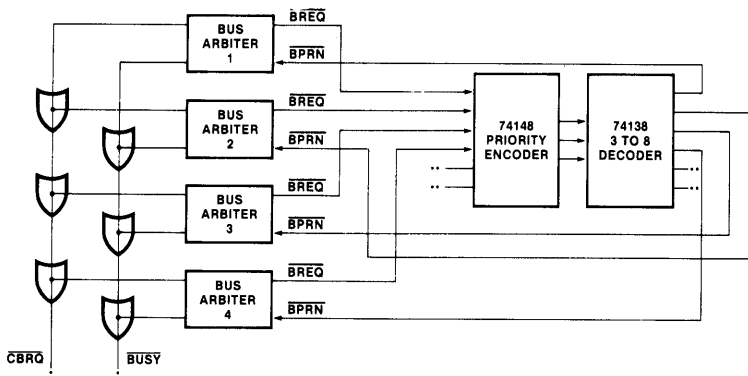
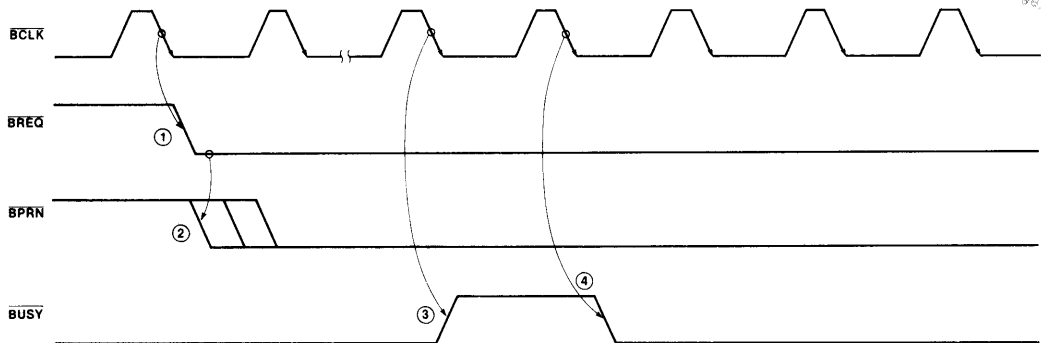


Figure 4. Parallel Priority Resolving Technique.



- ① HIGHER PRIORITY BUS ARBITER REQUESTS THE MULTI-MASTER SYSTEM BUS.
- ② ATTAINS PRIORITY.
- ③ LOWER PRIORITY BUS ARBITER RELEASES BUSY.
- ④ HIGHER PRIORITY BUS ARBITER THEN ACQUIRES THE BUS AND PULLS BUSY DOWN.

Figure 5. Higher Priority Arbitrer obtaining the Bus from a Lower Priority Arbitrer.

Note that all multi-master system bus transactions are synchronized to the bus clock ($\overline{\text{BCLK}}$). This allows the parallel priority resolving circuitry or any other priority resolving scheme employed to settle.

Serial Priority Resolving

The serial priority resolving technique eliminates the need for the priority encoder-decoder arrangement by daisy-chaining the bus arbiters together, connecting the higher priority bus arbiter's $\overline{\text{BPRO}}$ (Bus Priority Out) output to the $\overline{\text{BPRN}}$ of the next lower priority. See Figure 6.

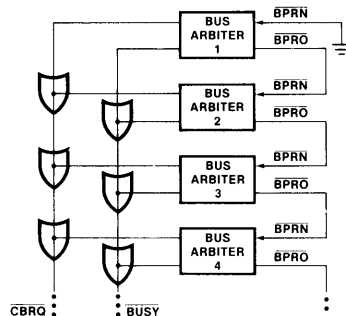
Rotating Priority Resolving

The rotating priority resolving technique is similar to that of the parallel priority resolving technique except that priority is dynamically re-assigned. The priority encoder is replaced by a more complex circuit which rotates priority between requesting arbiters thus allowing each arbiter an equal chance to use the multi-master system bus, over time.

WHICH PRIORITY RESOLVING TECHNIQUE TO USE

There are advantages and disadvantages for each of the techniques described above. The rotating priority resolving technique requires substantial external logic to implement while the serial technique uses no external logic but can accommodate only a limited number of bus arbiters before the daisy-chain propagation delay

exceeds the multi-master's system bus clock ($\overline{\text{BCLK}}$). The parallel priority resolving technique is in general a good compromise between the other two techniques. It allows for many arbiters to be present on the bus while not requiring too much logic to implement.



THE NUMBER OF ARBITERS THAT MAY BE DAISY-CHAINED TOGETHER IN THE SERIAL PRIORITY RESOLVING SCHEME IS A FUNCTION OF $\overline{\text{BCLK}}$ AND THE PROPAGATION DELAY FROM ARBITER TO ARBITER. NORMALLY, AT 10 MHz ONLY 3 ARBITER MAY BE DAISY-CHAINED.

Figure 6. Serial Priority Resolving.

8289 MODES OF OPERATION

There are two types of processors in the 8086 family. An Input/Output processor (the 8089 IOP) and the 8086/8088 CPUs. Consequently, there are two basic operating modes in the 8289 bus arbiter. One, the \overline{IOB} (I/O Peripheral Bus) mode, permits the processor access to both an I/O Peripheral Bus and a multi-master system bus. The second, the RESB (Resident Bus mode), permits the processor to communicate over both a Resident Bus and a multi-master system bus. An I/O Peripheral Bus is a bus where all devices on that bus, including memory, are treated as I/O devices and are addressed by I/O commands. All memory commands are directed to another bus, the multi-master system bus. A Resident Bus can issue both memory and I/O commands, but it is a distinct and separate bus from the multi-master system bus. The distinction is that the Resident Bus has only one master, providing full availability and being dedicated to that one master.

The \overline{IOB} strapping option configures the 8289 Bus Arbiter into the \overline{IOB} mode and the strapping option RESB configures it into the RESB mode. It might be noted at this point that if both strapping options are strapped false, the arbiter interfaces the processor to a multi-master system bus only (see Figure 7). With both options strapped true, the arbiter interfaces the processor

to a multi-master system bus, a Resident Bus, and an I/O Bus.

In the \overline{IOB} mode, the processor communicates and controls a host of peripherals over the Peripheral Bus. When the I/O Processor needs to communicate with system memory, it does so over the system memory bus. Figure 8 shows a possible I/O Processor system configuration.

The 8086 and 8088 processor can communicate with a Resident Bus and a multi-master system bus. Two bus controllers and only one Bus Arbiter would be needed in such a configuration as shown in Figure 9. In such a system configuration the processor would have access to memory and peripherals of both busses. Memory mapping techniques are applied to select which bus is to be accessed. The SYSB/ \overline{RESB} input on the arbiter serves to instruct the arbiter as to whether or not the system bus is to be accessed. The signal connected to SYSB/ \overline{RESB} also enables or disables commands from one of the bus controllers.

A summary of the modes that the 8289 has, along with its response to its status lines inputs, is summarized in Table 1.

*In some system configurations it is possible for a non-I/O Processor to have access to more than one Multi-Master System Bus, see 8289 Application Note.

	Status Lines From 8086 or 8088 or 8089			<u>IOB Mode</u> Only	<u>RESB (Mode) Only</u> $\overline{IOB} = \text{High}$ $\overline{RESB} = \text{High}$		<u>IOB Mode RESB Mode</u> $\overline{IOB} = \text{Low}$ $\overline{RESB} = \text{High}$		<u>Single Bus Mode</u> $\overline{IOB} = \text{High}$ $\overline{RESB} = \text{Low}$
	S2	S1	S0	$\overline{IOB} = \text{Low}$	SYSB/ $\overline{RESB} = \text{High}$	SYSB/ $\overline{RESB} = \text{Low}$	SYSB/ $\overline{RESB} = \text{High}$	SYSB/ $\overline{RESB} = \text{Low}$	
	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
MEM 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 = Multi-Master System Bus is allowed to be Surrendered.
2. ✓ = Multi-Master System Bus is Requested.

Mode	Pin Strapping	Multi-Master System Bus	
		Requested**	Surrendered*
Single Bus Multi-Master Mode	$\overline{IOB} = \text{High}$ $\overline{RESB} = \text{Low}$	Whenever the processor's status lines go active	HLT + TI • CBRQ + HPBRQ [†]
RESB Mode Only	$\overline{IOB} = \text{High}$ $\overline{RESB} = \text{High}$	SYSB/ $\overline{RESB} = \text{High}$ • ACTIVE STATUS	(SYSB/ $\overline{RESB} = \text{Low} + \text{TI}$) • CBRQ + HLT + HPBRQ
IOB Mode Only	$\overline{IOB} = \text{Low}$ $\overline{RESB} = \text{Low}$	Memory Commands	(I/O Status + TI) • CBRQ + HLT + HPBRQ
IOB Mode RESB Mode	$\overline{IOB} = \text{Low}$ $\overline{RESB} = \text{High}$	(Memory Command) • (SYSB/ $\overline{RESB} = \text{High}$)	((I/O Status Commands) + SYSB/ $\overline{RESB} = \text{Low}$) • CBRQ + HPBRQ [†] + HLT

NOTES:

*LOCK prevents surrender of Bus to any other arbiter, \overline{CRQLCK} prevents surrender of Bus to any lower priority arbiter.

**Except for HALT and Passive or IDLE Status.

[†]HPBRQ, Higher priority Bus request or BPRN = 1.

1. IOB Active Low.

2. RESB Active High.

3. + is read as "OR" and • as "AND."

4. TI = Processor Idle Status $\overline{S2}$, $\overline{S1}$, $\overline{S0} = 111$

5. HLT = Processor Halt Status $\overline{S2}$, $\overline{S1}$, $\overline{S0} = 011$

Table 1. Summary of 8289 Modes, Requesting and Relinquishing the Multi-master system bus.

PRELIMINARY
 Notice: This is not a final specification. Some parametric limits are subject to change.

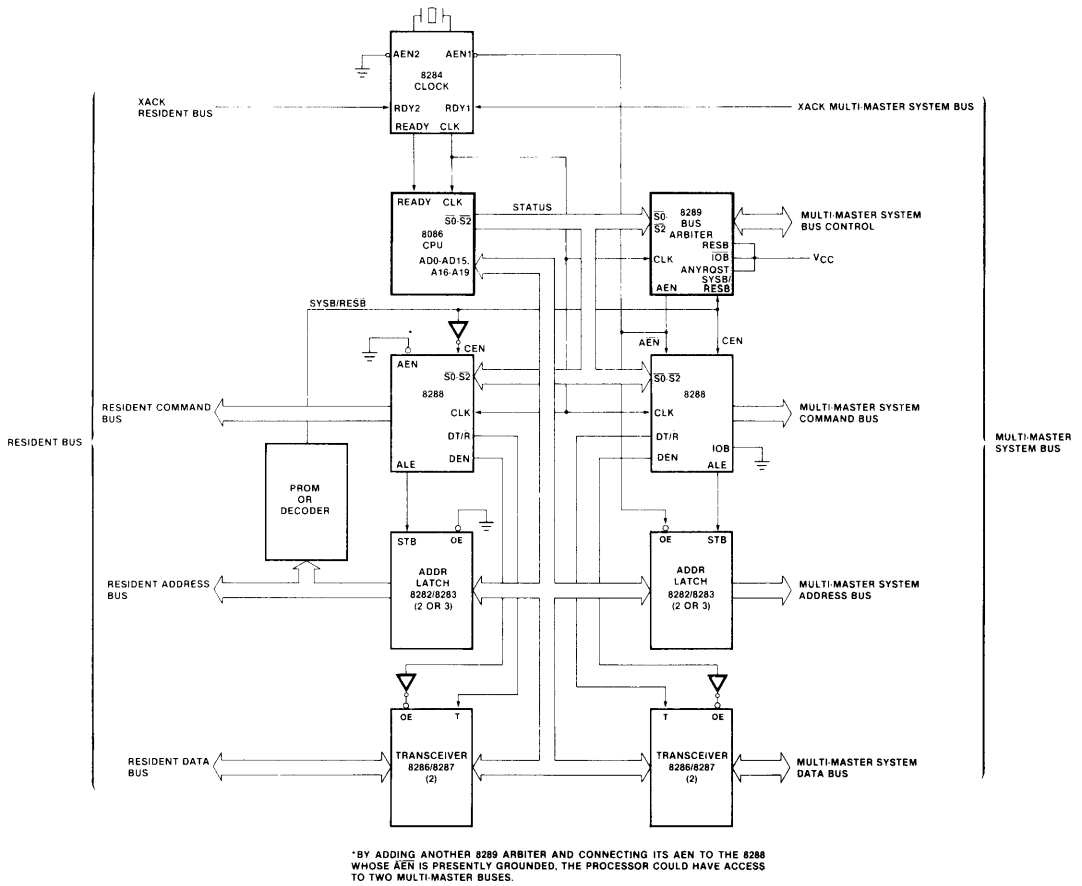


Figure 9. 8289 Bus Arbiter Shown in System-Resident Bus Configuration.

PRELIMINARY
 Notice: This is not a final specification. Some
 parametric limits are subject to change.

PIN DEFINITIONS

Name	I/O	Function	Name	I/O	Function
V _{CC}		+5V supply $\pm 10\%$			
GND		Ground			
$\overline{S0}, \overline{S1}, \overline{S2}$	I	STATUS INPUT PINS: These pins are the status input pins from an 8086, 8088 or 8089 processor. The 8289 decodes these pins to initiate bus request and surrender actions. (See Table 1)			ming IO commands and is requested whenever the processor performs a memory command. Interrupt cycles are assumed as coming from the peripheral bus and are treated as would be an IO command.
CLK	I	CLOCK: This is the clock from the 8284 clock chip and serves to establish when bus arbiter actions are initiated.	\overline{AEN}	O	ADDRESS ENABLE. \overline{AEN} is the output of the 8289 Arbiter to the processor's address latches, to the 8288 Bus Controller and 8284 Clock Generator. \overline{AEN} serves to instruct the Bus Controller and address latches when to tri-state their output drivers.
\overline{LOCK}	I	LOCK: \overline{LOCK} is a processor generated signal which when activated (low) serves to prevent the arbiter from surrendering the multi-master system bus to any other bus arbiter, regardless of its priority.	$\overline{SYSB}/\overline{RESB}$	I	SYSTEM BUS/RESIDENT BUS: $\overline{SYSB}/\overline{RESB}$ is an input signal when the arbiter is configured in the S.R. Mode (RESB is strapped high) which serves to determine when the multi-master system bus is requested and when the multi-master system bus surrendering is permitted. The signal is intended to originate from some form of address mapping circuitry such as a decoder or PROM attached to the resident address bus. Signal transitions and glitches are permitted on this pin from $\emptyset 1$ of T4 to $\emptyset 1$ of T2 of the processor cycle. During the period from $\emptyset 1$ of T2 to $\emptyset 1$ of T4 only clean transitions are permitted on this pin (no glitches). If a glitch does occur the arbiter may capture or miss it, and the multi-master system bus may be requested or surrendered, depending upon the state of the glitch. The arbiter requests the multi-master system bus in the S.R. Mode when the state of the $\overline{SYSB}/\overline{RESB}$ pin is high and permits the bus to be surrendered when this pin is low.
\overline{CRQLCK}	I	COMMON REQUEST LOCK: \overline{CRQLCK} is an active low signal which serves to prevent the arbiter from surrendering the multi-master system bus to any other bus arbiter requesting the bus through the \overline{CBRQ} input pin.			
RESB	I	RESB: RESIDENT BUS is a strapping option to configure the arbiter to operate in systems having both a multi-master system bus and a Resident Bus. When it is strapped high the multi-master system bus is requested or surrendered as a function of the $\overline{SYSB}/\overline{RESB}$ input pin. When it is strapped low the $\overline{SYSB}/\overline{RESB}$ input is ignored.			
ANYRQST	I	ANY REQUEST: ANYRQST is a strapping option which permits the multi-master system bus to be surrendered to a lower priority arbiter as though it were an arbiter of higher priority (i.e., when a lower priority arbiter requests the use of the multi-master system bus, the bus is surrendered as soon as it is possible). Strapping \overline{CBRQ} low and ANYRQST high forces the 8289 arbiter to surrender the multi-master system bus after each transfer cycle. Note that when surrender occurs BREQ is driven false (high).	\overline{CBRQ}	I/O	COMMON BUS REQUEST: \overline{CBRQ} is an input signal which serves to instruct the arbiter if there are any other arbiters of lower priority requesting the use of the multi-master system bus. The \overline{CBRQ} pins (open-collector output) of all the 8289 Bus Arbiters which are to surrender the multi-master system bus upon request are connected together. The Bus Arbiter running the current transfer cycle will not itself pull the \overline{CBRQ} line low. Any other arbiter connected to the \overline{CBRQ} line can request the multi-master system bus. The arbiter presently running the current transfer cycle drops its BREQ signal and surrenders the bus whenever the
\overline{IOB}	I	IO BUS: \overline{IOB} is a strapping option which configures the 8289 Arbiter to operate in systems having both an IO Bus (Peripheral Bus) and a multi-master system bus. The arbiter requests and surrenders the use of the multi-master system bus as a function of the status line, S2. The multi-master system bus is permitted to be surrendered while the processor is perform-			

PRELIMINARY
 Notice: This is not a final specification. Some parametric limits are subject to change.

PIN DEFINITIONS (Cont'd)

Name	I/O	Function	Name	I/O	Function
		proper <u>surrender</u> conditions exist. Strapping <u>CBREQ</u> low and <u>ANYRQST</u> high allows the multi-master system bus to be surrendered after each transfer cycle. See the pin definition of <u>ANYRQST</u> .			edge of <u>BCLK</u> . <u>BPRN</u> indicates to the arbiter that it is the highest priority requesting arbiter presently on the bus. The loss of <u>BPRN</u> instructs the arbiter that it has loss priority to a higher priority arbiter.
<u>INIT</u>	I	INITIALIZE: <u>INIT</u> is an active low multi-master system bus input signal which is used to reset all the bus arbiters on the multi-master system bus. After initialization, no arbiters have the use of the multi-master system bus.	<u>BPRO</u>	O	BUS PRIORITY OUT: <u>BPRO</u> is an active low output signal which is used in the serial priority resolving scheme where <u>BPRO</u> is daisy chained to <u>BPRN</u> of the next lower priority arbiter.
<u>BCLK</u>	I	BUS CLOCK: <u>BCLK</u> is the multi-master system bus clock to which all multi-master system bus interface signals are synchronized.	<u>BUSY</u>	I/O	BUSY: <u>BUSY</u> is an active low open collector multi-master system bus interface signal which is used to instruct all the arbiters on the bus when the multi-master system bus is available. When the multi-master system bus is available the highest requesting arbiter (determined by <u>BPRN</u>) seizes the bus and pulls <u>BUSY</u> low to keep other arbiters off of the bus. When the arbiter is done with the bus it releases the <u>BUSY</u> signal permitting it to go high and thereby allowing another arbiter to acquire the multi-master system bus.
<u>BREQ</u>	O	BUS REQUEST: <u>BREQ</u> is an active low output signal in the parallel Priority Resolving Scheme which the arbiter activates to request the use of the multi-master system bus.			
<u>BPRN</u>	I	BUS PRIORITY IN: <u>BPRN</u> is the active low signal returned to the arbiter to instruct it that it may acquire the multi-master system bus on the next falling			

PRELIMINARY
 Notice: This is not a final specification. Some parametric limits may be subject to change.

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias	0°C to 70°C
Storage Temperature	- 65°C to + 150°C
All Output and Supply Voltages	- 0.5V to + 7V
All Input Voltages	- 1.0V to + 5.5V
Power Dissipation	1.5 Watt

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.

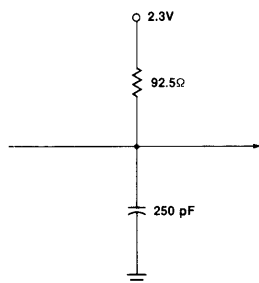
D.C. CHARACTERISTICS FOR THE 8289

CONDITIONS: $T_A = 0^\circ$ to 70°C , $V_{CC} = 5\text{V} \pm 10\%$

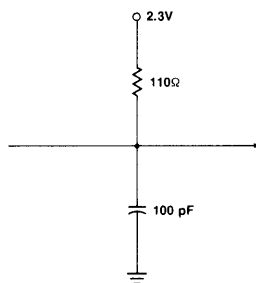
Symbol	Parameter	Min.	Max.	Units	Test Condition
V_C	Input Clamp Voltage		- 1.0	V	$V_{CC} = 4.50\text{V}$, $I_C = - 5\text{ mA}$
I_F	Input Forward Current		- 0.5	mA	$V_{CC} = 5.50\text{V}$, $V_F = 0.45\text{V}$
I_R	Reverse Input Leakage Current		60	μA	$V_{CC} = 5.50$, $V_R = 5.50$
V_{OL}	Output Low Voltage		0.45	V	$I_{OL} = 20\text{ mA}$, $C_L = 250\text{ pF}$ 1)
	$\overline{\text{BUSY}}$, $\overline{\text{CBRQ}}$		0.45	V	$I_{OL} = 16\text{ mA}$, $C_L = 100\text{ pF}$ 2)
	$\overline{\text{AEN}}$ $\overline{\text{BPRO}}$, $\overline{\text{BREQ}}$		0.45	V	$I_{OL} = 10\text{ mA}$, $C_L = 60\text{ pF}$ 3)
V_{OH}	Output High Voltage	Open Collector			
	$\overline{\text{BUSY}}$, $\overline{\text{CBRQ}}$				
	All Other Outputs	2.4		V	$I_{OH} = 400\ \mu\text{A}$
I_{CC}	Power Supply Current		165	mA	
V_{IL}	Input Low Voltage		.8	V	
V_{IH}	Input High Voltage	2.0		V	
C_{in} Status	Input Capacitance		25	pF	
C_{in} (Others)	Input Capacitance		12	pF	

TEST CIRCUITS:

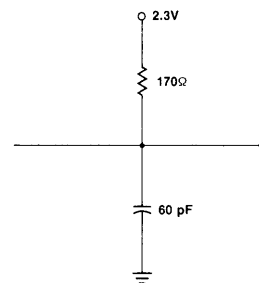
1) $\overline{\text{BUSY}}$, $\overline{\text{CBRQ}}$



2) $\overline{\text{AEN}}$



3) $\overline{\text{BPRO}}$, $\overline{\text{BREQ}}$



A.C. CHARACTERISTICS FOR THE 8289

CONDITIONS: $V_{CC} = 5V \pm 10\%$, $T_A = 0^\circ C$ to $70^\circ C$

Timing Requirements

Symbol	Parameter	Min.	Max.	Unit
TCLCL	CLK Cycle Period	125		ns
TCLCH	CLK Low Time	65		ns
TCHCL	CLK High Time	35		ns
TSVCH	Status Active Setup	65	TCLCL-10	ns
TSHCL	Status Inactive Setup	50	TCLCL-10	ns
THVCH	Status Active Hold	10		ns
THVCL	Status Inactive Hold	10		ns
TBYSBL	$\overline{BUSY}\uparrow$ Setup to $\overline{BCLK}\downarrow$	20		ns
TCBSBL	$\overline{CBRQ}\uparrow$ Setup to $\overline{BCLK}\downarrow$	20		ns
TBLBL	\overline{BCLK} Cycle Time	100		ns
TBHCL	\overline{BCLK} High Time	30	.65[TBLBL]	ns
TCLL1	\overline{LOCK} Inactive Hold	20		ns
TCLL2	\overline{LOCK} Active Setup	40		ns
TPNBL	$\overline{BPRN}\uparrow$ to \overline{BCLK} Setup Time	15		ns
TCLSR1	SYSB/ \overline{RESB} Setup	0		ns
TCLSR2	SYSB/ \overline{RESB} Hold	20		ns
TIVIH	Initialization Pulse Width	3 TBLBL + 3 TCLCL		ns

Timing Responses

Symbol	Parameter	Min.	Max.	Unit	Loading
TBLBRL	\overline{BCLK} to \overline{BREQ} Delay \uparrow		35	ns	
TBLPOH	\overline{BCLK} to $\overline{BPRO}\uparrow$ (See Note 1)		40	ns	
TPNPO	$\overline{BPRN}\uparrow$ to $\overline{BPRO}\uparrow$ Delay (See Note 1)		25	ns	
TBLBYL	\overline{BCLK} to \overline{BUSY} Low		60	ns	
TBLBYH	\overline{BCLK} to \overline{BUSY} Float (See Note 2)		35	ns	
TCLAEH	CLK to \overline{AEN} High		65	ns	
TBLAEL	\overline{BCLK} to \overline{AEN} Low		40	ns	
TBLCBL	\overline{BCLK} to \overline{CBRQ} Low		60	ns	
TBLCBH	\overline{BCLK} to \overline{CBRQ} Float (See Note 2)		35	ns	

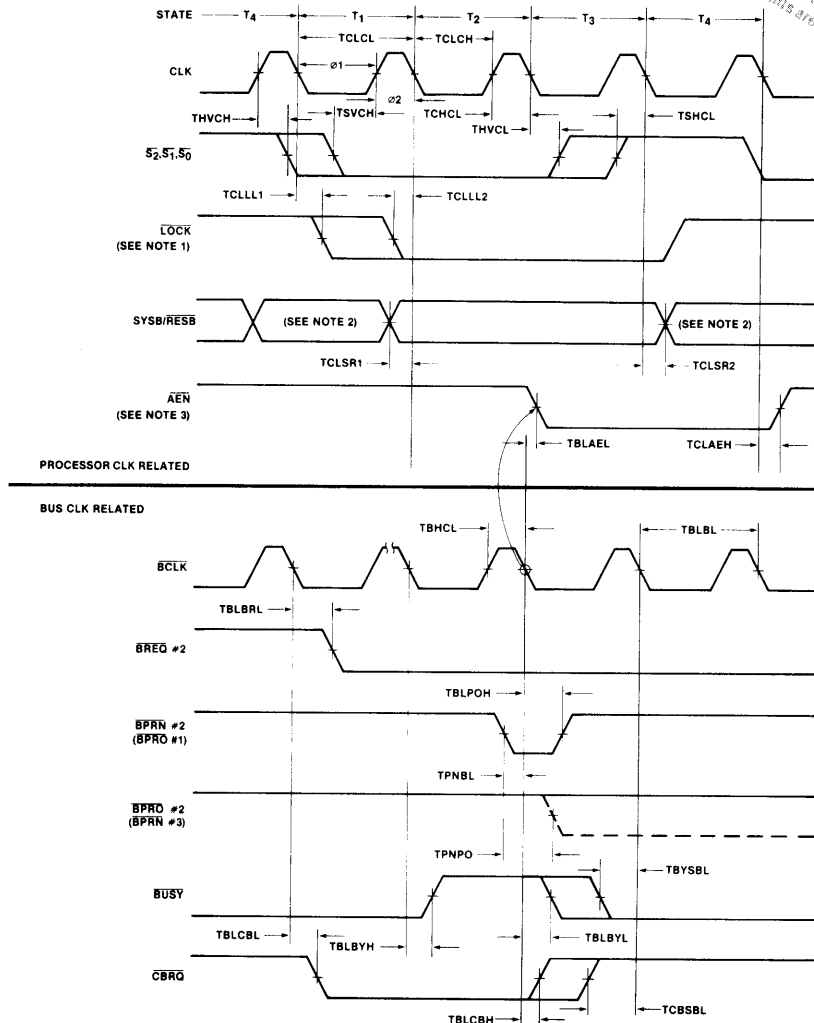
 \uparrow Denotes that spec applies to both transitions of the signal.NOTE 1: \overline{BCLK} generates the first \overline{BPRO} wherein subsequent \overline{BPRO} changes lower in the chain are generated through \overline{BPRN} .

NOTE 2: Measured at .5V above GND.

INITIALIZATION: (INIT can be either pulsed or held low through power up)

8289 TIMING DIAGRAM

PRELIMINARY
 Note: This is not a final specification. Some parametric limits are subject to change.



NOTES:

1. LOCK ACTIVE CAN OCCUR DURING ANY T STATE, AS LONG AS THE RELATIONSHIPS SHOWN ABOVE WITH RESPECT TO THE CLK ARE MAINTAINED. LOCK INACTIVE HAS NO CRITICAL TIME AND CAN BE ASYNCHRONOUS.
 -CLOCK HAS NO CRITICAL TIMING AND IS CONSIDERED AN ASYNCHRONOUS INPUT SIGNAL.
2. GLITCHING OF SYSB/RESB PIN IS PERMITTED DURING THIS TIME, AFTER $\phi 2$ OF T1, AND BEFORE $\phi 1$ OF T4. ONLY CLEAN TRANSITIONS ARE ACCEPTED.
3. AEN LEADING EDGE IS RELATED TO BCLK, TRAILING EDGE TO CLK. THE TRAILING EDGE OF AEN OCCURS AFTER BUS PRIORITY IS LOST.

ADDITIONAL NOTES:

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 3 bus arbiters of priorities 1, 2 and 3 configured in serial priority resolving scheme as shown in Figure 6. Assume arbiter #1 has the bus and is holding busy low. Arbiter #2 detects its processor wants the bus and pulls low BREQ#2. If BPRN#2 is high (as shown), arbiter #2 will pull low CBRQ line. CBRQ signals to the higher priority arbiter #1 that a lower priority arbiter wants the bus. [A higher priority arbiter would be granted BPRN when it makes the bus request rather than having to wait for another arbiter to release the bus through CBRQ].** Arbiter #1 will relinquish the multi-master system bus when it enters a state not requiring it (see Table 1), by lowering its BPRO#1 (tied to BPRN#2) and releasing BUSY. Arbiter #2 now sees that it has priority from BPRN#2 being low and releases CBRQ. As soon as BUSY signifies the bus is available (high), arbiter #2 pulls BUSY low on 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 arbiter by lowering its BPRO #2 [TPNPO].

**Note that even a higher priority arbiter which is acquiring the bus through BPRN will momentarily drop CBRQ until it has acquired the bus.



8237/8237-2 HIGH PERFORMANCE PROGRAMMABLE DMA CONTROLLER

- Enable/Disable Control of Individual DMA Requests
 - Four Independent DMA Channels
 - Independent Autoinitialization of all Channels
 - Memory-to-Memory Transfers
 - Memory Block Initialization
 - Address Increment or Decrement
- High Performance: Transfers up to 1.6M Bytes/Second with 5 MHz 8237-2
 - Directly Expandable to any Number of Channels
 - End of Process Input for Terminating Transfers
 - Software DMA Requests
 - Independent Polarity Control for DREQ and DACK Signals

The 8237 Multimode Direct Memory Access (DMA) Controller is a peripheral interface circuit for microprocessor systems. It is designed to improve system performance by allowing external devices to directly transfer information to or from the system memory. Memory-to-memory transfer capability is also provided. The 8237 offers a wide variety of programmable control features to enhance data throughput and system optimization and to allow dynamic reconfiguration under program control.

The 8237 is designed to be used in conjunction 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 programmability of the types of DMA service by the user. Each channel can be individually programmed to Autoinitialize to its original condition following an End of Process (EOP).

Each channel has a full 64K address and word count capability.

The 8237-2 is a 5 MHz selected version of the standard 3 MHz 8237.



BLOCK DIAGRAM

Figure 1. Pin Configuration

PIN DEFINITIONS

V_{cc}: +5 volt supply

V_{ss}: Ground

CLK (Clock, Input)

This input controls the internal operations of the 8237 and its rate of data transfers. The input may be driven at up to 3 MHz for the standard 8237 and up to 5 MHz for the 8237-2.

\overline{CS} (Chip Select, Input)

Chip Select is an active low input used to select the 8237 as an I/O device during the Idle cycle. This allows CPU communication on the data bus.

RESET (Reset, Input)

Reset is an asynchronous active high input which clears the Command, Status, Request and Temporary registers. It also clears the first/last flip/flop and sets the Mask register. Following a Reset the device is in the Idle cycle.

READY (Ready, Input)

Ready is an input used to extend the memory read and write pulses from the 8237 to accommodate slow memories or I/O peripheral devices.

HLDA (Hold Acknowledge, Input)

The active high Hold Acknowledge from the CPU indicates that control of the system buses have been relinquished.

DREQ0-DREQ3 (DMA Request, Input)

The DMA Request lines are individual asynchronous channel request inputs used by peripheral circuits to obtain DMA service. In Fixed Priority, DREQ0 has the highest priority and DREQ3 has the lowest priority. A request is generated by activating the DREQ line of a channel. DACK will acknowledge the recognition of DREQ signal. Polarity of DREQ is programmable. Reset initializes these lines to active high. DREQ must be maintained until the corresponding DACK goes active.

DB0-DB7 (Data Bus, Input/Output)

The Data Bus lines are bidirectional three-state signals connected to the system data bus. The outputs are enabled in the Program Condition during the I/O Read to output the contents of an Address register, a Status register, the Temporary register or a Word Count register to the CPU. The outputs are disabled and the inputs are read during an I/O Write cycle when the CPU is programming the 8237 control registers. During DMA cycles the most significant 8 bits of the address are output onto the data bus to be strobed into an external latch by ADSTB. In memory-to-memory operations, data from the memory comes into the 8237 on the data bus during the read-from-memory transfer. In the write-to-memory transfer, the data bus outputs place the data into the new memory location.

\overline{IOR} (I/O Read, Input/Output)

I/O Read is a bidirectional active low three-state line. In the Idle cycle, it is an input control signal used by the CPU to read the control registers. In the Active cycle, it is an output control signal used by the 8237 to access data from a peripheral during a DMA Write transfer.

\overline{IOW} (I/O Write, Input/Output)

I/O Write is a bidirectional active low three-state line. In the Idle cycle, it is an input control signal used by the CPU to load information into the 8237. In the Active cycle, it is an output control signal used by the 8237 to load data to the peripheral during a DMA Read transfer.

\overline{EOP} (End of Process, Input/Output)

\overline{EOP} is an active low bidirectional signal. Information concerning the completion of DMA services is available at the bidirectional \overline{EOP} pin. The 8237 allows an external signal to terminate an active DMA service. This is accomplished by pulling the \overline{EOP} input low with an external \overline{EOP} signal. The 8237 also generates a pulse when the terminal count (TC) for any channel is reached. This generates an \overline{EOP} signal which is output through the \overline{EOP} Line. The reception of \overline{EOP} , either internal or external, will cause the 8237 to terminate the service, reset the request, and, if Autoinitialize is enabled, to write the base registers to the current registers of that channel. The mask bit and TC bit in the status word will be set for the currently active channel by \overline{EOP} unless the channel is programmed for Autoinitialize. In that case, the mask bit remains clear. During memory-to-memory transfers, \overline{EOP} will be output when the TC for channel 1 occurs. \overline{EOP} should be tied high with a pull-up resistor if it is not used to prevent erroneous end of process inputs.

A0-A3 (Address, Input/Output)

The four least significant address lines are bidirectional three-state signals. In the Idle cycle they are inputs and are used by the 8237 to address the control register to be loaded or read. In the Active cycle they are outputs and provide the lower 4 bits of the output address.

A4-A7 (Address, Output)

The four most significant address lines are three-state outputs and provide 4 bits of address. These lines are enabled only during the DMA service.

HRQ (Hold Request, Output)

This is the Hold Request to the CPU and is used to request control of the system bus. If the corresponding mask bit is clear, the presence of any valid DREQ causes the 8237 to issue the HRQ. After HRQ goes active at least one clock cycle (TCY) must occur before HLDA goes active.

DACK0-DACK3 (DMA Acknowledge, Output)

DMA Acknowledge is used to notify the individual peripherals when one has been granted a DMA cycle. The sense of these lines is programmable. Reset initializes them to active low.

AEN (Address Enable, Output)

This output enables the 8-bit latch containing the upper 8 address bits onto the system address bus. AEN can also be used to disable other system bus drivers during DMA transfers. AEN is active HIGH.

ADSTB (Address Strobe, Output)

The active high Address Strobe is used to strobe the upper address byte into an external latch.

MEMR (Memory Read, Output)

The Memory Read signal is an active low three-state output used to access data from the selected memory location during a DMA Read or a memory-to-memory transfer.

MEMW (Memory Write, Output)

The Memory Write signal is an active low three-state output used to write data to the selected memory location during a DMA Write or a memory-to-memory transfer.

FUNCTIONAL DESCRIPTION

The 8237 block diagram includes the major logic blocks and all of the internal registers. The data interconnection paths are also shown. Not shown are the various control signals between the blocks. The 8237 contains 344 bits of internal memory in the form of registers. Figure 2 lists these registers by name and shows the size of each. A detailed description of the registers and their functions can be found under Register Description.

Name	Size	Number
Base Address Registers	16 bits	4
Base Word Count Registers	16 bits	4
Current Address Registers	16 bits	4
Current Word Count Registers	16 bits	4
Temporary Address Register	16 bits	1
Temporary Word Count Register	16 bits	1
Status Register	8 bits	1
Command Register	8 bits	1
Temporary Register	8 bits	1
Mode Registers	6 bits	4
Mask Register	4 bits	1
Request Register	4 bits	1

Figure 2. 8237 Internal Registers

The 8237 contains three basic blocks of control logic. The Timing Control block generates internal timing and external control signals for the 8237. The Program Command Control block decodes the various commands given to the 8237 by the microprocessor prior to servicing a DMA Request. It also decodes the Mode Control word used to select the type of DMA during the servicing. The Priority Encoder block resolves priority contention between DMA channels requesting service simultaneously.

The Timing Control block derives internal timing from the clock input. In 8237 systems this input will usually be the ϕ 2 TTL clock from an 8224 or CLK from an 8085A. However, any appropriate system clock will suffice.

DMA OPERATION

The 8237 is designed to operate in two major cycles. These are called Idle and Active cycles. Each device cycle is made up of a number of states. The 8237 can assume seven separate states, each composed of one full clock period. State I (SI) is the inactive state. It is entered when the 8237 has no valid DMA requests pending. While in SI, the DMA controller is inactive but may be in the Program Condition, being programmed by the processor. State O (SO) is the first state of a DMA service. The 8237 has requested a hold but the processor has not yet returned an acknowledge. An acknowledge from the CPU will signal that transfers may begin. S1, S2, S3 and S4 are the working states of the DMA service. If more time is needed to complete a transfer than is available with normal timing, wait states (SW) can be inserted between S2 or S3 and S4 by the use of the Ready line on the 8237.

Memory-to-memory transfers require a read-from and a write-to-memory to complete each transfer. The states, which resemble the normal working states, use two digit numbers for identification. Eight states are required for a single transfer. The first four states (S11, S12, S13, S14) are used for the read-from-memory half and the last four states (S21, S22, S23, S24) for the write-to-memory half of the transfer.

IDLE CYCLE

When no channel is requesting service, the 8237 will enter the Idle cycle and perform "SI" states. In this cycle the 8237 will sample the DREQ lines every clock cycle to determine if any channel is requesting a DMA service. The device will also sample \overline{CS} , looking for an attempt by the microprocessor to write or read the internal registers of the 8237. When \overline{CS} is low and HRQ is low, the 8237 enters the Program Condition. The CPU can now establish, change or inspect the internal definition of the part by reading from or writing to the internal registers. Address lines A0-A3 are inputs to the device and select which registers will be read or written. The \overline{IOR} and \overline{IOW} lines are used to select and time reads or writes. Due to the number and size of the internal registers, an internal flip-flop is used to generate an additional bit of address. This bit is used to determine the upper or lower byte of the 16-bit Address and Word Count registers. The flip-flop is reset by Master Clear or Reset. A separate software command can also reset this flip-flop.

Special software commands can be executed by the 8237 in the Program Condition. These commands are decoded as sets of addresses with the \overline{CS} and \overline{IOW} . The commands do not make use of the data bus. Instructions include Clear First/Last Flip-flop and Master Clear.

ACTIVE CYCLE

When the 8237 is in the Idle cycle and a channel requests a DMA service, the device will output an HRQ to the microprocessor and enter the Active cycle. It is in this cycle that the DMA service will take place, in one of four modes:

Single Transfer Mode — In Single Transfer mode the device is programmed to make one transfer only. The

word count will be decremented and the address decremented or incremented following each transfer. When the word count goes to zero, a Terminal Count (TC) will cause an Autoinitialize if the channel has been programmed to do so.

DREQ must be held active until DACK becomes active in order to be recognized. If DREQ is held active throughout the single transfer, HRQ will go inactive and release the bus to the system. It will again go active and, upon receipt of a new HLDA, another single transfer will be performed. In 8080A/8085A systems this will ensure one full machine cycle execution between DMA transfers. Details of timing between the 8237 and other bus control protocols will depend upon the characteristics of the microprocessor involved.

Block Transfer Mode — In Block Transfer mode the device is activated by DREQ to continue making transfers during the service until a TC, caused by word count going to zero, or an external End of Process (EOP) is encountered. DREQ need only be held active until DACK becomes active. Again, an Autoinitialization will occur at the end of the service if the channel has been programmed for it.

Demand Transfer Mode — In Demand Transfer mode the device is programmed to continue making transfers until a TC or external EOP is encountered or until DREQ goes inactive. Thus transfers may continue until the I/O device has exhausted its data capacity. After the I/O device has had a chance to catch up, the DMA service is re-established by means of a DREQ. During the time between services when the microprocessor is allowed to operate, the intermediate values of address and word count are stored in the 8237 Current Address and Current Word Count registers. Only an EOP can cause an Autoinitialize at the end of the service. EOP is generated either by TC or by an external signal.

Cascade Mode — This mode is used to cascade more than one 8237 together for simple system expansion. The HRQ and HLDA signals from the additional 8237 are connected to the DREQ and DACK signals of a channel of the initial 8237. This allows the DMA requests of the additional device to propagate through the priority network circuitry of the preceding device. The priority chain is preserved and the new device must wait for its turn to acknowledge requests. Since the cascade channel in the initial device is used only for prioritizing the additional device, it does not output any address or control signals of its own. These would conflict with the outputs of the active channel in the added device. The 8237 will respond to DREQ and DACK but all other outputs except HRQ will be disabled.

Figure 3 shows two additional devices cascaded into an initial device using two of the previous channels. This forms a two level DMA system. More 8237s could be added at the second level by using the remaining channels of the first level. Additional devices can also be added by cascading into the channels of the second level devices, forming a third level.

TRANSFER TYPES

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

and Verify. Write transfers move data from an I/O device to the memory by activating $\overline{\text{MEMW}}$ and $\overline{\text{IOR}}$. Read transfers move data from memory to an I/O device by activating $\overline{\text{MEMR}}$ and $\overline{\text{IOW}}$. Verify transfers are pseudo transfers. The 8237 operates as in Read or Write transfers generating addresses, and responding to EOP, etc. However, the memory and I/O control lines all remain inactive.

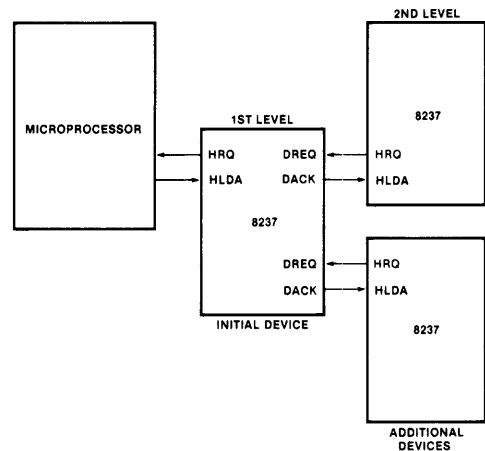


Figure 3. Cascaded 8237s

Memory-to-Memory — To perform block moves of data from one memory address space to another with a minimum of program effort and time, the 8237 includes a memory-to-memory transfer feature. Programming a bit in the Command register selects channels 0 and 1 to operate as memory-to-memory transfer channels. The transfer is initiated by setting the software DREQ for channel 0. The 8237 requests a DMA service in the normal manner. After HLDA is true, the device, using eight-state transfers in Block Transfer mode, reads data from the memory. The channel 0 Current Address register is the source for the address used and is decremented or incremented in the normal manner. The data byte read from the memory is stored in the 8237 internal Temporary register. Channel 1 then writes the data from the Temporary register to memory using the address in its Current Address register and incrementing or decrementing it in the normal manner. The channel 1 Current Word Count is decremented. When the word count of channel 1 goes to zero, a TC is generated causing an EOP output, terminating the service.

Channel 0 may be programmed to retain the same address for all transfers. This allows a single word to be written to a block of memory.

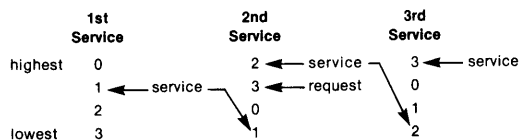
The 8237 will respond to external $\overline{\text{EOP}}$ signals during memory-to-memory transfers. Data comparators in block search schemes may use this input to terminate the service when a match is found. The timing of memory-to-memory transfers is found in Diagram 4. Memory-to-memory operations can be detected as an active AEN with no DACK outputs.

Autoinitialize — By programming a bit in the Mode register, a channel may be set up as an Autoinitialize

channel. During Autoinitialize initialization, the original values of the Current Address and Current Word Count registers are automatically restored from the Base Address and Base Word Count registers of that channel following EOP. The base registers are loaded simultaneously with the current registers by the microprocessor and remain unchanged throughout the DMA service. The mask bit is not set when the channel is in Autoinitialize. Following Autoinitialize the channel is ready to perform another service without CPU intervention.

Priority — The 8237 has two types of priority encoding available as software selectable options. The first is Fixed Priority which fixes the channels in priority order based upon the descending value of their number. The channel with the lowest priority is 3 followed by 2, 1 and the highest priority channel, 0. After the recognition of any one channel for service, the other channels are prevented from interfering with that service until it is completed.

The second scheme is Rotating Priority. The last channel to get service becomes the lowest priority channel with the others rotating accordingly.



With Rotating Priority in a single chip DMA system, any device requesting service is guaranteed to be recognized after no more than three higher priority services have occurred. This prevents any one channel from monopolizing the system.

Compressed Timing — In order to achieve even greater throughput where system characteristics permit, the 8237 can compress the transfer time to two clock cycles. From Timing Diagram 3 it can be seen that state S3 is used to extend the access time of the read pulse. By removing state S3, the read pulse width is made equal to the write pulse width and a transfer consists only of state S2 to change the address and state S4 to perform the read/write. S1 states will still occur when A8-A15 need updating (see Address Generation). Timing for compressed transfers is found in Diagram 6.

Address Generation — In order to reduce pin count, the 8237 multiplexes the eight higher order address bits on the data lines. State S1 is used to output the higher order address bits to an external latch from which they may be placed on the address bus. The falling edge of Address Strobe (ADSTB) is used to load these bits from the data lines to the latch. Address Enable (AEN) is used to enable the bits onto the address bus through a three-state enable. The lower order address bits are output by the 8237 directly. Lines A0-A7 should be connected to the address bus. Timing Diagram 3 shows the time relationships between CLK, AEN, ADSTB, DB0-DB7 and A0-A7.

During Block and Demand Transfer mode services, which include multiple transfers, the addresses generated will be sequential. For many transfers the data held in the external address latch will remain the same. This data need only change when a carry or borrow from A7 to A8 takes place in the normal sequence of addresses. To save time and speed transfers, the 8237 executes S1 states only when updating of A8-A15 in the latch is necessary. This means for long services, S1 states may occur only once every 256 transfers, a savings of 255 clock cycles for each 256 transfers.

REGISTER DESCRIPTION

Current Address Register — Each channel has a 16-bit Current Address register. This register holds the value of the address used during DMA transfers. The address is automatically incremented or decremented after each transfer and the intermediate values of the address are stored in the Current Address register during the transfer. This register is written or read by the microprocessor in successive 8-bit bytes. It may also be reinitialized by an Autoinitialize back to its original value. Autoinitialize takes place only after an EOP.

Current Word Register — Each channel has a 16-bit Current Word Count register. This register holds the number of transfers to be performed. The word count is decremented after each transfer. The intermediate value of the word count is stored in the register during the transfer. When the value in the register goes to zero, a TC will be generated. This register is loaded or read in successive 8-bit bytes by the microprocessor in the Program Condition. Following the end of a DMA service it may also be reinitialized by an Autoinitialization back to its original value. Autoinitialize can occur only when an EOP occurs.

Base Address and Base Word Count Registers — Each channel has a pair of Base Address and Base Word Count registers. These 16-bit registers store the original value of their associated current registers. During Autoinitialize these values are used to restore the current registers to their original values. The base registers are written simultaneously with their corresponding current register in 8-bit bytes in the Program Condition by the microprocessor. These registers cannot be read by the microprocessor.

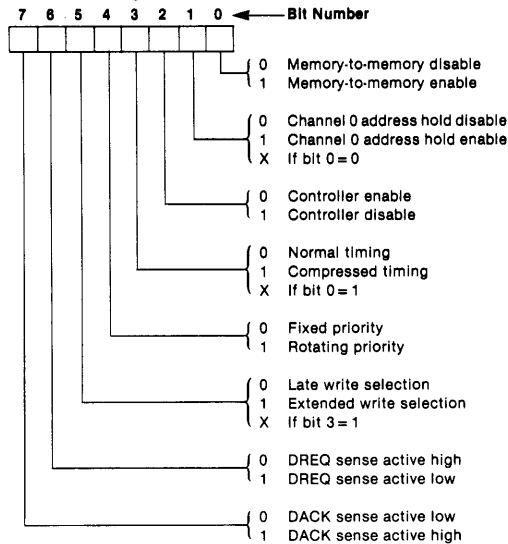
Command Register — This 8-bit register controls the operation of the 8237. It is programmed by the microprocessor in the Program Condition and is cleared by Reset. The following table lists the function of the command bits. See Figure 6 for address coding.

Mode Register — Each channel has a 6-bit Mode register associated with it. When the register is being written to by the microprocessor in the Program Condition, bits 0 and 1 determine which channel Mode register is to be written.

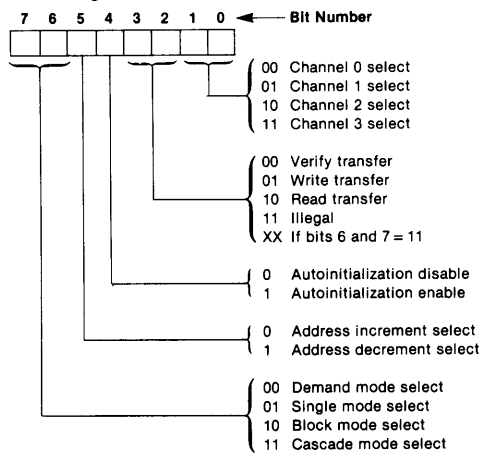
Request Register — The 8237 can respond to requests for DMA service which are initiated by software as well as by a DREQ. Each channel has a request bit associated with it in the 4-bit Request register. These are non-

maskable and subject to prioritization by the Priority Encoder network. Each register bit is set or reset separately under software control or is cleared upon generation of a TC or external EOP. The entire register is cleared by a Reset. To set or reset a bit, the software loads the proper form of the data word. See Figure 4 for address coding.

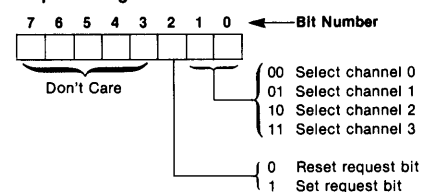
Command Register



Mode Register

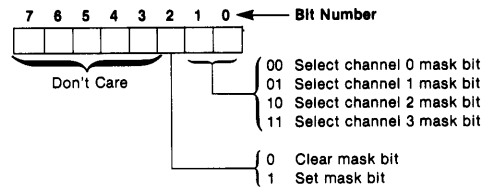


Request Register

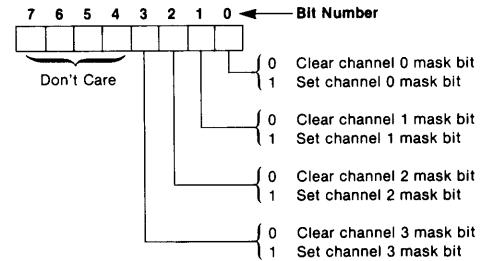


Software requests will be serviced only if the channel is in Block mode. When initiating a memory-to-memory transfer, the software request for channel 0 should be set.

Mask Register — Each channel has associated with it a mask bit which can be set to disable the incoming DREQ. Each mask bit is set when its associated channel produces an EOP if the channel is not programmed for Autoinitialize. Each bit of the 4-bit Mask register may also be set or cleared separately under software control. The entire register is also set by a Reset. This disables all DMA requests until a clear Mask register instruction allows them to occur. The instruction to separately set or clear the mask bits is similar in form to that used with the Request register. See Figure 4 for instruction addressing.



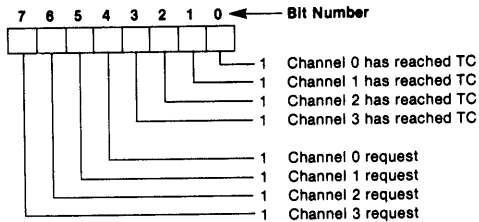
All four bits of the Mask register may also be written with a single command.



Register	Operation	Signals						
		CS	IOR	IOW	A3	A2	A1	A0
Command	Write	0	1	0	1	0	0	0
Mode	Write	0	1	0	1	0	1	1
Request	Write	0	1	0	1	0	0	1
Mask	Set/Reset	0	1	0	1	0	1	0
Mask	Write	0	1	0	1	1	1	1
Temporary	Read	0	0	1	1	1	0	1
Status	Read	0	0	1	1	0	0	0

Figure 4. Definition of Register Codes

Status Register — The Status register is available to be read out of the 8237 by the microprocessor. It contains information about the status of the devices at this point. This information includes which channels have reached a terminal count and which channels have pending DMA requests. Bits 0-3 are set every time a TC is reached by that channel or an external EOP is applied. These bits are cleared upon Reset and on each Status Read. Bits 4-7 are set whenever their corresponding channel is requesting service.



Temporary Register — The Temporary register is used to hold data during memory-to-memory transfers. Following the completion of the transfers, the last word moved can be read by the microprocessor in the Program Condition. The Temporary register always contains the last byte transferred in the previous memory-to-memory operation, unless cleared by a Reset.

Software Commands — These are additional special software commands which can be executed in the Program Condition. They do not depend on any specific bit pattern on the data bus. The two software commands are:

Clear First/Last Flip-Flop: This command is executed prior to writing or reading new address or word count information to the 8237. This initializes the flip-flop to a known state so that subsequent accesses to register contents by the microprocessor will address upper and lower bytes in the correct sequence.

Master Clear: This software instruction has the same effect as the hardware Reset. The Command, Status, Request, Temporary, and Internal First/Last Flip-Flop registers are cleared and the Mask register is set. The 8237 will enter the Idle cycle.

Figure 5 lists the address codes for the software commands:

Signals						Operation
A3	A2	A1	A0	IOR	IOW	
1	0	0	0	0	1	Read Status Register
1	0	0	0	1	0	Write Command Register
1	0	0	1	0	1	Illegal
1	0	0	1	1	0	Write Request Register
1	0	1	0	0	1	Illegal
1	0	1	0	1	0	Write Single Mask Register Bit
1	0	1	1	0	1	Illegal
1	0	1	1	1	0	Write Mode Register
1	1	0	0	0	1	Illegal
1	1	0	0	1	0	Clear Byte Pointer Flip/Flop
1	1	0	1	0	1	Read Temporary Register
1	1	0	1	1	0	Master Clear
1	1	1	0	0	1	Illegal
1	1	1	0	1	0	Illegal
1	1	1	1	0	1	Illegal
1	1	1	1	1	0	Write All Mask Register Bits

Figure 5. Software Command Codes

Channel	Register	Operation	Signals						Internal Flip-Flop	Data Bus DB0-DB7	
			CS	IOR	IOW	A3	A2	A1			A0
0	Base and Current Address	Write	0	1	0	0	0	0	0	0	A0-A7
			0	1	0	0	0	0	0	1	A8-A15
	Current Address	Read	0	0	1	0	0	0	0	0	A0-A7
			0	0	1	0	0	0	0	1	A8-A15
1	Base and Current Address	Write	0	1	0	0	0	1	0	0	A0-A7
			0	1	0	0	0	1	0	1	A8-A15
	Current Address	Read	0	0	1	0	0	1	0	0	A0-A7
			0	0	1	0	0	1	0	1	A8-A15
2	Base and Current Address	Write	0	1	0	0	0	1	0	0	A0-A7
			0	1	0	0	1	0	0	1	A8-A15
	Current Address	Read	0	0	1	0	1	0	0	0	A0-A7
			0	0	1	0	1	0	0	1	A8-A15
3	Base and Current Address	Write	0	1	0	0	1	1	0	0	A0-A7
			0	1	0	0	1	1	0	1	A8-A15
	Current Address	Read	0	0	1	0	1	1	0	0	A0-A7
			0	0	1	0	1	1	0	1	A8-A15
2	Base and Current Address	Write	0	1	0	0	1	1	1	0	W0-W7
			0	1	0	0	1	1	1	1	W8-W15
	Current Address	Read	0	0	1	0	1	1	1	0	W0-W7
			0	0	1	0	1	1	1	1	W8-W15

Figure 6. Word Count and Address Register Command Codes

APPLICATION INFORMATION

Figure 7 shows a convenient method for configuring a DMA system with the 8237 controller and an 8080A/8085A microprocessor system. The multimode DMA controller issues a HRQ to the processor whenever there is at least one valid DMA request from a peripheral device. When the processor replies with a HLDA signal, the 8237 takes control of the address bus, the data bus and the control bus. The address for the

first transfer operation comes out in two bytes — the least significant 8 bits on the eight address outputs and the most significant 8 bits on the data bus. The contents of the data bus are then latched into the 8282 8-bit latch to complete the full 16 bits of the address bus. The 8282 is a high speed, 8-bit, three-state latch in a 20-pin package. After the initial transfer takes place, the latch is updated only after a carry or borrow is generated in the least significant address byte. Four DMA channels are provided when one 8237 is used.

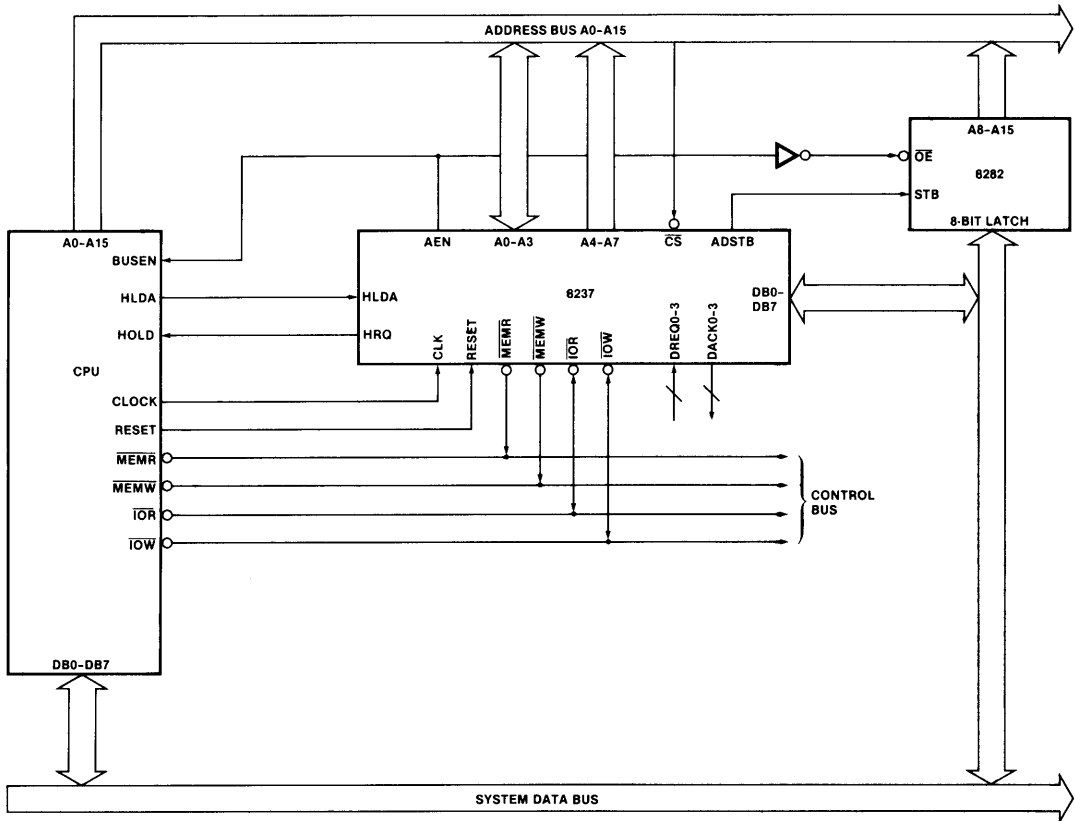


Figure 7

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature under Bias 0°C to 70°C
 Storage Temperature - 65°C to + 150°C
 Voltage on any Pin with
 Respect to Ground - 0.5 to 7V
 Power Dissipation 1.5 Watt

*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.

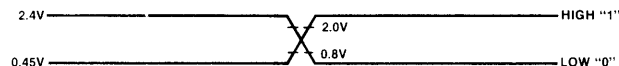
D.C. CHARACTERISTICS

$T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = 5.0\text{V} \pm 5\%$, $\text{GND} = 0\text{V}$

Symbol	Parameter	Min.	Typ. ⁽¹⁾	Max.	Unit	Test Conditions
V_{OH}	Output HIGH Voltage	2.4			V	$I_{OH} = -200 \mu\text{A}$
		3.3			V	$I_{OH} = -100 \mu\text{A}$ (HRQ Only)
V_{OL}	Output LOW Voltage			0.4	V	$I_{OL} = 3.2 \text{ mA}$
V_{IH}	Input HIGH Voltage			$V_{CC} + 0.5$	V	
V_{IL}	Input LOW Voltage	- 0.5		0.8	V	
I_{LI}	Input Load Current			± 10	μA	$V_{SS} \leq V_I \leq V_{CC}$
I_{LO}	Output Leakage Current			± 10	μA	$V_{CC} \leq V_O \leq V_{SS} + 0.40$
I_{CC}	V_{CC} Supply Current		65	130	mA	$T_A = +25^\circ\text{C}$
			75	150	mA	$T_A = 0^\circ\text{C}$
C_O	Output Capacitance		4	8	pF	$f_c = 1.0 \text{ MHz}$, Inputs = 0V
C_I	Input Capacitance		8	15	pF	
C_{IO}	I/O Capacitance		10	18	pF	

Notes:

- Typical values are for $T_A = 25^\circ\text{C}$, nominal supply voltage and nominal processing parameters.
- Input timing parameters assume transition times of 20 ns or less. Waveform measurement points for both input and output signals are 2.0V for HIGH and 0.8V for LOW, unless otherwise noted.
- Output loading is 1 TTL gate plus 50 pF capacitance, unless otherwise noted.
- The net \overline{IOW} or \overline{MEMW} Pulse width for normal write will be $TCY - 100$ ns and for extended write will be $2TCY - 100$ ns. The net \overline{IOR} or \overline{MEMR} pulse width for normal read will be $2TCY - 50$ ns and for compressed read will be $TCY - 50$ ns.
- TDQ is specified for two different output HIGH levels. TDQ1 is measured at 2.0V. TDQ2 is measured at 3.3V. The value for TDQ2 assumes an external 3.3 k Ω pull-up resistor connected from HRQ to V_{CC} .
- DREQ should be held active until DACK is returned.
- DREQ and DACK signals may be active high or active low. Timing diagrams assume the active high mode.
- Output loading on the data bus is 1 TTL gate plus 100 pF capacitance.
- Successive read and/or write operations by the external processor to program or examine the controller must be timed to allow at least 600 ns for the 8237 and at least 400 ns for the 8237-2 as recovery time between active read or write pulses.
- Parameters are listed in alphabetical order.
- Pin 5 is an input that should always be at a logic high level. An internal pull-up resistor will establish a logic high when the pin is left floating. Alternatively, pin 5 may be tied to V_{CC} .

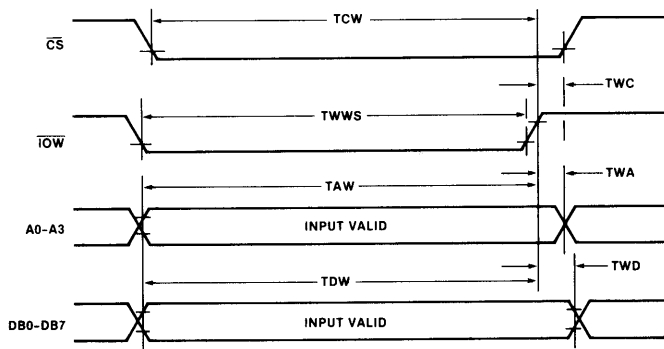
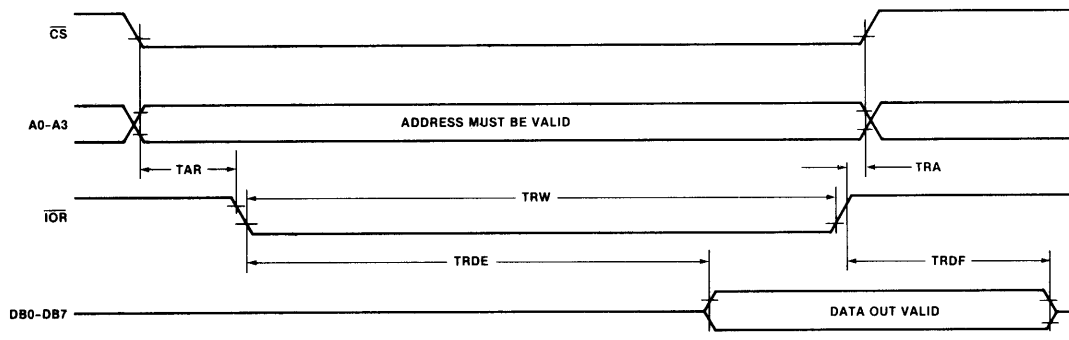
A.C. TEST WAVEFORM

A.C. CHARACTERISTICS: DMA (MASTER) MODET_A = 0 °C to 70 °C, V_{CC} = 5.0V ± 5%, GND = 0V

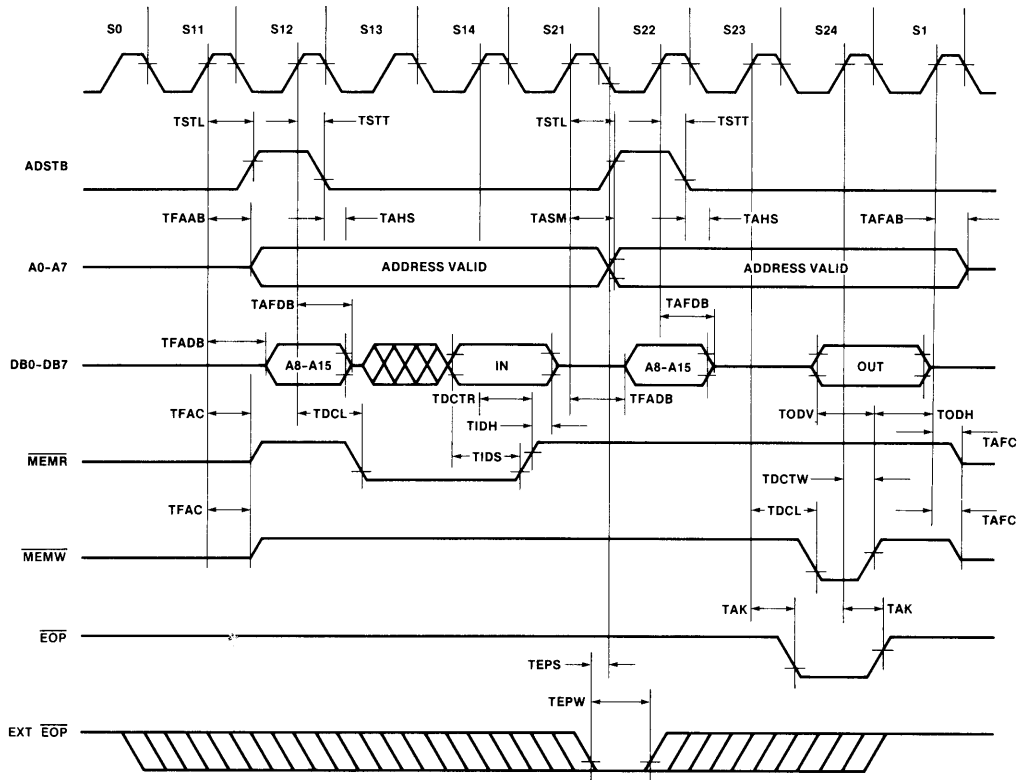
Symbol	Parameter	8237		8237-2		Unit
		Min.	Max.	Min.	Max.	
TAEL	AEN HIGH from CLK LOW (S1) Delay Time		300		200	ns
TAET	AEN LOW from CLK HIGH (S1) Delay Time		200		130	ns
TAFAB	ADR Active to Float Delay from CLK HIGH		150		90	ns
TAFC	$\overline{\text{READ}}$ or $\overline{\text{WRITE}}$ Float from CLK HIGH		150		120	ns
TAFDB	DB Active to Float Delay from CLK HIGH		250		170	ns
TAHR	ADR from $\overline{\text{READ}}$ HIGH Hold Time	TCY-100		TCY-100		ns
TAHS	DB from ADSTB LOW Hold Time	50		30		ns
TAHW	ADR from $\overline{\text{WRITE}}$ HIGH Hold Time	TCY-50		TCY-50		ns
TAK	DACK Valid from CLK LOW Delay Time		250		170	ns
	$\overline{\text{EOP}}$ HIGH from CLK HIGH Delay Time		250		170	ns
	$\overline{\text{EOP}}$ LOW to CLK HIGH Delay Time		250		100	ns
TASM	ADR Stable from CLK HIGH		250		170	ns
TASS	DB to ADSTB LOW Setup Time	100		100		ns
TCH	Clock High Time (Transitions \leq 10 ns)	120		70		ns
TCL	Clock LOW Time (Transitions \leq 10 ns)	150		50		ns
TCY	CLK Cycle Time	320		200		ns
TDCL	CLK HIGH to $\overline{\text{READ}}$ or $\overline{\text{WRITE}}$ LOW Delay (Note 4)		270		190	ns
TDCTR	$\overline{\text{READ}}$ HIGH from CLK HIGH (S4) Delay Time (Note 4)		270		190	ns
TDCTW	$\overline{\text{WRITE}}$ HIGH from CLK HIGH (S4) Delay Time (Note 4)		200		130	ns
TDQ1	HRQ Valid from CLK HIGH Delay Time (Note 5)		160		120	ns
TDQ2			250		120	ns
TEPS	$\overline{\text{EOP}}$ LOW from CLK LOW Setup Time	60		40		ns
TEPW	$\overline{\text{EOP}}$ Pulse Width	300		220		ns
TFAAB	ADR Float to Active Delay from CLK HIGH		250		170	ns
TFAC	$\overline{\text{READ}}$ or $\overline{\text{WRITE}}$ Active from CLK HIGH		200		150	ns
TFADB	DB Float to Active Delay from CLK HIGH		300		200	ns
THS	HCDA Valid to CLK HIGH Setup Time	100		75		ns
TIDH	Input Data from $\overline{\text{MEMR}}$ HIGH Hold Time	0		0		ns
TIDS	Input Data to $\overline{\text{MEMR}}$ HIGH Setup Time	250		170		ns
TODH	Output Data from $\overline{\text{MEMW}}$ HIGH Hold Time	20		10		ns
TODV	Output Data Valid to $\overline{\text{MEMW}}$ HIGH	200		130		ns
TQS	DREQ to CLK LOW (S1, S4) Setup Time	0		0		ns
TRH	CLK to READY LOW Hold Time	20		20		ns
TRS	READY to CLK LOW Setup Time	100		75		ns
TSTL	ADSTB HIGH from CLK HIGH Delay Time		200		130	ns
TSTT	ADSTB LOW from CLK HIGH Delay Time		140		90	ns

A.C. CHARACTERISTICS: PERIPHERAL (SLAVE) MODET_A = 0°C to 70°C, V_{CC} = 5.0V ± 5%, GND = 0V

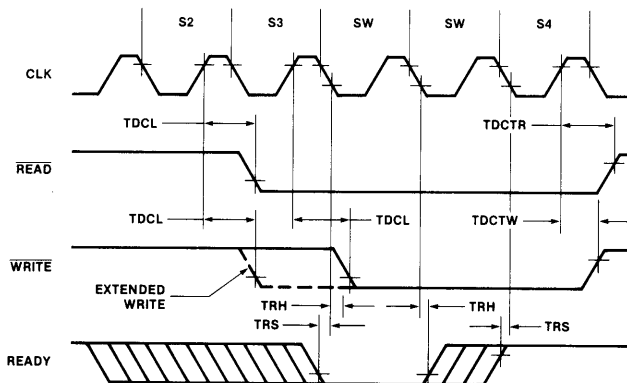
Symbol	Parameter	8237		8237-2		Unit
		Min.	Max.	Min.	Max.	
TAR	ADR Valid or \overline{CS} LOW to \overline{READ} LOW	50		50		ns
TAW	ADR Valid to \overline{WRITE} HIGH Setup Time	200		160		ns
TCW	CS LOW to \overline{WRITE} HIGH Setup Time	200		160		ns
TDW	Data Valid to \overline{WRITE} HIGH Setup Time	200		160		ns
TRA	ADR or CS Hold from \overline{READ} HIGH	0		0		ns
TRDE	Data Access from \overline{READ} LOW (Note 8)		200		140	ns
TRDF	DB Float Delay from \overline{READ} HIGH	20	100	0	70	ns
TRSTD	Power Supply HIGH to RESET LOW Setup Time	500		500		μs
TRSTS	RESET to First \overline{IOWR}	2TCY		2TCY		ns
TRSTW	RESET Pulse Width	300		300		ns
TRW	\overline{READ} Width	300		200		ns
TWA	ADR from \overline{WRITE} HIGH Hold Time	20		0		ns
TWC	\overline{CS} HIGH from \overline{WRITE} HIGH Hold Time	20		0		ns
TWD	Data from \overline{WRITE} HIGH Hold Time	30		10		ns
TWWS	Write Width	200		160		ns

TIMING DIAGRAM #1 — SLAVE MODE WRITE TIMING**TIMING DIAGRAM #2 — SLAVE MODE READ TIMING**

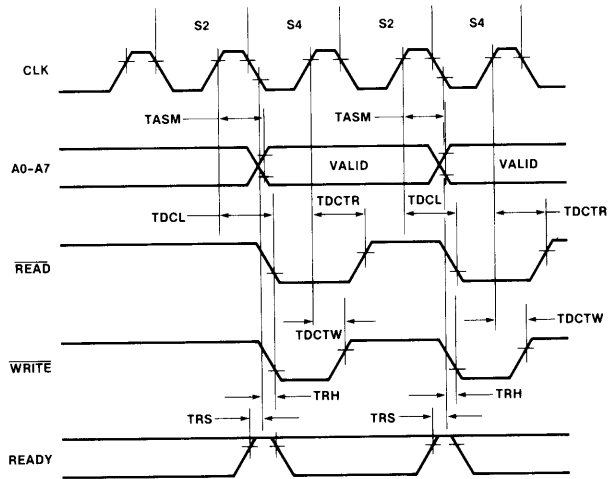
TIMING DIAGRAM #4 — MEMORY TO MEMORY TRANSFER TIMING



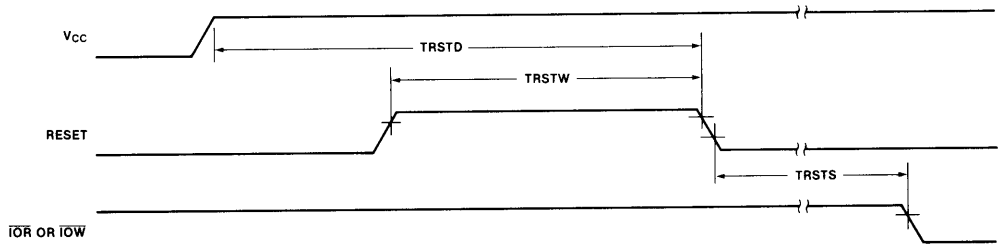
TIMING DIAGRAM #5 — READY TIMING



TIMING DIAGRAM #6 — COMPRESSED TRANSFER TIMING



TIMING DIAGRAM #7 — RESET TIMING





8259A/8259A-2/8259A-8 PROGRAMMABLE INTERRUPT CONTROLLER

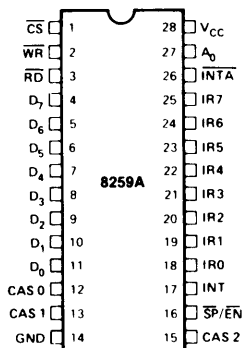
- 8086/8088 Compatible
 - MCS-80/85™ Compatible
 - Eight-Level Priority Controller
 - Expandable to 64 Levels
- Programmable Interrupt Modes
 - Individual Request Mask Capability
 - Single +5V Supply (No Clocks)
 - 28-Pin Dual-In-Line Package

The Intel® 8259A Programmable Interrupt Controller handles up to eight vectored priority interrupts for the CPU. It is cascadable for up to 64 vectored priority interrupts without additional circuitry. It is packaged in a 28-pin DIP, uses NMOS technology and requires a single +5V supply. Circuitry is static, requiring no clock input.

The 8259A is designed to minimize the software and real time overhead in handling multi-level priority interrupts. It has several modes, permitting optimization for a variety of system requirements.

The 8259A is fully upward compatible with the Intel® 8259. Software originally written for the 8259 will operate the 8259A in all 8259 equivalent modes (MCS-80/85, Non-Buffered, Edge Triggered).

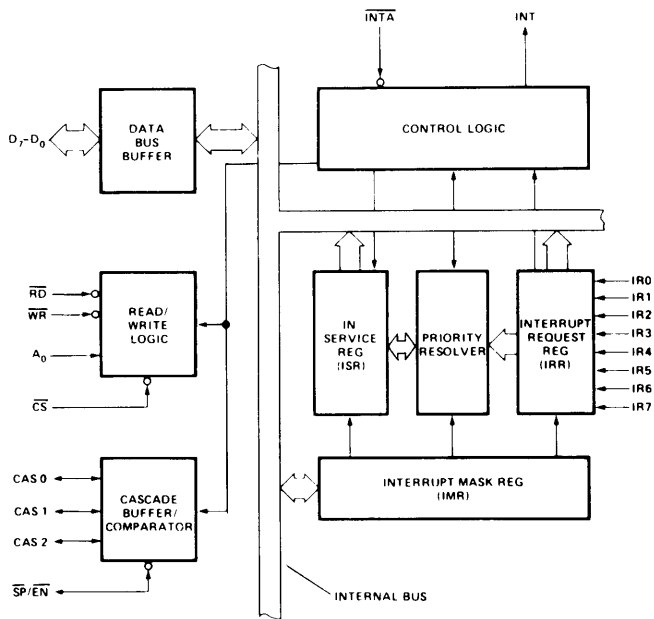
PIN CONFIGURATION



PIN NAMES

D ₇ -D ₀	DATA BUS (BI DIRECTIONAL)
RD	READ INPUT
WR	WRITE INPUT
A ₀	COMMAND SELECT ADDRESS
CS	CHIP SELECT
CAS2 CAS0	CASCADE LINES
SP/EN	SLAVE PROGRAM/ENABLE BUFFER
INT	INTERRUPT OUTPUT
INTA	INTERRUPT ACKNOWLEDGE INPUT
IR0-IR7	INTERRUPT REQUEST INPUTS

BLOCK DIAGRAM



INTERRUPTS IN MICROCOMPUTER SYSTEMS

Microcomputer system design requires that I/O devices such as keyboards, displays, sensors and other components receive servicing in an efficient manner so that large amounts of the total system tasks can be assumed by the microcomputer with little or no effect on throughput.

The most common method of servicing such devices is the *Polled* approach. This is where the processor must test each device in sequence and in effect "ask" each one if it needs servicing. It is easy to see that a large portion of the main program is looping through this continuous polling cycle and that such a method would have a serious, detrimental effect on system throughput, thus limiting the tasks that could be assumed by the microcomputer and reducing the cost effectiveness of using such devices.

A more desirable method would be one that would allow the microprocessor to be executing its main program and only stop to service peripheral devices when it is told to do so by the device itself. In effect, the method would provide an external asynchronous input that would inform the processor that it should complete whatever instruction that is currently being executed and fetch a new routine that will service the requesting device. Once this servicing is complete, however, the processor would resume exactly where it left off.

This method is called *Interrupt*. It is easy to see that system throughput would drastically increase, and thus more tasks could be assumed by the microcomputer to further enhance its cost effectiveness.

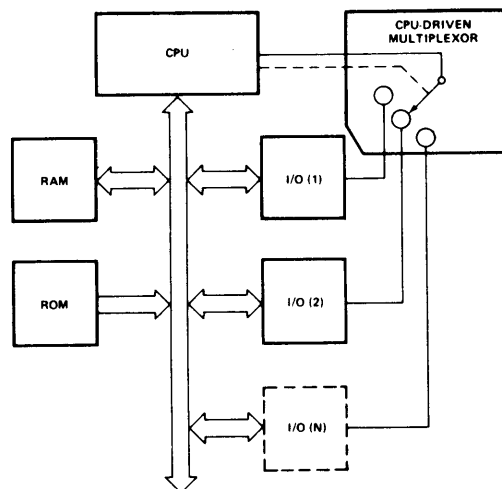
The Programmable Interrupt Controller (PIC) functions as an overall manager in an Interrupt-Driven system environment. It accepts requests from the peripheral equipment, determines which of the incoming requests is of the highest importance (priority), ascertains whether the incoming request has a higher priority value than the level currently being serviced, and issues an interrupt to the CPU based on this determination.

Each peripheral device or structure usually has a special program or "routine" that is associated with its specific functional or operational requirements; this is referred to as a "service routine". The PIC, after issuing an interrupt to the CPU, must somehow input information into the CPU that can "point" the Program Counter to the service routine associated with the requesting device. This "pointer" is an address in a vectoring table and will often be referred to, in this document, as vectoring data.

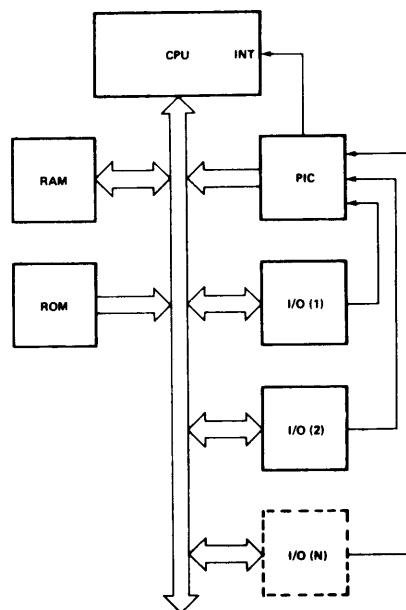
8259A BASIC FUNCTIONAL DESCRIPTION GENERAL

The 8259A is a device specifically designed for use in real time, interrupt driven microcomputer systems. It manages eight levels or requests and has built-in features for expandability to other 8259A's (up to 64 levels). It is programmed by the system's software as an I/O peripheral. A selection of priority modes is available to the programmer so that the manner in which the requests are processed by the 8259A can be configured to

match his system requirements. The priority modes can be changed or reconfigured dynamically at any time during the main program. This means that the complete interrupt structure can be defined as required, based on the total system environment.



Polled Method



Interrupt Method

INTERRUPT REQUEST REGISTER (IRR) AND IN-SERVICE REGISTER (ISR)

The interrupts at the IR input lines are handled by two registers in cascade, the Interrupt Request Register (IRR) and the In-Service Register (ISR). The IRR is used to store all the interrupt levels which are requesting service; and the ISR is used to store all the interrupt levels which are being serviced.

PRIORITY RESOLVER

This logic block determines the priorities of the bits set in the IRR. The highest priority is selected and strobed into the corresponding bit of the ISR during INTA pulse.

INTERRUPT MASK REGISTER (IMR)

The IMR stores the bits which mask the interrupt lines to be masked. The IMR operates on the IRR. Masking of a higher priority input will not affect the interrupt request lines of lower priority.

INT (INTERRUPT)

This output goes directly to the CPU interrupt input. The V_{OH} level on this line is designed to be fully compatible with the 8080A, 8085A, 8086 and 8088.

INTA (INTERRUPT ACKNOWLEDGE)

INTA pulses will cause the 8259A to release vectoring information onto the data bus. The format of this data depends on the system mode (μ PM) of the 8259A.

DATA BUS BUFFER

This 3-state, bidirectional 8-bit buffer is used to interface the 8259A to the system Data Bus. Control words and status information are transferred through the Data Bus Buffer.

READ/WRITE CONTROL LOGIC

The function of this block is to accept OUTPUT commands from the CPU. It contains the Initialization Command Word (ICW) registers and Operation Command Word (OCW) registers which store the various control formats for device operation. This function block also allows the status of the 8259A to be transferred onto the Data Bus.

\overline{CS} (CHIP SELECT)

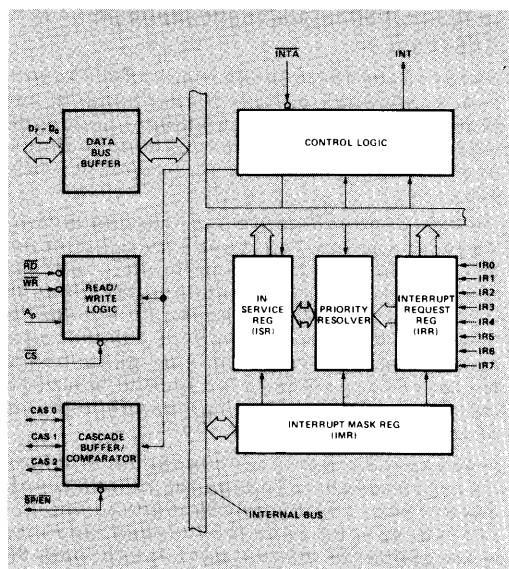
A LOW on this input enables the 8259A. No reading or writing of the chip will occur unless the device is selected.

\overline{WR} (WRITE)

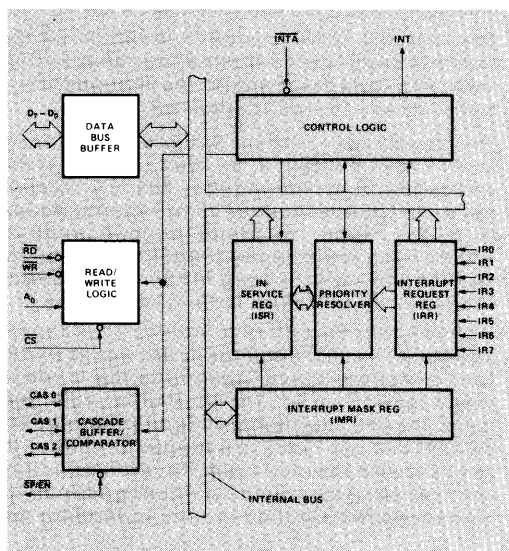
A LOW on this input enables the CPU to write control words (ICWs and OCWs) to the 8259A.

\overline{RD} (READ)

A LOW on this input enables the 8259A to send the status of the Interrupt Request Register (IRR), In Service Register (ISR), the Interrupt Mask Register (IMR), or the Interrupt level onto the Data Bus.



8259A Block Diagram



8259A Block Diagram

A_0

This input signal is used in conjunction with \overline{WR} and \overline{RD} signals to write commands into the various command registers, as well as reading the various status registers of the chip. This line can be tied directly to one of the address lines.

THE CASCADE BUFFER/COMPARATOR

This function block stores and compares the IDs of all 8259A's used in the system. The associated three I/O pins (CAS0-2) are outputs when the 8259A is used as a master and are inputs when the 8259A is used as a slave. As a master, the 8259A sends the ID of the interrupting slave device onto the CAS0-2 lines. The slave thus selected will send its preprogrammed subroutine address onto the Data Bus during the next one or two consecutive $\overline{\text{INTA}}$ pulses. (See section "Cascading the 8259A".)

INTERRUPT SEQUENCE

The powerful features of the 8259A in a microcomputer system are its programmability and the interrupt routine addressing capability. The latter allows direct or indirect jumping to the specific interrupt routine requested without any polling of the interrupting devices. The normal sequence of events during an interrupt depends on the type of CPU being used.

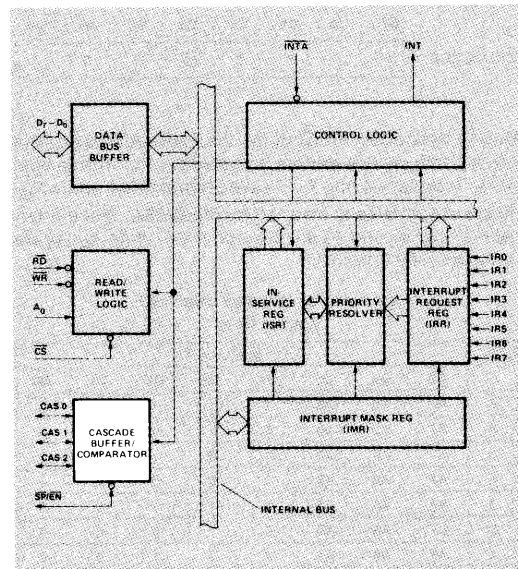
The events occur as follows in an MCS-80/85 system:

1. One or more of the INTERRUPT REQUEST lines (IR7-0) are raised high, setting the corresponding IRR bit(s).
2. The 8259A evaluates these requests, and sends an INT to the CPU, if appropriate.
3. The CPU acknowledges the INT and responds with an $\overline{\text{INTA}}$ pulse.
4. Upon receiving an $\overline{\text{INTA}}$ from the CPU group, the highest priority ISR bit is set, and the corresponding IRR bit is reset. The 8259A will also release a CALL instruction code (11001101) onto the 8-bit Data Bus through its D7-0 pins.
5. This CALL instruction will initiate two more $\overline{\text{INTA}}$ pulses to be sent to the 8259A from the CPU group.
6. These two $\overline{\text{INTA}}$ pulses allow the 8259A to release its preprogrammed subroutine address onto the Data Bus. The lower 8-bit address is released at the first $\overline{\text{INTA}}$ pulse and the higher 8-bit address is released at the second $\overline{\text{INTA}}$ pulse.
7. This completes the 3-byte CALL instruction released by the 8259A. In the AEOL mode the ISR bit is reset at the end of the third $\overline{\text{INTA}}$ pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt sequence.

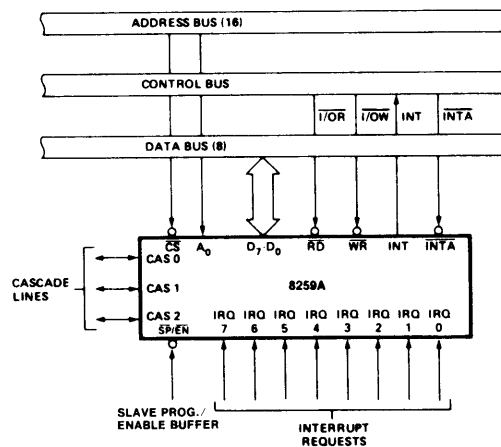
The events occurring in an 8086/8088 system are the same until step 4.

4. Upon receiving an $\overline{\text{INTA}}$ from the CPU group, the highest priority ISR bit is set and the corresponding IRR bit is reset. The 8259A does not drive the Data Bus during this cycle.
5. The 8086/8088 CPU will initiate a second $\overline{\text{INTA}}$ pulse. During this pulse, the 8259A releases an 8-bit pointer onto the Data Bus where it is read by the CPU.
6. This completes the interrupt cycle. In the AEOL mode the ISR bit is reset at the end of the second $\overline{\text{INTA}}$ pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

If no interrupt request is present at step 4 of either sequence (i.e., the request was too short in duration) the 8259A will issue an interrupt level 7. Both the vectoring bytes and the CAS lines will look like an interrupt level 7 was requested.



8259A Block Diagram



8259A Interface to Standard System Bus

INTERRUPT SEQUENCE OUTPUTS**MCS-80/85 MODE**

This sequence is timed by three \overline{INTA} pulses. During the first \overline{INTA} pulse the CALL opcode is enabled onto the data bus.

**Content of First Interrupt
Vector Byte**

	D7	D6	D5	D4	D3	D2	D1	D0
CALL CODE	1	1	0	0	1	1	0	1

During the second \overline{INTA} pulse the lower address of the appropriate service routine is enabled onto the data bus. When Interval = 4 bits A₅-A₇ are programmed, while A₀-A₄ are automatically inserted by the 8259A. When Interval = 8 only A₆ and A₇ are programmed, while A₀-A₅ are automatically inserted.

**Content of Second Interrupt
Vector Byte**

IR	Interval = 4							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	A5	1	1	1	0	0
6	A7	A6	A5	1	1	0	0	0
5	A7	A6	A5	1	0	1	0	0
4	A7	A6	A5	1	0	0	0	0
3	A7	A6	A5	0	1	1	0	0
2	A7	A6	A5	0	1	0	0	0
1	A7	A6	A5	0	0	1	0	0
0	A7	A6	A5	0	0	0	0	0

IR	Interval = 8							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	1	1	1	0	0	0
6	A7	A6	1	1	0	0	0	0
5	A7	A6	1	0	1	0	0	0
4	A7	A6	1	0	0	0	0	0
3	A7	A6	0	1	1	0	0	0
2	A7	A6	0	1	0	0	0	0
1	A7	A6	0	0	1	0	0	0
0	A7	A6	0	0	0	0	0	0

During the third \overline{INTA} pulse the higher address of the appropriate service routine, which was programmed as byte 2 of the initialization sequence (A₈-A₁₅), is enabled onto the bus.

**Content of Third Interrupt
Vector Byte**

D7	D6	D5	D4	D3	D2	D1	D0
A15	A14	A13	A12	A11	A10	A9	A8

8086/8088 Mode

8086/8088 mode is similar to MCS80/85 mode except that only two Interrupt Acknowledge cycles are issued by the processor and no CALL opcode is sent to the processor. The first interrupt acknowledge cycle is similar to that of MCS-80/85 systems in that the 8259A uses it to internally freeze the state of the interrupts for priority resolution and as a master it issues the interrupt code on the cascade lines at the end of the \overline{INTA} pulse. On this first cycle it does not issue any data to the processor and leaves its data bus buffers disabled. On the second interrupt acknowledge cycle in 8086/8088 mode the master (or slave if so programmed) will send a byte of data to the processor with the acknowledged interrupt code composed as follows (note the state of the ADI mode control is ignored and A₅-A₁₁ are unused in 8086/8088 mode):

	D7	D6	D5	D4	D3	D2	D1	D0
IR7	T7	T6	T5	T4	T3	1	1	1
IR6	T7	T6	T5	T4	T3	1	1	0
IR5	T7	T6	T5	T4	T3	1	0	1
IR4	T7	T6	T5	T4	T3	1	0	0
IR3	T7	T6	T5	T4	T3	0	1	1
IR2	T7	T6	T5	T4	T3	0	1	0
IR1	T7	T6	T5	T4	T3	0	0	1
IR0	T7	T6	T5	T4	T3	0	0	0

PROGRAMMING THE 8259A

The 8259A accepts two types of command words generated by the CPU:

1. *Initialization Command Words (ICWs)*: Before normal operation can begin, each 8259A in the system must be brought to a starting point — by a sequence of 2 to 4 bytes timed by WR pulses. This sequence is described in Figure 1.
2. *Operation Command Words (OCWs)*: These are the command words that are sent to the 8259A for various forms of operation, such as:
 - Interrupt Masking
 - End of Interrupt
 - Priority Rotation
 - Interrupt Status

The OCWs can be written into the 8259A anytime after initialization.

INITIALIZATION**GENERAL**

Whenever a command is issued with A0=0 and D4=1, this is interpreted as Initialization Command Word 1 (ICW1). ICW1 starts the initialization sequence during which the following automatically occur.

- a. The edge sense circuit is reset, which means that following initialization, an interrupt request (IR) input must make a low-to-high transition to generate an interrupt.
- b. The Interrupt Mask Register is cleared.
- c. R7 input is assigned priority 7.
- d. The slave mode address is set to 7.
- e. Special Mask Mode is cleared and Status Read is set to IRR.
- f. If IC4=0, then all functions selected in ICW4 are set to zero. (Non-Buffered mode*, no Auto-EOI, MCS-80/85 system).

*Note: Master/Slave in ICW4 is only used in the buffered mode.

A ₀	D ₄	D ₃	\overline{RD}	\overline{WR}	\overline{CS}	INPUT OPERATION (READ)
0			0	1	0	IRR, ISR or Interrupting Level → DATA BUS (Note 1) IMR → DATA BUS
1			0	1	0	
						OUTPUT OPERATION (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 — 3-STATE (NO OPERATION)
X	X	X	X	X	1	DATA BUS — 3-STATE (NO OPERATION)

- Notes:** 1. Selection of IRR, ISR or Interrupting Level is based on the content of OCW3 written before the READ operation.
2. On-chip sequencer logic queues these commands into proper sequence.

8259A Basic Operation

INITIALIZATION COMMAND WORDS 1 AND 2 (ICW1, ICW2)

A_5-A_{15} : Page starting address of service routines. In an MCS 80/85 system, the 8 request levels will generate CALLs to 8 locations equally spaced in memory. These can be programmed to be spaced at intervals of 4 or 8 memory locations, thus the 8 routines will occupy a page of 32 or 64 bytes, respectively.

The address format is 2 bytes long (A_0-A_{15}). When the routine interval is 4, A_0-A_4 are automatically inserted by the 8259A, while A_5-A_{15} are programmed externally. When the routine interval is 8, A_0-A_5 are automatically inserted by the 8259A, while A_6-A_{15} are programmed externally.

The 8-byte interval will maintain compatibility with current software, while the 4-byte interval is best for a compact jump table.

In an MCS-86 system T7-T3 are inserted in the five most significant bits of the vectoring byte and the 8259A sets the three least significant bits according to the interrupt level. $A_{10}-A_5$ are ignored and ADI (Address Interval) has no effect.

LTIM: If $LTIM = 1$, then the 8259A will operate in the level interrupt mode. Edge detect logic on the interrupt inputs will be disabled.

ADI: CALL address interval. $ADI = 1$ then interval = 4; $ADI = 0$ then interval = 8.

SNGL: Single. Means that this is the only 8259A in the system. If $SNGL = 1$ no ICW3 will be issued.

IC4: If this bit is set — ICW4 has to be read. If ICW4 is not needed, set $IC4 = 0$.

INITIALIZATION COMMAND WORD 3 (ICW3)

This word is read only when there is more than one 8259A in the system and cascading is used, in which case $SNGL = 0$. It will load the 8-bit slave register. The functions of this register are:

- In the master mode (either when $\overline{SP} = 1$, or in buffered mode when $M/S = 1$ in ICW4) a "1" is set for each slave in the system. The master then will release byte 1 of the call sequence (for MCS-80/85 system) and will enable the corresponding slave to release bytes 2 and 3 (for 8086/8088 only byte 2) through the cascade lines.
- In the slave mode (either when $\overline{SP} = 0$, or if $BUF = 1$ and $M/S = 0$ in ICW4) bits 2-0 identify the slave. The slave compares its cascade input with these bits and if they are equal, bytes 2 and 3 of the call sequence (or just byte 2 for 8086/8088) are released by it on the Data Bus.

INITIALIZATION COMMAND WORD 4 (ICW4)

SFNM: If $SFNM = 1$ the special fully nested mode is programmed.

BUF: If $BUF = 1$ the buffered mode is programmed. In buffered mode $\overline{SP}/\overline{EN}$ becomes an enable output and the master/slave determination is by M/S .

M/S: If buffered mode is selected: $M/S = 1$ means the 8259A is programmed to be a master, $M/S = 0$ means the 8259A is programmed to be a slave. If $BUF = 0$, M/S has no function.

AEOI: If $AEOI = 1$ the automatic end of interrupt mode is programmed.

μ PM: Microprocessor mode: μ PM = 0 sets the 8259A for MCS-80/85 system operation, μ PM = 1 sets the 8259A for MCS-86 system operation.

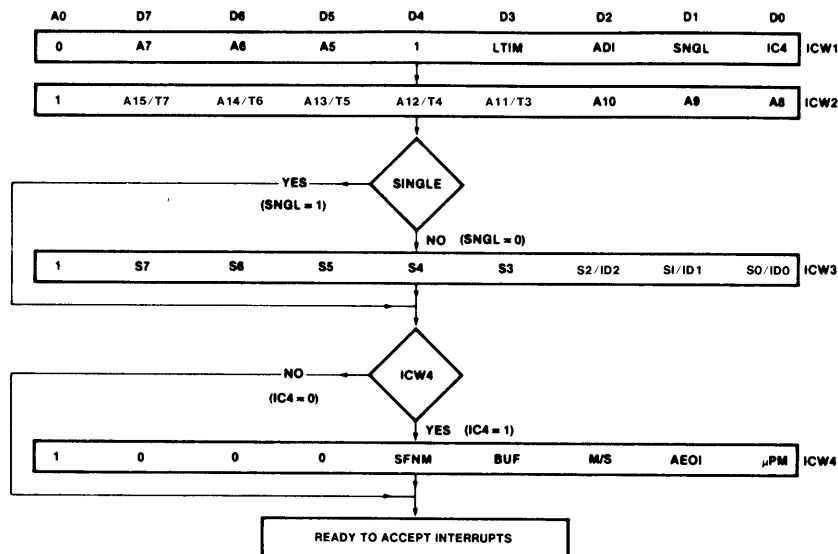
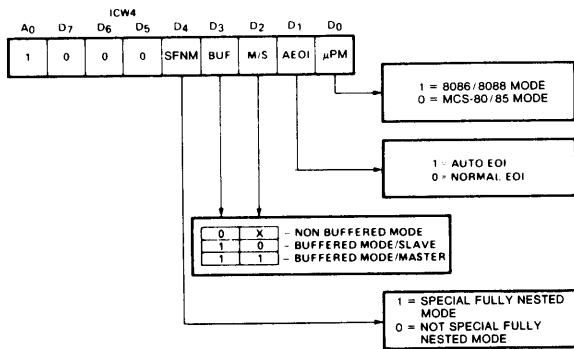
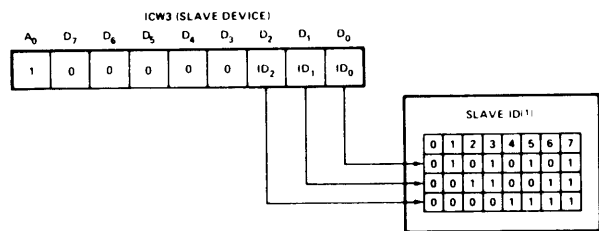
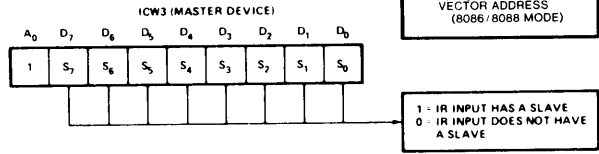
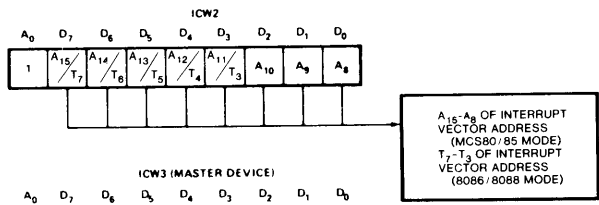
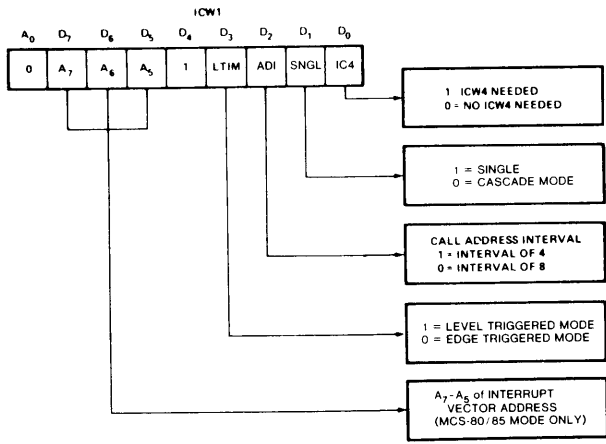


Figure 1. Initialization Sequence

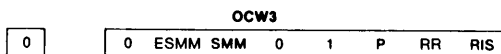
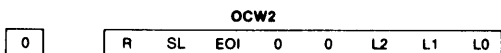
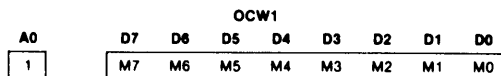


NOTE 1: SLAVE ID IS EQUAL TO THE CORRESPONDING MASTER IR INPUT.

Initialization Command Word Format

OPERATION COMMAND WORDS (OCWs)

After the Initialization Command Words (ICWs) are programmed into the 8259A, the chip is ready to accept interrupt requests at its input lines. However, during the 8259A operation, a selection of algorithms can command the 8259A to operate in various modes through the Operation Command Words (OCWs).

OPERATION CONTROL WORDS (OCWs)**OPERATION CONTROL WORD 1 (OCW1)**

OCW1 sets and clears the mask bits in the interrupt Mask Register (IMR). $M_7 - M_0$ represent the eight mask bits. $M = 1$ indicates the channel is masked (inhibited), $M = 0$ indicates the channel is enabled.

OPERATION CONTROL WORD 2 (OCW2)

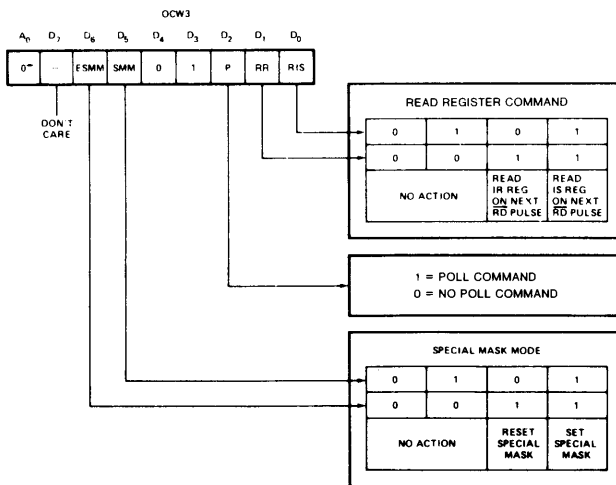
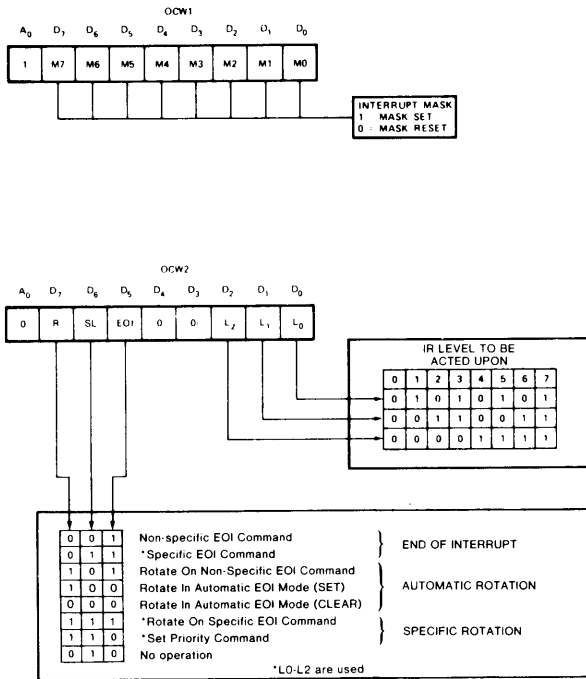
R, SL, EOI — These three bits control the Rotate and End if Interrupt modes and combinations of the two. A chart of these combinations can be found on the Operation Command Word Format.

L_2, L_1, L_0 — These bits determine the interrupt level acted upon when the SEOI bit is active.

OPERATION CONTROL WORD 3 (OCW3)

ESMM — Enable Special Mask Mode. When this bit is set to 1 it enables the SMM bit to set or reset the Special Mask Mode. When $ESMM = 0$ the SMM bit becomes a "don't care".

SMM — Special Mask Mode. If $ESMM = 1$ and $SMM = 1$ the 8259A will enter Special Mask Mode. If $ESMM = 1$ and $SMM = 0$ the 8259A will revert to normal mask mode. When $ESMM = 0$, SMM has no effect.



Operation Command Word Format

INTERRUPT MASKS

Each Interrupt Request input can be masked individually by the Interrupt Mask Register (IMR) programmed through OCW1. Each bit in the IMR masks one interrupt channel if it is set (1). Bit 0 masks IR0, Bit 1 masks IR1 and so forth. Masking an IR channel does not affect the other channels operation.

SPECIAL MASK MODE

Some applications may require an interrupt service routine to dynamically alter the system priority structure during its execution under software control. For example, the routine may wish to inhibit lower priority requests for a portion of its execution but enable some of them for another portion.

The difficulty here is that if an Interrupt Request is acknowledged and an End of Interrupt command did not reset its IS bit (i.e., while executing a service routine), the 8259A would have inhibited all lower priority requests with no easy way for the routine to enable them

That is where the Special Mask Mode comes in. In the special Mask Mode, when a mask bit is set in OCW1, it inhibits further interrupts at that level *and enables* interrupts from *all other* levels (lower as well as higher) that are not masked.

Thus, any interrupts may be selectively enabled by loading the mask register.

The special Mask Mode is set by OCW3 where: SMM = 1, SMM = 1, and cleared where SMM = 1, SMM = 0.

BUFFERED MODE

When the 8259A is used in a large system where bus driving buffers are required on the data bus and the cascading mode is used, there exists the problem of enabling buffers.

The buffered mode will structure the 8259A to send an enable signal on $\overline{SP/\overline{EN}}$ to enable the buffers. In this mode, whenever the 8259A's data bus outputs are enabled, the $\overline{SP/\overline{EN}}$ output becomes active.

This modification forces the use of software programming to determine whether the 8259A is a master or a slave. Bit 3 in ICW4 programs the buffered mode, and bit 2 in ICW4 determines whether it is a master or a slave.

FULLY NESTED MODE

This mode is entered after initialization unless another mode is programmed. The interrupt requests are ordered in priority form 0 through 7 (0 highest). When an interrupt is acknowledged the highest priority request is determined and its vector placed on the bus. Additionally, a bit of the Interrupt Service register (ISO-7) is set. This bit remains set until the microprocessor issues an End of Interrupt (EOI) command immediately before returning from the service routine, or if AEOI (Automatic End of Interrupt) bit is set, until the trailing edge of the last INTA. While the IS bit is set, all further interrupts of the same or lower priority are inhibited, while higher levels will generate an interrupt (which will be acknowledged only if the microprocessor internal Interrupt enable flip-flop has been re-enabled through software).

After the initialization sequence, IR0 has the highest priority and IR7 the lowest. Priorities can be changed, as will be explained, by priority rotation.

THE SPECIAL FULLY NESTED MODE

This mode will be used in the case of a big system where cascading is used, and the priority has to be conserved within each slave. In this case the special fully nested mode will be programmed to the master (using ICW4). This mode is similar to the normal fully nested mode with the following exceptions:

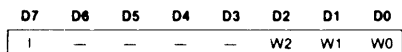
- a. When an interrupt request from a certain slave is in service this slave is not locked out from the master's priority logic and further interrupt requests from higher priority IR's within the slave will be recognized by the master and will initiate interrupts to the processor. (In the normal nested mode a slave is masked out when its request is in service and no higher requests from the same slave can be serviced.)
- b. When exiting the Interrupt Service routine the software has to check whether the interrupt serviced was the only one from that slave. This is done by sending a non-specific End of Interrupt (EOI) command to the slave and then reading its In-Service register and checking for zero. If it is empty, a non-specific EOI can be sent to the master too. If not, no EOI should be sent.

POLL

In this mode the microprocessor internal Interrupt Enable flip-flop is reset, disabling its interrupt input. Service to devices is achieved by programmer initiative using a Poll command.

The Poll command is issued by setting P = "1" in OCW3. The 8259A treats the next \overline{RD} pulse to the 8259A (i.e., $\overline{RD} = 0, \overline{CS} = 0$) as an interrupt acknowledge, sets the appropriate IS bit if there is a request, and reads the priority level. Interrupt is frozen from \overline{WR} to \overline{RD} .

The word enabled onto the data bus during \overline{RD} is:



W0-W2: Binary code of the highest priority level requesting service.

1: Equal to a "1" if there is an interrupt.

This mode is useful if there is a routine command common to several levels so that the \overline{INTA} sequence is not needed (saves ROM space). Another application is to use the poll command to expand the number of priority levels to more than 64.

END OF INTERRUPT (EOI)

The In Service (IS) bit can be reset either automatically following the trailing edge of the last in sequence \overline{INTA} pulse (when AEOI bit in ICW1 is set) or by a command word that must be issued to the 8259A before returning from a service routine (EOI command). An EOI command must be issued twice, once for themaster and once for the corresponding slave if slaves are in use.

There are two forms of EOI command: Specific and Non-Specific. When the 8259A is operated in modes which preserve the fully nested structure, it can determine which IS bit to reset on EOI. When a Non-Specific EOI command is issued the 8259A will automatically reset the highest IS bit of those that are set, since in the nested mode the highest IS level was necessarily the last level acknowledged and serviced.

However, when a mode is used which may disturb the fully nested structure, the 8259A may no longer be able to determine the last level acknowledged. In this case a Specific End of Interrupt (SEOI) must be issued which includes as part of the command the IS level to be reset. EOI is issued whenever $EOI = 1$, in OCW2, where L0-L2 is the binary level of the IS bit to be reset. Note that although the Rotate command can be issued together with an EOI where $EOI = 1$, it is not necessarily tied to it.

It should be noted that an IS bit that is masked by an IMR bit will not be cleared by a non-specific EOI if the 8259A is in the Special Mask Mode.

AUTOMATIC END OF INTERRUPT (AEOI) MODE

If $AEOI = 1$ in ICW4, then the 8259A will operate in AEOI mode continuously until reprogrammed by ICW4. In this mode the 8259A will automatically perform a non-specific EOI operation at the trailing edge of the last interrupt acknowledge pulse (third pulse in MCS-80/85,

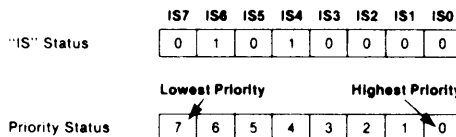
second in MCS-86). Note that from a system standpoint, this mode should be used only when a nested multilevel interrupt structure is not required within a single 8259A.

To achieve automatic rotation within AEOI, there is a special rotate flip-flop. It is set by OCW2 with $R = 1, SL = 0, EOI = 0$, and cleared with $R = 0, SEOI = 0, EOI = 0$.

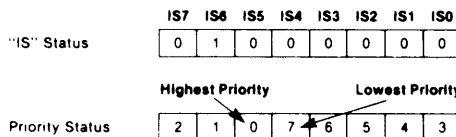
AUTOMATIC ROTATION (Equal Priority Devices)

In some applications there are a number of interrupting devices of equal priority. In this mode a device, after being serviced, receives the lowest priority, so a device requesting an interrupt will have to wait, in the worst case until each of 7 other devices are serviced at most once. For example, if the priority and "in service" status is:

Before Rotate (IR4 the highest priority requiring service)



After Rotate (IR4 was serviced, all other priorities rotated correspondingly)



There are two ways to accomplish Automatic Rotation using OCW2, the Rotate on Non-Specific EOI Command ($R = 1, SL = 0, EOI = 1$) and the Rotate in Automatic EOI Mode which is set by ($R = 1, SL = 0, EOI = 0$) and cleared by ($R = 0, SL = 0, EOI = 0$).

SPECIFIC ROTATION (Specific Priority)

The programmer can change priorities by programming the bottom priority and thus fixing all other priorities; i.e., if IR5 is programmed as the bottom priority device, then IR6 will have the highest one.

The Set Priority command is issued in OCW2 where: $R = 1, SEOI = 1$; L0-L2 is the binary priority level code of the bottom priority device.

Observe that in this mode internal status is updated by software control during OCW2. However, it is independent of the End of Interrupt (EOI) command (also executed by OCW2). Priority changes can be executed during an EOI command by using the Rotate on Specific EOI Command in OCW2 ($R = 1, SL = 1, EOI = 1$ and L0-L2 = IR level to receive bottom priority).

