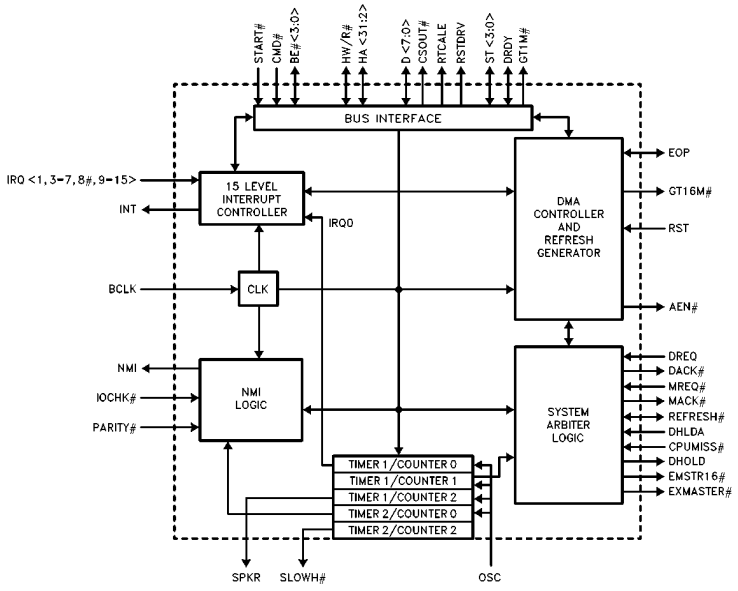
PRELIMINARY

82357
INTEGRATED SYSTEM PERIPHERAL (ISP)

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

  (See Packaging Specifications: Order Number 240800, Package Type NG)

82357 Internal Block Diagram



290253–78

Intel486 is a trademark of Intel Corporation.

March 1994

Order Number: 290253-006

82357 Integrated System Peripheral (ISP)

CONTENTS

PAGE

CONTENTS

PAGE

CONTENTS

PAGE

CONTENTS

PAGE

CONTENTS

PAGE

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

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

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

1.0 ISP SYSTEM INTERFACE ILLUSTRATION

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

2.0 FUNCTIONAL OVERVIEW

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

2.1 Master and Slave Modes

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

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

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

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

2.2 DMA Controller

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

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

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

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

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

2.3 System Arbiter

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

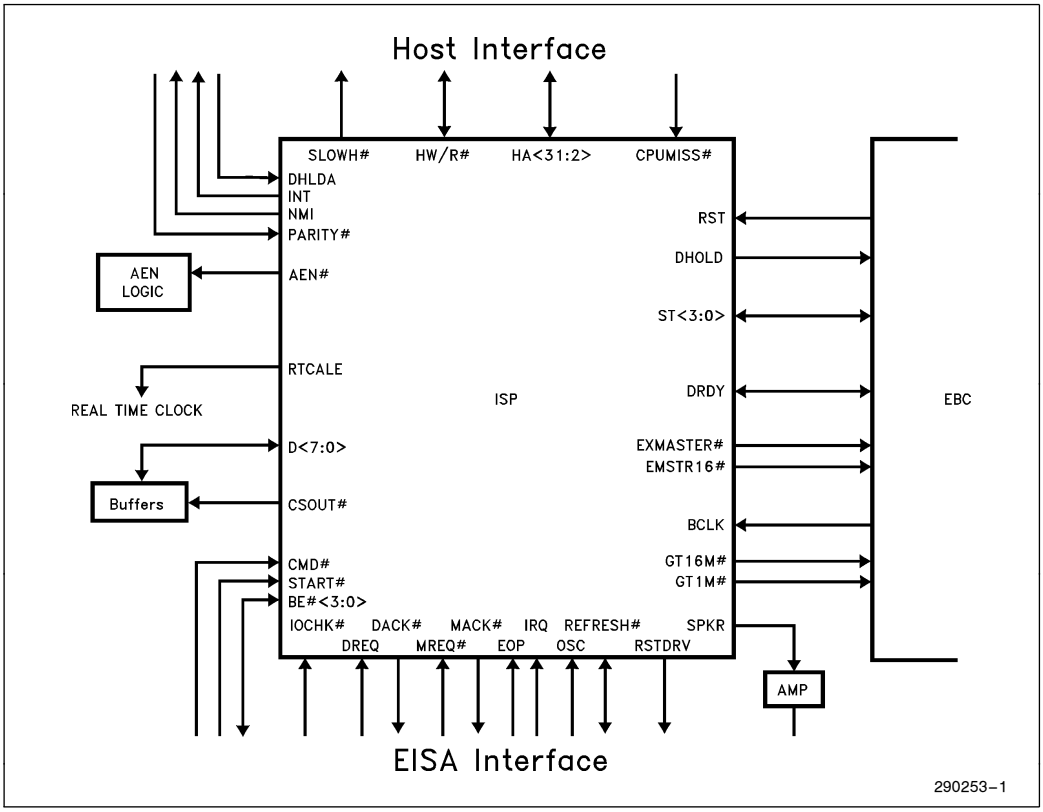


Figure 1-1. ISP System Interface

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

The DMA channels, when used by 16-bit ISA masters for direct access to the bus, must be placed in Cascade mode. In Cascade mode, the DMA controller will respond to DREQ with DACK# and EMSTR16#, but HW/R#, ST0, ST1, A<31:2>, and BE<3:0># outputs will be disabled.

2.3.1 PRIORITIES OF REQUESTERS

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

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

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

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

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

2.3.2 Preemption of EISA Masters and DMA Devices

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

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

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

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

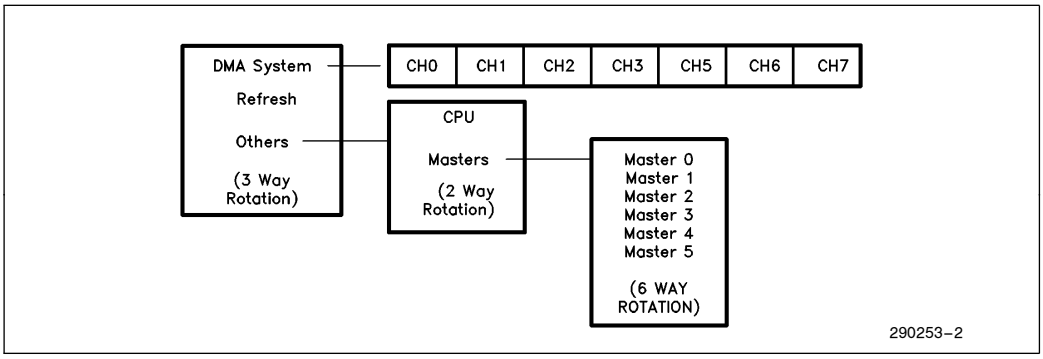


Figure 2-1. Arbitration Priority

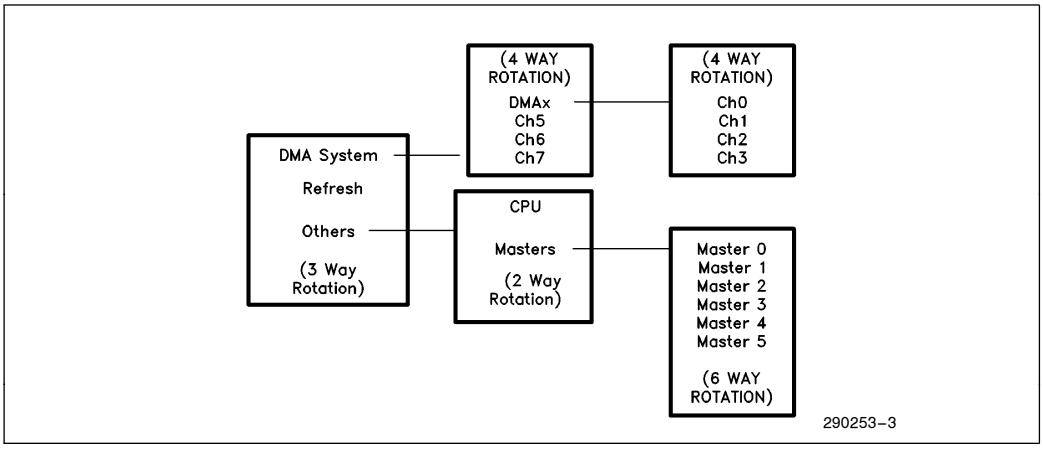


Figure 2-2. DMA Channels in Rotating Priority

2.3.3 ARBITRATION DURING NON-MASKABLE INTERRUPTS

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

2.3.4 DREQ AND DACK LATENCY CONTROL

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

2.4 Refresh Generation

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

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

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

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

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

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

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

2.5 Interrupt Controller

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

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

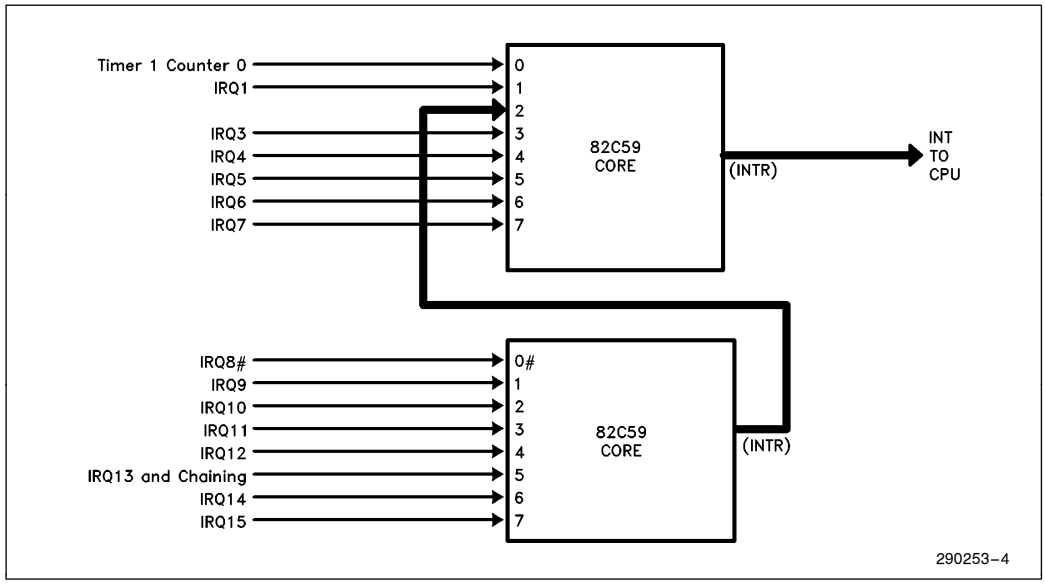


Figure 2-3. Block Diagram of the Interrupt Controller

2.6 Non-Maskable Interrupt (NMI)

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

NMI interrupts are caused by the following conditions:

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

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

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

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

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

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

NMI Source Enable/Disable and Status Port Bits

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

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

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

2.7 Interval Timers

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

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

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

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

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

Figure 2-4 lists the Interval Timer Functions.

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

Figure 2-4. Interval Timer Functions

2.8 Register Access

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

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

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

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

3.0 DMA CONTROLLER

3.1 DMA Transfer Modes

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

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

3.1.1 SINGLE TRANSFER MODE

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

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

3.1.2 BLOCK TRANSFER MODE

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

3.1.3 DEMAND TRANSFER MODE

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

3.1.4 CASCADE MODE

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

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

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

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

3.2 Transfer Types

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

3.2.1 WRITE TRANSFER

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

3.2.2 READ TRANSFER

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

3.2.3 VERIFY TRANSFER

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

3.3 Autoinitialize

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

3.4 Channel Priority

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

3.4.1 FIXED PRIORITY

The initial fixed priority structure is as follows:

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

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

3.4.2 ROTATING PRIORITY

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

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

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

Figure 3-1 will demonstrate rotation:

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

Figure 3-1

3.5 Buffer Chaining

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

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

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

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

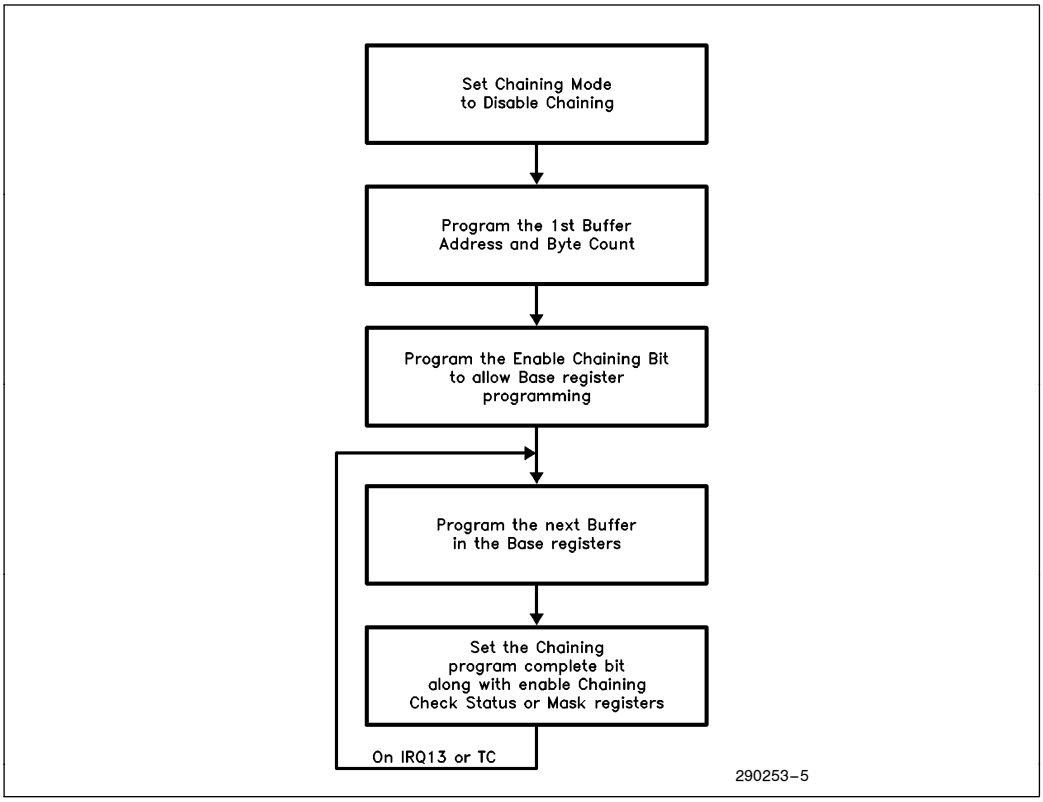


Figure 3-2. Buffer Chaining Programming Flow

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

3.6 DMA Enhanced Timing

3.6.1 COMPATIBLE TIMING

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

3.6.2 TYPE ''A'' TIMING

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

3.6.3 TYPE ''B'' TIMING

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

3.6.4 TYPE ''C'' (BURST TIMING)

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

3.7 Register Description

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

3.7.1. STOP REGISTERS (RING BUFFER DATA STRUCTURE)

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

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

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

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

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

NOTE:
I/O writes must match the I/O slave size (i.e., 8-bit writes must be used to program the ISP registers). When writing to the DMA registers, the DMA channels also must be masked.

For example:

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

Table 3-1

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

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

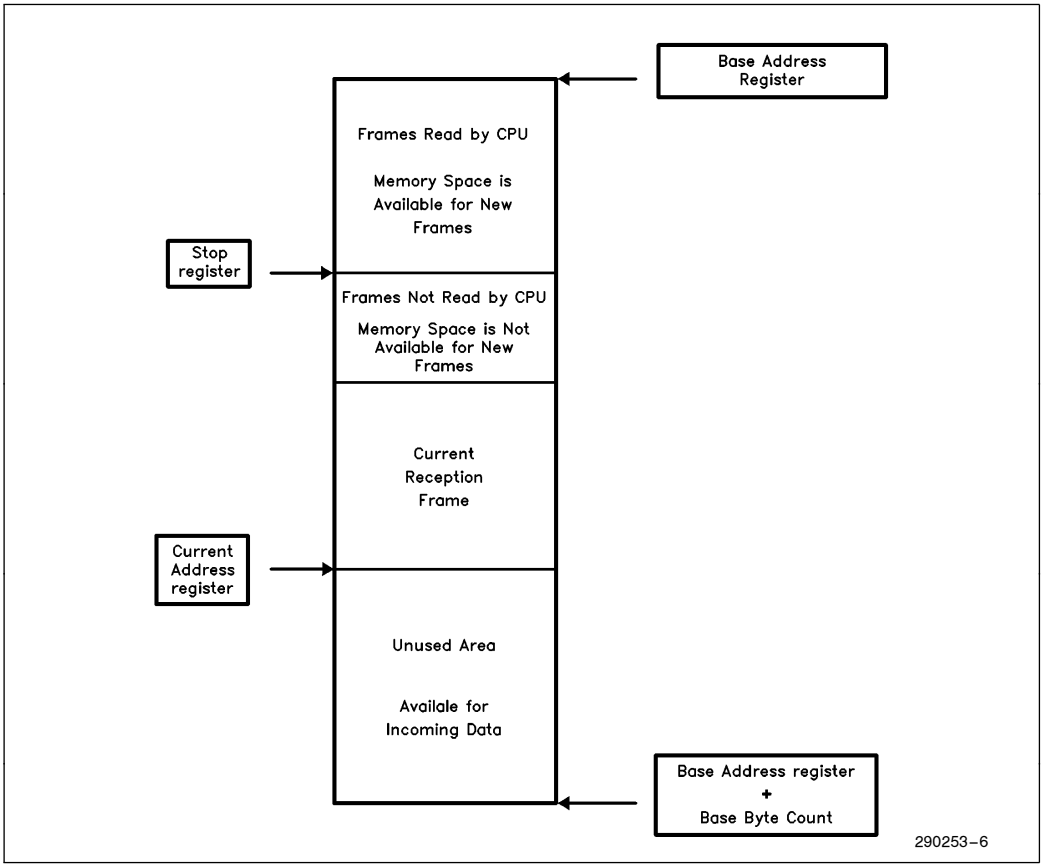


Figure 3-3

3.7.2 DMA MEMORY LOW PAGE REGISTER (READ/WRITE)

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

3.7.3 DMA MEMORY HIGH PAGE REGISTER (READ/WRITE)

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

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

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

3.7.3.1 Address Compatibility Mode

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

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

3.7.4 CURRENT ADDRESS REGISTER (READ/WRITE)

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

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

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

Table 3-2

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

* PC AT is a trademark of IBM.

3.7.5 CURRENT WORD REGISTER (READ/WRITE)

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

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

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

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

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

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

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

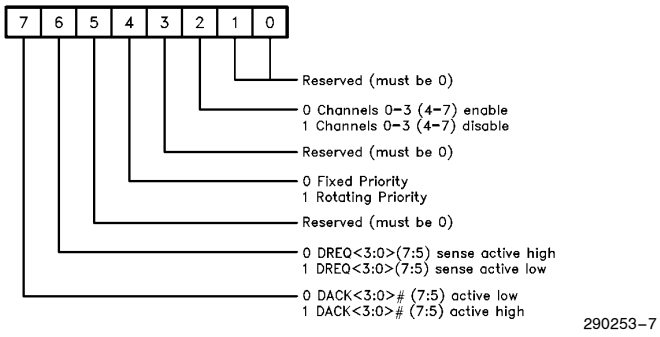
Each channel has a set of Base Page, Base Address and Base Word Count registers. These registers store the original value of their associated Current registers. During Autoinitialize these values are used to restore the Current registers to their original values. The Base registers are written simultaneously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

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

3.7.7 COMMAND REGISTER (WRITE ONLY)

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

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



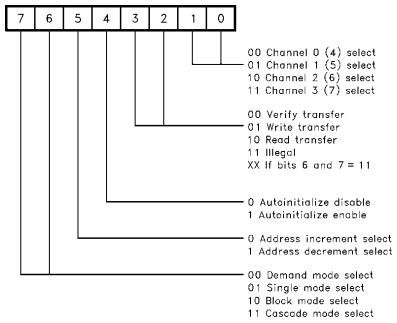
NOTE:
Disabling channels 4–7 will also disable channels 0–3, as channels 0–3 are logically cascaded into channel 4.

Figure 3-4. Command Register (Write Only)

3.7.8 MODE REGISTER (WRITE ONLY)

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

Channels 0–3 port address—0Bh
Channels 4–7 port address—0D6h



290253–8

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

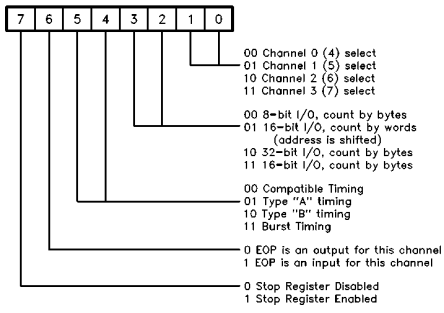
Figure 3-5. Mode Register (Write Only)

3.7.9 EXTENDED MODE REGISTERS (WRITE ONLY)

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

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

Channels 0–3 port address—040Bh
Channels 4–7 port address—04D6h



290253–9

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

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

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

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

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

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

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

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

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

3.7.9.5 EOP Input/Output Selection

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

3.7.9.6 Stop Register Selection

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

3.7.9.7 Summary of the DMA Transfer Sizes

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

Table 3-3

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

3.7.10 REQUEST REGISTER (WRITE ONLY)

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

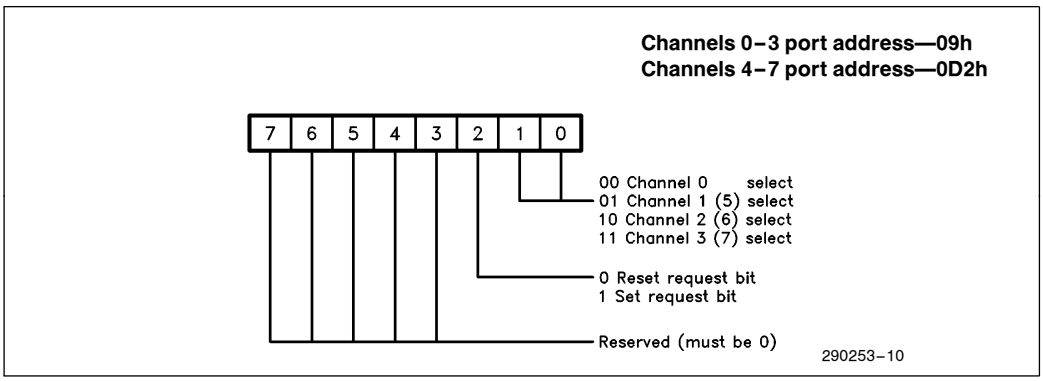


Figure 3-7. Request Register (Write Only)

3.7.11 MASK REGISTER

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

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

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

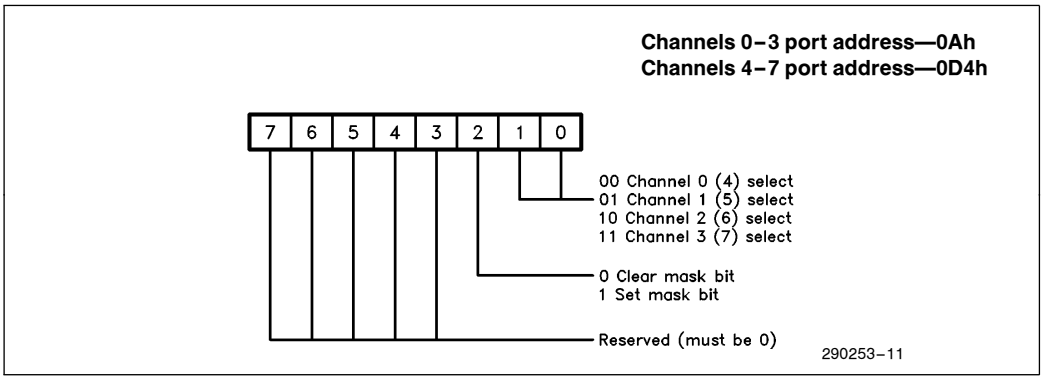


Figure 3-8. Write Single Mask Bit (Write Only)

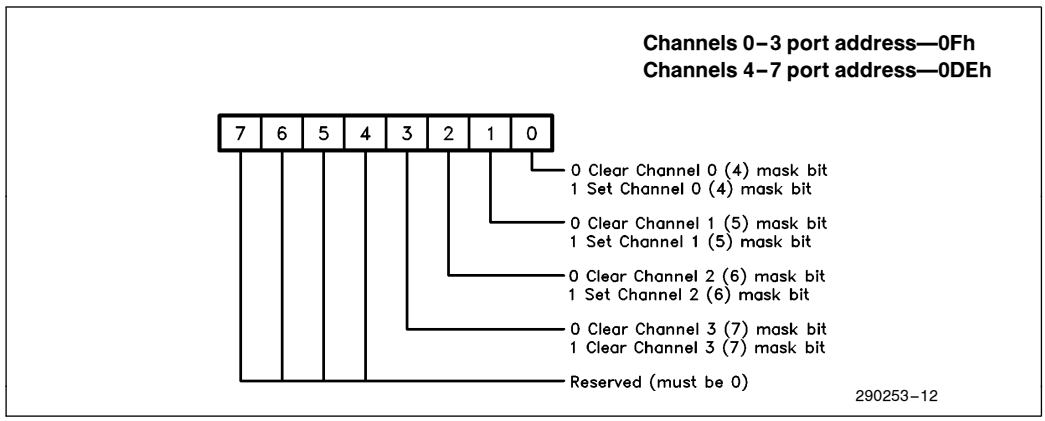
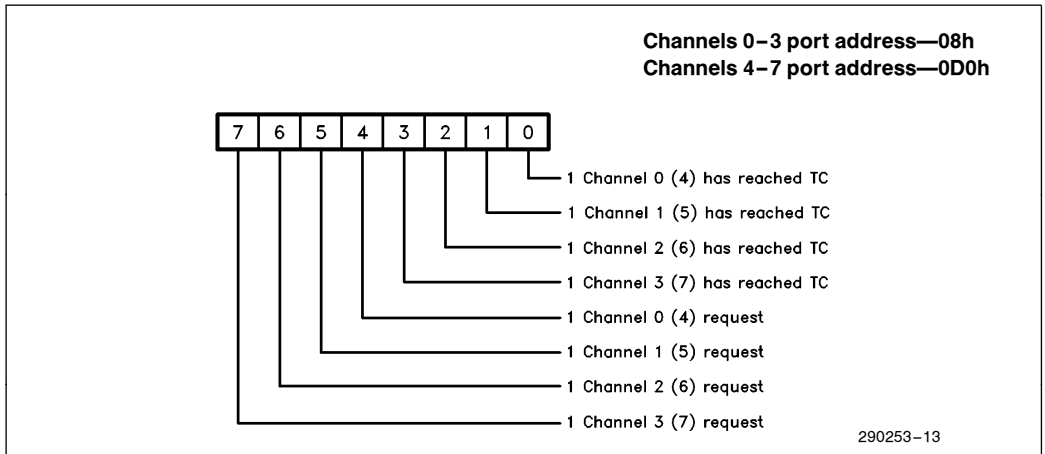


Figure 3-9. Write All Mask Register Bits (Read/Write)

3.7.12 STATUS REGISTER (READ ONLY)

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



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

Figure 3-10. Status Register (Read Only)

3.7.13 SET CHAINING MODE REGISTER (WRITE ONLY)

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

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

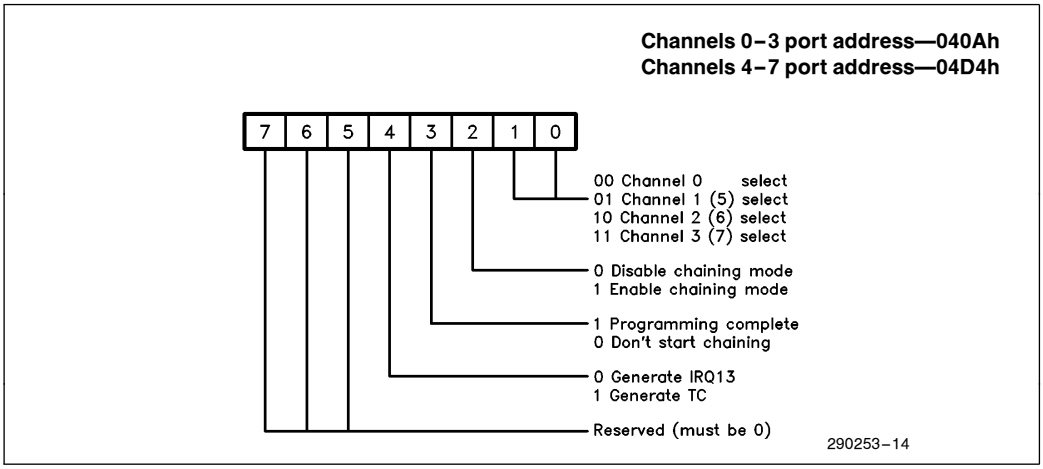


Figure 3-11. Set Chaining Mode Register (Write Only)

The **Enable Chaining mode** bit is used to control the chaining mode logic. If the bit is programmed to a 1 after the initial DMA address and count are programmed, then the Base address and Count registers will become available for programming the next chain buffer.

After the Base registers are programmed (as indicated above), the **Enable Chaining mode** and the **Programming Complete** bits are set to begin a DMA chaining sequence. The DMA channel is then ready to begin transferring data (assuming the mask bit is cleared).

When a chaining mode interrupt or TC (for EISA programming masters) occurs, the next Address and Count registers should be programmed and the **Programming Complete** bit should be set to set up for the next transfer. When the Programming Complete bit is set, the Enable Chaining Mode bit and the Generate IRQ bit both need to be written to the correct state. Upon this programming the interrupt request for that channel is reset, if it was active.

Bit 4 of the Set Chaining Mode register is used to determine the response to the expiration of a DMA buffer. Normally the Host CPU needs to be informed to program the next set of Base registers. In this case bit 4 should be set to a zero to generate an IRQ13. If an EISA bus master is using the DMA to assist in data transfer, then bit 4 can be set to a 1 to generate an EOP(TC) instead of an IRQ13. The EISA master can then use the EOP(TC) in the same way that the CPU uses an interrupt. In this mode the EOP signal will only be driven active while the Channel that caused it is running as determined by the DACK lines.

3.7.14 SET CHAINING MODE STATUS REGISTER (READ ONLY)

This register is read only and is used to determine if Chaining mode for a particular channel is enabled or disabled. A ''1'' read in this register indicates that the channel's chaining mode is enabled. A ''0'' indicates that the chaining mode is disabled. All Chaining mode bits are disabled after a reset. After the DMA is used in Chaining mode the CPU will need to clear the Chaining mode enable bit if non-Chaining mode is desired. This bit is programmed in bit 2 of the Set Chaining Mode register. (See Figure 3-12.)

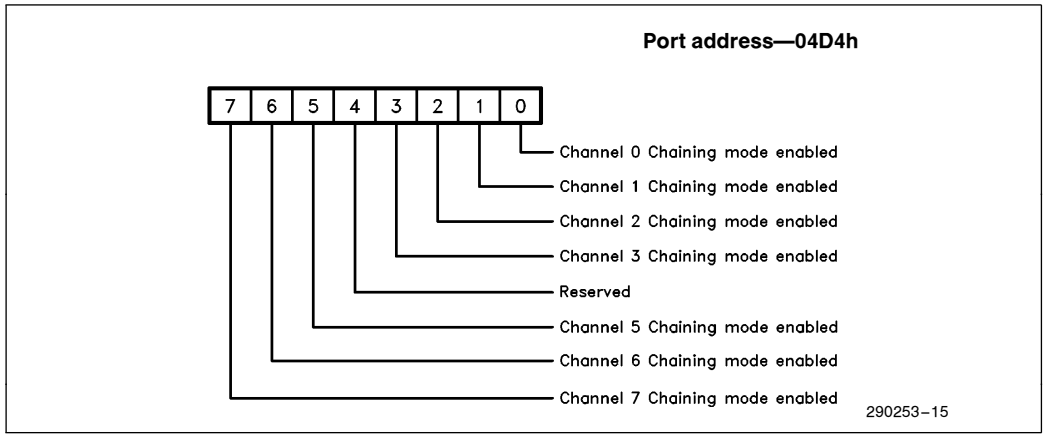


Figure 3-12. Set Chaining Mode Status Register (Read Only)

3.7.15 CHANNEL INTERRUPT STATUS REGISTER (READ ONLY)

Channel Interrupt Status is a read only register and is used to indicate the source (channel) of a DMA chaining interrupt on IRQ13. The DMA controller drives IRQ13 active after reaching terminal count, with chaining mode enabled. It does not drive IRQ13 active during the initial programming sequence that loads the Base registers. (See Figure 3-13.)

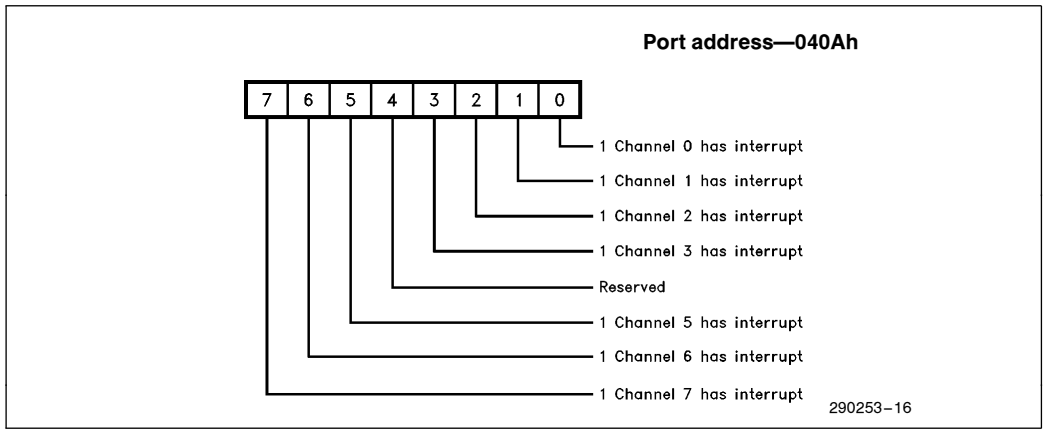


Figure 3-13. Channel Interrupt Status Register (Read Only)

3.7.16 CHAIN BUFFER EXPIRATION CONTROL REGISTER (READ ONLY)

This register is read only and reflects the outcome of the expiration of a chain buffer. If a channel bit is set to a 0, IRQ13 will be activated, otherwise a TC will be issued. This bit is programmed in bit 4 of the Set Chaining Mode register. (See Figure 3-14.)

3.8 Software Commands

These are additional special software commands which can be executed in the Program Condition. They do not depend on any specific bit pattern on the data bus. The three software commands are: (1) Clear Byte Pointer Flip-Flop (2) Master Clear and (3) Clear Mask Register.

3.8.1 CLEAR BYTE POINTER FLIP-FLOP

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

When the Host CPU is reading or writing DMA registers, two Byte Pointer Flip-Flops are used; one for channels 0–3 and one for channels 4–7. Both of these act independently. There are separate software commands for clearing each of them (0Ch for Channels 0–3, 0D8h for Channels 4–7).

An additional Byte Pointer Flip-Flop has been added for use when EISA masters are reading and writing DMA registers. (The arbiter state will be used to determine the current master of the bus.) This Flip-Flop is cleared when an EISA Master performs a write to either 0Ch or 0D8h, there is only one for all eight DMA channels. This new Byte Pointer was added to eliminate the problem of the Host CPU's byte pointer getting out of sync if an EISA Master takes the bus during the Host CPU's DMA programming.

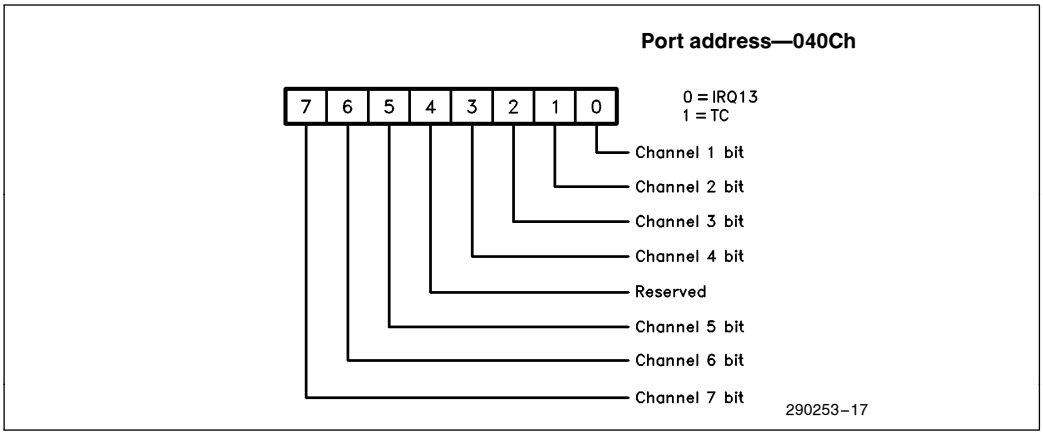


Figure 3-14. Chain Buffer Expiration Control Register (Read Only)

3.8.2 MASTER CLEAR

This software instruction has the same effect as the hardware Reset. The Command, Status, Request, and Internal First/Last Flip-Flop registers are cleared and the Mask register is set. The DMA controller will enter the idle cycle.

There are two independent Master Clear Commands, 0Dh which acts on channels 0–3, and 0DAh which acts on channels 4–7.

3.8.3 CLEAR MASK REGISTER

This command clears the mask bits of all four channels, enabling them to accept DMA requests. I/O port 0Eh is used for Channels 0–3 and I/O port 0DCh is used for Channels 4–7.

3.9 Terminal Count/EOP Summary

Table 3-4 summarizes the events that will happen as a result of a terminal count or external EOP when running DMA cycles in various modes.

Table 3-4. Terminal Count/EOP Summary

Event								
Word Counter Expired	Yes	X	Yes	X	Yes	X	X	X
Stop Reg. Limit Reached	X	X	X	X	X	X	Yes	Yes
EOP Input	X	Asserted	X	Asserted	X	Asserted	X	X
Conditions								
AUTOINIT	No	No	Yes	Yes	No	No	X	X
Chain and Base Loaded	No	No	X	X	Yes	Yes	X	X
Result								
Status TC	set	set	set	set	—	—	—	—
Mask	set	set	—	—	—	—	set	set
SW Request	clr	clr	clr	clr	—	—	—	—
Current Reg.	—	—	load	load	load	load	—	—

(Load = load current from base, — no change, X = Don't Care)

4.0 BUS ARBITRATION

4.1 Bus Timeout

A bus timeout will cause an NMI to the CPU and will activate RSTDRV.

NOTE:
A bus timeout will not occur during the time the system CPU has control of the bus.

4.1.1 8 μs BUS TIMEOUT

An 8 μs bus timeout is determined by counting 64 BCLKs, beginning with the rising edge of BCLK after a MACK# is negated. After the 64 BCLKs (on the rising edge of BCLK), MREQ# and ST3 (CIP#) are sampled. If MREQ# is still active, a bus timeout will occur. Note that MREQ# can be inactive and the bus is still owned by the master until ST3 (bus cycle in progress) goes inactive. The Arbiter will wait until ST3 (bus cycle in progress) goes inactive before granting the bus to another requester.

Figure 4-1 illustrates the longest that MREQ# can be held active without causing a Bus Time out. It also shows when the last bus master cycle must begin with a START# pulse, which in turn will activate ST3 (CIP#).

4.1.2 32 μs BUS TIMEOUT

To prevent the case of a slave from locking up the system when it does not release EXRDY or CHRDY, a bus timeout will occur if CMD# is active for more than 256 BCLKs. The counting of the 256 BCLKs will be conditioned on a request for the bus from another source. This will allow Burst Cycles to run unimpeded if there are no pending requests. (Although there should be a request at least every 15 μs from Refresh). The 32 μs bus timeout is independent of the type of master on the bus.

4.2 4 μs Limit on DMA Transfers

If the ISP DMA is the master of the bus and another request for the bus is received by the Arbiter and the DMA is not doing compatible timing transfers, a 4 μs timer will be started. Upon the expiration of the 4 μs timer, the DACK will be inactivated after the current DMA cycle has completed. The bus will then be arbitrated for and granted to the highest priority requester.

The 4 μs timer is not used in compatible timing mode, it is also not used for 16-bit ISA masters cascaded through the DMA DREQ lines.

If the DMA channel that was preempted by the 4 μs timer was operating in Block mode, an internal bit will be set so that the channel will be arbitrated for again, independent of the state of DREQ. This bit is reset on RSTDRV.

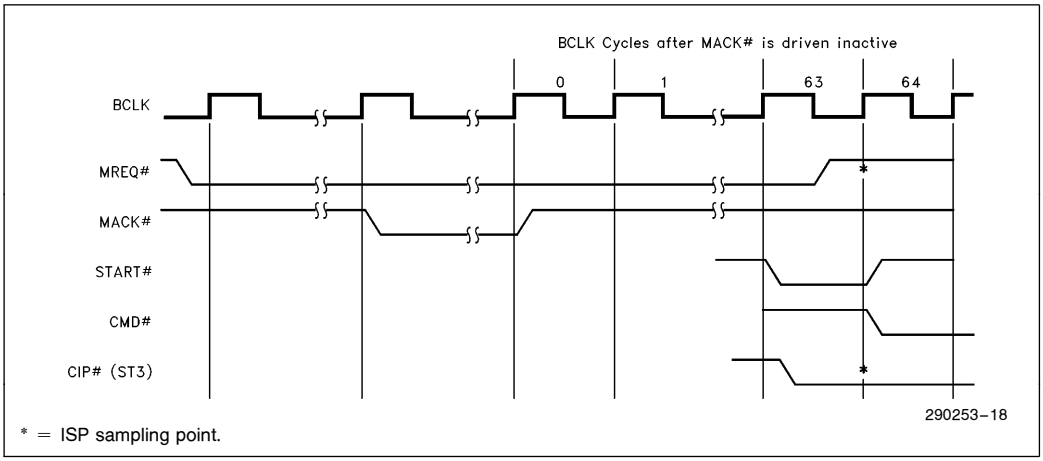


Figure 4-1. 8 μs Bus Timeout

4.3 EISA Bus Master Status Latch

In order to simplify testing of EISA bus master operations, a CPU readable status latch is included which contains information about which EISA bus master most recently had control of the bus. This latch is located at port address 0464h and is read only. (See Figure 4-2.) A read value of (0) indicates that the slot was most recently granted the bus.

An NMI service routine may read this latch to determine which Bus Master controlled the Bus when a Bus preempt Timeout occurred.

Port 0465h is reserved for an additional eight Bus Master status latch slots, it is not implemented on the ISP.

Note that the bits for slot 7 and 8 are driven by the ISP, though they will always be inactive (high).

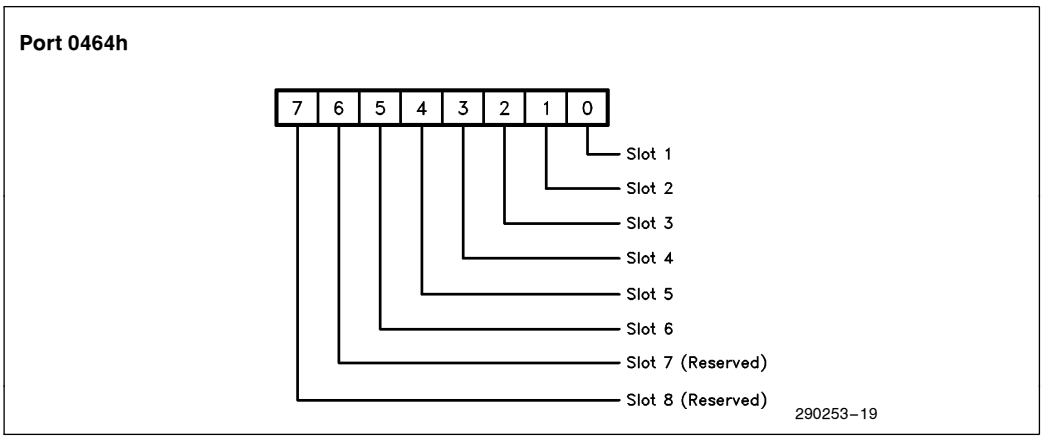


Figure 4-2. EISA Bus Master Status Latch

5.0 INTERRUPT CONTROLLER

The Interrupt Controller consists of two separate 82C59 cores. Interrupt Controller 1 (CNTRL-1) and Interrupt Controller 2 (CNTRL-2) are initialized separately, and can be programmed to operate in different modes. The default settings are: 80x86 Mode, Edge Sensitive (IRQ0–15), Normal EOI, Non-Buffered Mode, Special Fully Nested disabled, and Cascade Mode. CNTLR-1 is connected as the master Interrupt Controller and CNTRL-2 is connected as the slave Interrupt Controller.

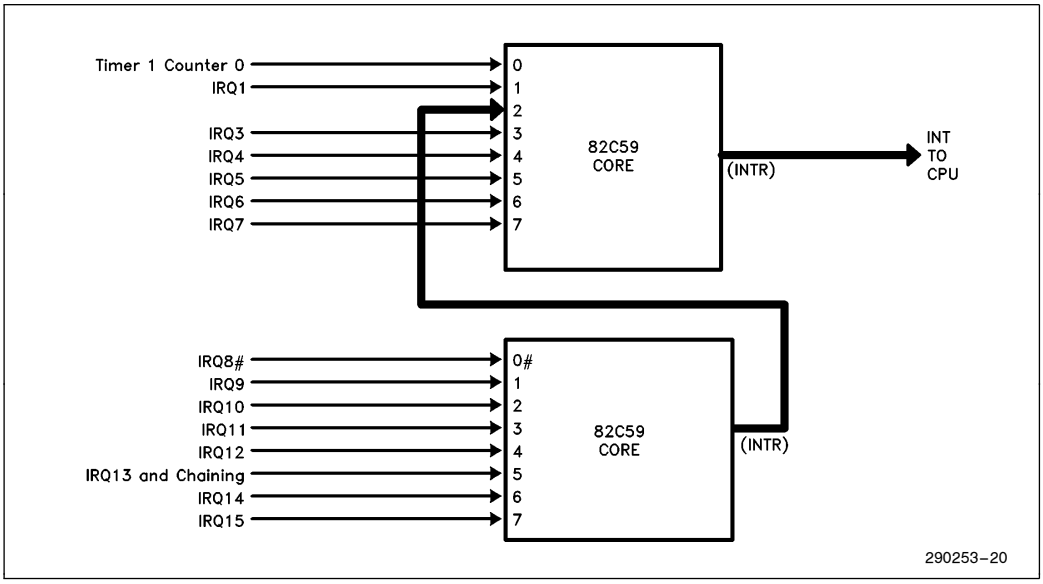


Figure 5-1. Interrupt Controller

5.1 Interrupt Controller I/O Address Map

Table 5-1 lists the I/O port address map for the interrupt registers.

Table 5-1

Interrupts	I/O Address	# of Bits	Register
IRQ<7:0>	0020h	8	CNTLR-1 Control Register
IRQ<7:0>	0021h	8	CNTRL-1 Mask Register
IRQ<7:0>	04D0h	8	CNTRL-1 Edge/Level Control Register
IRQ<15:8>	00A0h	8	CNTLR-2 Control Register
IRQ<15:8>	00A1h	8	CNTRL-2 Mask Register
IRQ<15:8>	04D1h	8	CNTRL-2 Edge/Level Register

5.2 Interrupt Definition Table

IRQ0 and IRQ2, illustrated in Table 5-2, are connected to the interrupt controllers internally.

Table 5-2

Priority	Label	Controller	Typical Interrupt Source
1	IRQ0	1	Interval Timer 1, Counter 0 OUT
2	IRQ1	1	Keyboard
3–10	IRQ2	1	Interrupt from Controller 2
3	IRQ8#	2	Real Time Clock
4	IRQ9	2	Expansion Bus Pin B04
5	IRQ10	2	Expansion Bus Pin D03
6	IRQ11	2	Expansion Bus Pin D04
7	IRQ12	2	Expansion Bus Pin D05
8	IRQ13	2	Coprocessor Error, Chaining
9	IRQ14	2	Fixed Disk Drive Controller Expansion Bus Pin D07
10	IRQ15	2	Expansion Bus Pin D06
11	IRQ3	1	Serial Port 2, Exp Bus B25
12	IRQ4	1	Serial Port 1, Exp Bus B24
13	IRQ5	1	Parallel Port 2, Exp Bus B23
14	IRQ6	1	Diskette Controller, Exp Bus B22
15	IRQ7	1	Parallel Port 1, Exp Bus B21

5.3 Interrupt Request Register (IRR) and In-Service Register (ISR)

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

5.4 Priority Resolver

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

5.5 Interrupt Mask Register (IMR)

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

5.6 INT (Interrupt)

This output goes directly to the CPU interrupt input.

5.7 INTA# /(ST2#) (Interrupt Acknowledge)

INTA# pulses will cause the Interrupt Controller system to release vectoring information onto the data bus. The format of this data depends on the system mode (μPM) of the Interrupt Controller (programmed for x86 mode in EISA systems). The ISP uses the ST2# input as the Interrupt Acknowledge line.

5.8 Interrupt Sequence

The powerful features of the Interrupt Controller in a microcomputer system are its programmability and the interrupt routine addressing capability. The latter allows direct or indirect jumping to the specific interrupt routine requested without any polling of the interrupting devices. The following shows the interrupt sequence for an x86 type system (8080 mode must never be selected by EISA software).

1. One or more of the INTERRUPT REQUEST lines are raised high, setting the corresponding IRR bit(s).
2. The Interrupt Controller evaluates these requests, and sends an INT to the CPU, if appropriate.
3. The CPU acknowledges the INT and responds with an INTA# pulse.
4. Upon receiving an INTA# from the CPU, the highest priority ISR bit is set and the corresponding IRR bit is reset. The Interrupt Controller does not drive the Data Bus during this cycle.
5. The CPU will initiate a second INTA# pulse. During this pulse, the Interrupt Controller releases an 8-bit pointer onto the Data Bus where it is read by the CPU.
6. This completes the interrupt cycle. In the AEOI mode the ISR bit is reset at the end of the second INTA# pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

If no interrupt request is present at step four (i.e., the request was too short in duration) the Interrupt Controller will issue an interrupt level 7.

5.9 80x86 Mode

In the x86 mode the processor produces only two Interrupt Acknowledge cycles. The Interrupt Controller uses the first interrupt acknowledge cycle to internally freeze the state of the interrupts for priority resolution. The first controller (CNTRL-1), as a master, issues the interrupt code on the cascade lines (internal to the ISP) at the end of the INTA# pulse. On this first cycle it does not issue any data to the processor and leaves its data bus buffers disabled. On the second interrupt acknowledge cycle, the master (CNTRL-1) or slave (CNTRL-2), will send a byte of data to the processor with the acknowledged interrupt code composed as in Table 5-3 (note the state of the ADI mode control is ignored and A5–A11 are unused in 80x86 mode).

Table 5-3. Content of Interrupt Vector Byte for 80x86 System Mode

	D7	D6	D5	D4	D3	D2	D1	D0
IRQ7,15	T7	T6	T5	T4	T3	1	1	1
IRQ6,14	T7	T6	T5	T4	T3	1	1	0
IRQ5,13	T7	T6	T5	T4	T3	1	0	1
IRQ4,12	T7	T6	T5	T4	T3	1	0	0
IRQ3,11	T7	T6	T5	T4	T3	0	1	1
IRQ2,10	T7	T6	T5	T4	T3	0	1	0
IRQ1,9	T7	T6	T5	T4	T3	0	0	1
IRQ0,8	T7	T6	T5	T4	T3	0	0	0

T7–T3 represent the interrupt vector address (refer to Section 5.11.1).

5.10 Programming the Interrupt Controller

The Interrupt Controller accepts two types of command words generated by the CPU or bus master:

1. Initialization Command Words (ICWs): Before normal operation can begin, each Interrupt Controller in the system must be brought to a starting point—by a sequence of two to four bytes timed by I/O write pulses.

An I/O write to CNTRL-1 or CNTRL-2 base address with D4 = 1 and A0 = 0, is interpreted as ICW1. For EISA systems, two I/O writes to ''base address + 1'' must follow the ICW1. The first write to ''base address + 1'' performs ICW2, the second write performs ICW3. A third write to ''base address + 1'' performs ICW4. The base address for CNTRL-1 is 020h, and the base address for CNTRL-2 is 0A0h.

ICW1 starts the initialization sequence during which the following automatically occur:

a) The edge sense circuit is reset, which means that following initialization, an interrupt request (IRQ) input must make a low-to-high transition to generate an interrupt.

b) The Interrupt Mask register is cleared.

c) IR7 input is assigned priority 7.

d) The slave mode address is set to 7.

e) Special Mask Mode is cleared and Status Read is set to IRR.

f) If IC4 = 0, then all functions selected in ICW4 are set to zero. (Non-Buffered mode*, no Auto-EOI, MCS-80, 85 systems).

2. Operation Command Words (OCWs): These are the command words which command the Interrupt Controller to operate in various interrupt modes. These modes are:

a) Fully nested mode

b) Rotating priority mode

c) Special mask mode

d) Polled mode

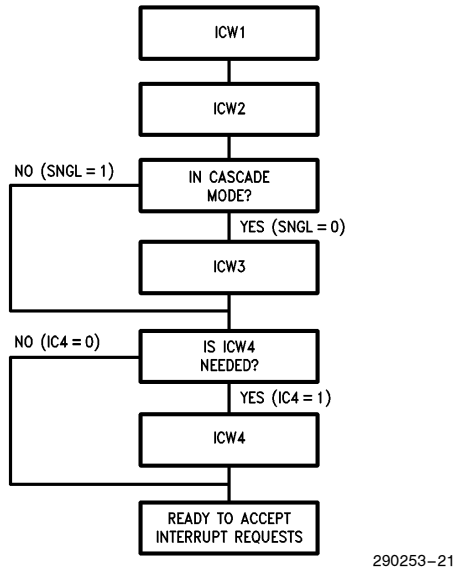
The OCWs can be written into the Interrupt Controller anytime after initialization.

The Base I/O address for CNTRL-1 is 020h; for CNTRL-2, 0A0h. Table 5-4 lists the initial values set-up at power-up by the BIOS.

Table 5-4. Initial Interrupt Controller Values

Port	Value	Description of Contents
020h	11h	CNTLR-1, ICW1
021h	08h	CNTLR-1, ICW2 Vector Address for 000020h
021h	04h	CNTLR-1, ICW3 Indicates Slave Connection
021h	01h	CNTLR-1, ICW4 8086 Mode
021h	B8h	CNTLR-1, Interrupt Mask (may vary)
04D0h	00h	CNTRL-1, Edge/Level Control Register
0A0h	11h	CNTLR-2, ICW1
0A1h	70h	CNTLR-2, ICW2 Vector Address for 0001C0h
0A1h	02h	CNTLR-2, ICW3 Indicates Slave ID
0A1h	01h	CNTLR-2, ICW4 8086 Mode
04D1h	00h	CNTRL-2, Edge/Level Control Register
0A1h	BDh	CNTLR-2, Interrupt Mask (may vary)

Figure 5-2 illustrates the sequence software must follow to load the interrupt controller Initialization Command Words (ICWs). The sequence must be executed for CNTRL-1 and CNTRL-2.



NOTE:
All ICWs must be programmed prior to programming the OCWs.

Figure 5-2. Initialization Sequence

5.11 Initialization Command Words

5.11.1 INITIALIZATION COMMAND WORDS 1 AND 2 (ICW1, ICW2)

In EISA systems the interrupt controllers are programmed for x86 mode. In an x86 system A15–A11 are inserted in the five most significant bits of the vectoring byte and the Interrupt Controller sets the three least significant bits according to the interrupt level. A10–A5 are ignored and ADI (Address interval) has no effect. (See Figures 5-3 and 5-4.)

ICW1 initializes the interrupt controller as follows:

LTIM: This bit is disabled in the EISA system. Its function is replaced by the Edge/Level Triggered Control register (ELCR). It allows each interrupt input to be programmed to either Edge or level mode on a channel-by-channel basis (refer to Section 5.14.7.1).

ADI: Ignored for EISA.

SNGL: This bit is set to 0 for EISA. It indicates that there is more than one interrupt controller in the system.

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

ICW2 initializes the interrupt controller with the five most-significant bits of the interrupt vector address (refer to Section 5.9).

CNTLR-1 (020h) program to 011h
CNTRL-2 (0A0H) program to 011h

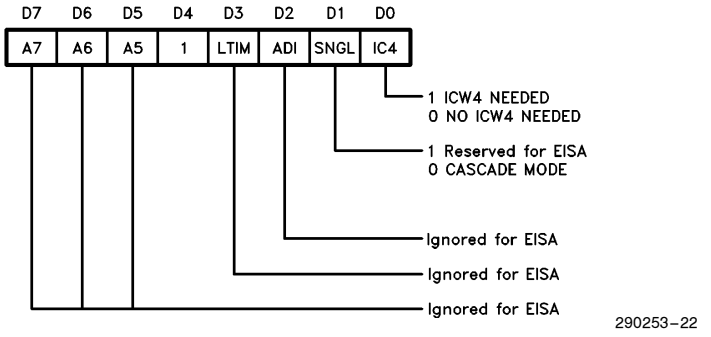


Figure 5-3. ICW1

CNTLR-1 (021h) program to 08h
CNTRL-2 (0A1h) program to 070h

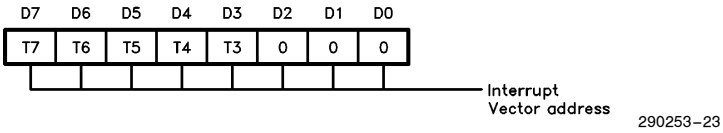


Figure 5-4. ICW2

5.11.2 INITIALIZATION COMMAND WORD 3 (ICW3)

This word is read only in EISA systems. An interrupt request on IRQ2 causes CNTLR-1 to enable CNTLR-2 to present the interrupt vector address during the second interrupt acknowledge cycle. (See Figures 5-5 and 5-6.)

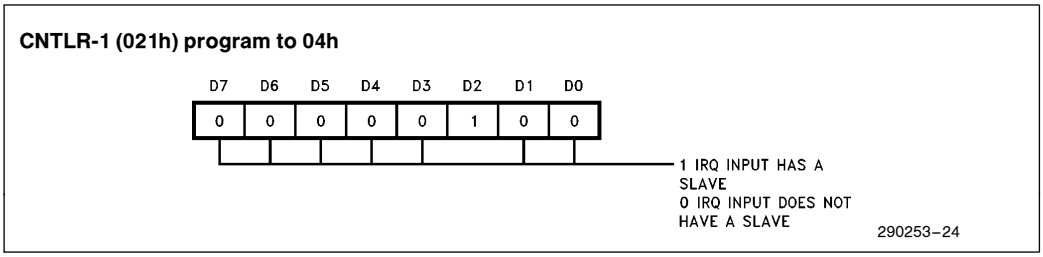


Figure 5-5. ICW3 (Master Device)

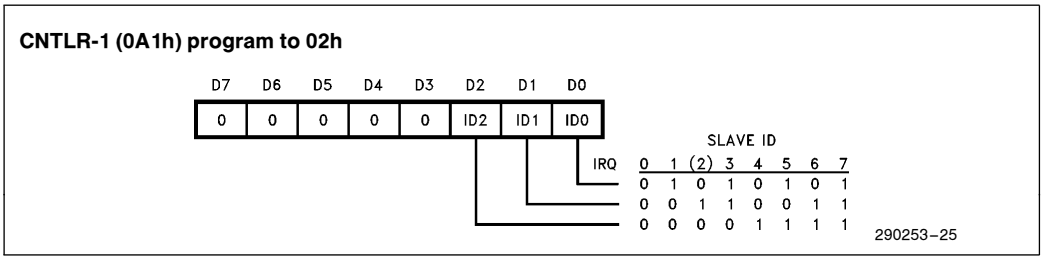


Figure 5-6. ICW3 (Slave Device)

5.11.3 INITIALIZATION COMMAND WORD 4 (ICW4)

(See Figure 5-7.)

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

BUF: Programmed to 0 for EISA.

M/S: Ignored for EISA.

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

μPM: Microprocessor mode: μPM = 0 sets the Interrupt Controller for MCS-80, 85 system operation (illegal for EISA systems), μPM = 1 sets the Interrupt Controller for x86 system operation.

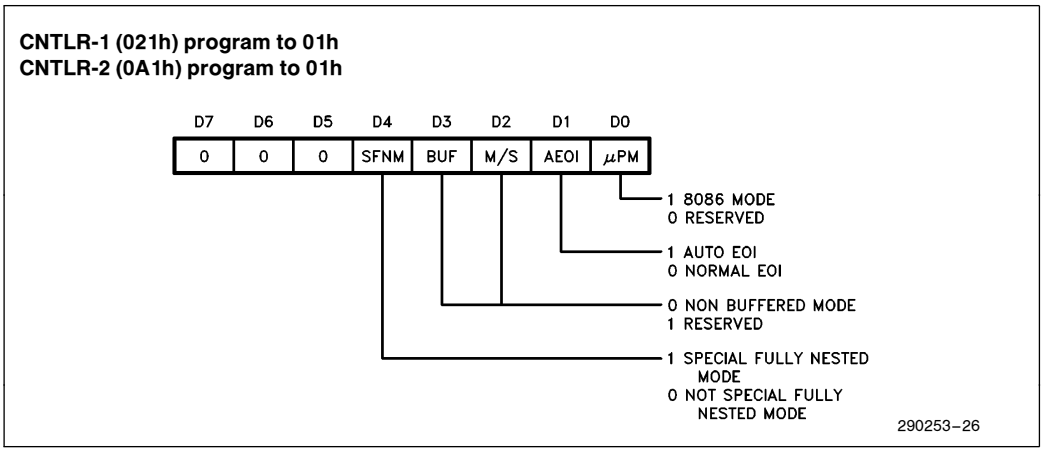


Figure 5-7. ICW4 021h(CNTLR-1) or 0A1h(CNTLR-2)

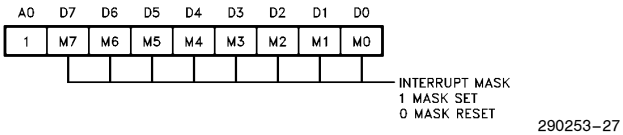
5.12 Operation Control Words (OCWs)

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

5.12.1 OPERATION CONTROL WORD 1 (OCW1)—READ/WRITE

OCW1 sets and clears the mask bits in the interrupt Mask register (IMR). M7–M0 represent the eight mask bits. M = 1 indicates the channel is masked (inhibited), M = 0 indicates the channel is enabled. (See Figure 5-8.)

IRQ<7:0>—Port 021h (CNTRL-1)
IRQ<15:0>—Port 0A1h (CNTRL-2)



290253–27

NOTE:
M0 = Mask for IRQ0(IRQ8#), M1 = Mask for IRQ1(IRQ9), M2 = Mask for IRQ2(IRQ10) and so on . . .

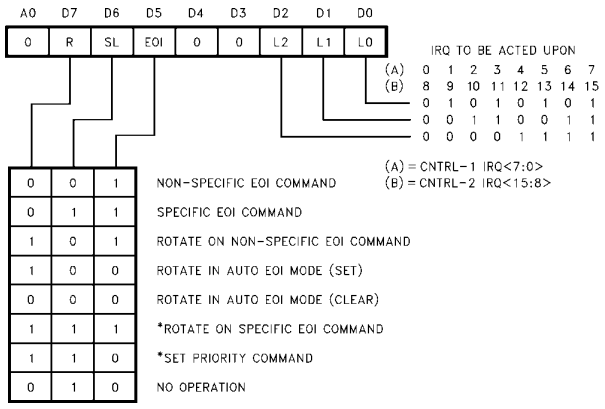
Figure 5-8. OCW1

5.12.2 OPERATION CONTROL WORD 2 (OCW2)

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

L2, L1, L0—These bits determine the interrupt level acted upon when the SL bit is active. (See Figure 5-9.)

IRQ<7:0>—Port 020h (CNTRL-1)
IRQ<15:0>—Port 0A0h (CNTRL-2)



0	0	1	NON-SPECIFIC EOI COMMAND
0	1	1	SPECIFIC EOI COMMAND
1	0	1	ROTATE ON NON-SPECIFIC EOI COMMAND
1	0	0	ROTATE IN AUTO EOI MODE (SET)
0	0	0	ROTATE IN AUTO EOI MODE (CLEAR)
1	1	1	*ROTATE ON SPECIFIC EOI COMMAND
1	1	0	*SET PRIORITY COMMAND
0	1	0	NO OPERATION

290253–28

*L0–L2 are used

Figure 5-9. OCW2

5.12.3 OPERATION CONTROL WORD 3 (OCW3)

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

SMM—Special Mask Mode. If ESMM = 1 and SSM = 1 the Interrupt Controller will enter Special Mask Mode. If ESMM = 1 and SMM = 0 the Interrupt Controller will revert to normal mask mode. When ESMM = 0, SMM has no effect. (See Figure 5-10.)

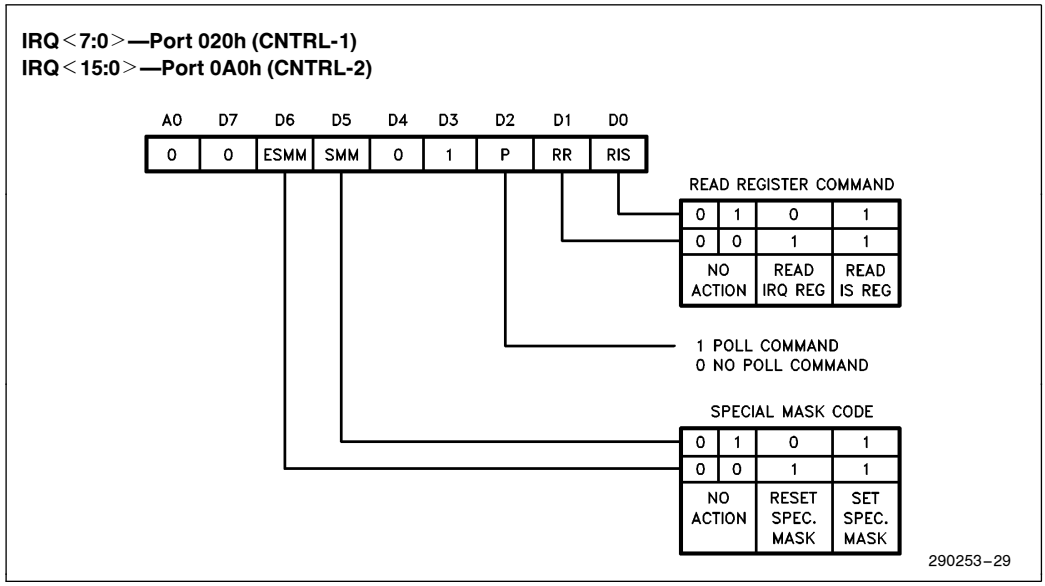


Figure 5-10. OCW3

5.13 End-Of-Interrupt Operation

5.13.1 END OF INTERRUPT (EOI)

The In Service (IS) bit can be reset either automatically following the trailing edge of the last in sequence INTA# pulse (when AEOI bit in ICW1 is set) or by a command word that must be issued to the Interrupt Controller before returning from a service routine (EOI command). An EOI command must be issued twice if the Cascade mode, once for the master and once for the slave.

There are two forms of EOI commands: Specific and Non-Specific. When the Interrupt Controller is operated in modes which preserve the fully nested structure, it can determine which IS bit to reset on EOI. When a Non-Specific EOI command is issued the Interrupt Controller will automatically reset the highest IS bit of those that are set, since in the fully nested mode the highest IS level was necessarily the last level acknowledged and serviced. A non-specific EOI can be issued with OCW2 (EOI = 1, SL = 0, R = 0).

When a mode is used which may disturb the fully nested structure, the Interrupt Controller may no longer be able to determine the last level acknowledged. In this case a Specific End of Interrupt must be issued which includes as part of the command the IS level to be reset. A specific EOI can be issued with OCW2 (EOI = 1, SL = 1, R = 0, and L0–L2 is the binary level of the IS bit to be reset).

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

5.13.2 AUTOMATIC END OF INTERRUPT (AEOI) MODE

If AEOI = 1 in ICW4, then the Interrupt Controller will operate in AEOI mode continuously until reprogrammed by ICW4. In this mode the Interrupt Controller will automatically perform a non-specific EOI operation at the trailing edge of the last interrupt acknowledge pulse. Note that from a system standpoint, this mode should be used only when a nested multilevel interrupt structure is not required within a single Interrupt Controller. The AEOI mode can only be used in a master Interrupt Controller and not a slave.

5.14 Modes of Operation

5.14.1 FULLY NESTED MODE

This mode is entered after initialization unless another mode is programmed. The interrupt requests are ordered in priority from 0 through 7 (0 highest). When an interrupt is acknowledged the highest priority request is determined and its vector placed on the bus. Additionally, a bit of the Interrupt Service register (IS0–7) is set. This bit remains set until the microprocessor issues an End of Interrupt (EOI) command immediately before returning from the service routine, or if AEOI (Automatic End of Interrupt) bit is set, until the trailing edge of the last INTA#. While the IS bit is set, all further interrupts of the same or lower priority are inhibited, while higher levels will generate an interrupt (which will be acknowledged only if the microprocessor internal Interrupt enable flip-flop has been re-enabled through software).

After the initialization sequence, IRQ0 has the highest priority and IRQ7 the lowest. Priorities can be changed, as will be explained, in the rotating priority mode.

5.14.2 THE SPECIAL FULLY NESTED MODE

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

a) When an interrupt request from a certain slave is in service this slave is not locked out from the master's priority logic and further interrupt requests from higher priority IRQ's within the slave will be recognized by the master and will initiate interrupts to the processor. (In the normal nested mode a slave is masked out when its request is in service and no higher requests from the same slave can be serviced.)

b) When exiting the Interrupt Service routine the software has to check whether the interrupt serviced was the only one from that slave. This is done by sending a non-specific End of Interrupt (EOI) command to the slave and then reading its In-Service register and checking for zero. If it is empty, a non-specific EOI can be sent to the master too. If not, no EOI should be sent.

5.14.3 AUTOMATIC ROTATION (EQUAL PRIORITY DEVICES)

In some applications there are a number of interrupting devices of equal priority. In this mode a device, after being serviced, receives the lowest priority, so a device requesting an interrupt will have to wait, in the worst case until each of seven other devices are serviced at most once. (See Figure 5-11.)

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

5.14.4 SPECIFIC ROTATION (SPECIFIC PRIORITY)

The programmer can change priorities by programming the bottom priority and thus fixing all other priorities; i.e., if IRQ5 is programmed as the bottom priority device, then IRQ6 will have the highest one.

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

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

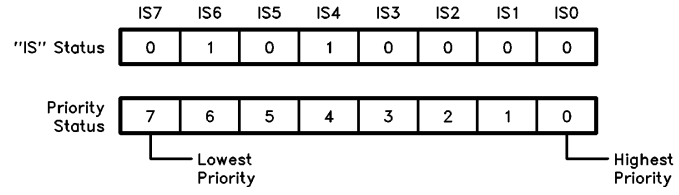
5.14.5 POLL COMMAND

In this mode the INT output is not used and the microprocessor internal Interrupt Enable flip-flop is reset, disabling its interrupt input. Service to devices is achieved by software using a Poll command.

The Poll command is issued by setting P = "1" in OCW3. The Interrupt Controller treats the next I/O read pulse to the Interrupt Controller as an interrupt acknowledge, sets the appropriate IS bit if there is a request, and reads the priority level. Interrupt is frozen from the I/O write to the I/O read.

If the priority and "in service" status is:

Before Rotate (IRQ4 the highest priority requiring service)



After Rotate (IRQ4 was serviced, all other priorities rotated correspondingly)

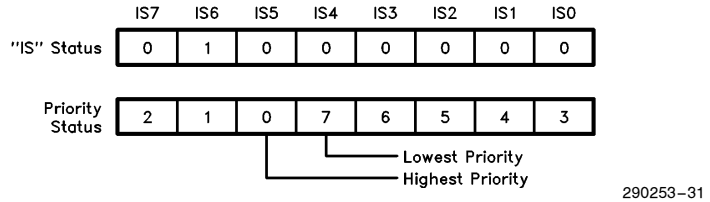
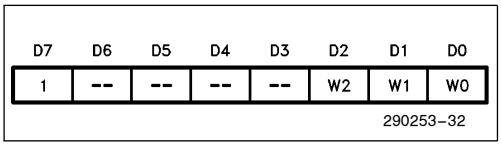


Figure 5-11

The word enabled onto the data bus during I/O read is:



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

1: Equal to ''1'' if there is an interrupt.

This mode is useful if there is a routine command common to several levels so that the INTA# sequence is not needed (saves ROM space).

5.14.6 CASCADE MODE

The Interrupt Controllers in the EISA system are interconnected in a system of one master with one slave to handle up to 15 priority levels.

In a cascade configuration, the slave interrupt outputs are connected to the master interrupt request inputs. When a slave request line is activated and afterwards acknowledged, the master will enable the corresponding slave to release the device routine address during byte 2 of INTA#.

Each Interrupt Controller in the system must follow a separate initialization sequence and can be programmed to work in a different mode. An EOI command must be issued twice: one for the master and once for the slave.

5.14.7 EDGE AND LEVEL TRIGGERED MODES

In ISA systems this mode is programmed using bit 3 in ICW1. In EISA systems the LTIM bit is disabled and a new register for level and edge triggered mode selection, per interrupt input, is included. This is the Edge/Level Control register ELCR. The default programming is equivalent to programming the LTIM bit (ICW1 bit 3) to a 0.

If an ELCR bit = ''0'', an interrupt request will be recognized by a low to high transition on the corresponding IRQ input. The IRQ input can remain high without generating another interrupt.

If an ELCR bit = ''1'', an interrupt request will be recognized by a ''low'' level on the corresponding IRQ input, and there is no need for an edge detection. For level triggered interrupt mode, the interrupt request signal must be removed before the EOI command is issued or the CPU interrupt must be disabled. This is necessary to prevent a second interrupt from occurring.

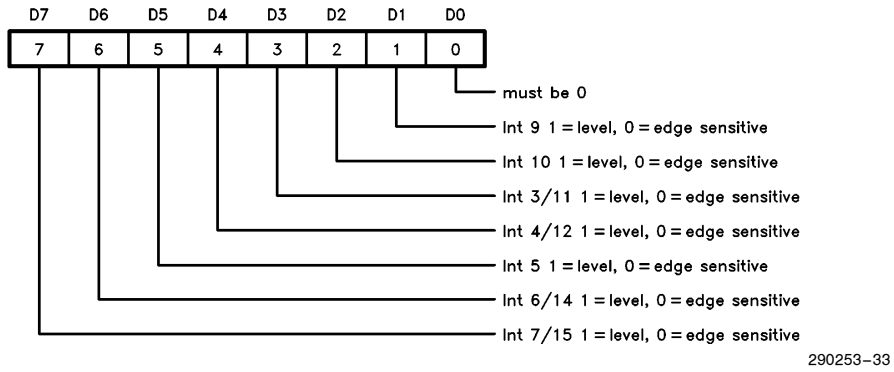
In both the edge and level triggered modes the IRQ inputs must remain active until after the falling edge of the first INTA#. If the IRQ input goes inactive before this time a DEFAULT IRQ7 will occur when the CPU acknowledges the interrupt. This can be a useful safeguard for detecting interrupts caused by spurious noise glitches on the IRQ inputs. To implement this feature the IRQ7 routine is used for ''clean up'' simply executing a return instruction, thus ignoring the interrupt. If IRQ7 is needed for other purposes a default IRQ7 can still be detected by reading the ISR. A normal IRQ7 interrupt will set the corresponding ISR bit, a default IRQ7 won't. If a default IRQ7 routine occurs during a normal IRQ7 routine, however, the ISR will remain set. In this case it is necessary to keep track of whether or not the IRQ7 routine was previously entered. If another IRQ7 occurs it is a default.

5.14.7.1 Edge/Level Triggered Control Register (ELCR) (Read/Write)

There are two ELCR registers one for each 82C59 bank. They are located at I/O ports 04D0h (for the Master Bank, IRQ<0:1,3:7>) and 04D1h (for the Slave Bank, IRQ<8#:15>). They allow the edge and level sense selection to be made on an interrupt by interrupt basis instead of on a complete bank. Only the interrupts that connect to the EISA bus may be programmed for level sensitivity. That is IRQ (0, 1, 2, 8#, 13) must be programmed for edge sensitive operation. (See Figure 5-12.)

IRQ13 still appears externally to be an edge sensitive interrupt even though it is shared internally with the Chaining interrupt. The Chaining interrupt is ''ORed'' after the edge sense logic.

IRQ<7:0>—Port 4D0h (CNTRL-1)
IRQ<15:0>—Port 4D1h (CNTRL-2)



These registers default to 0 upon RESET

Figure 5-12

5.15 Interrupt Masks

5.15.1 MASKING ON AN INDIVIDUAL INTERRUPT REQUEST BASIS

Each Interrupt Request input can be masked individually by the Interrupt Mask register (IMR) programmed through OCW1. Each bit in the IMR masks one interrupt channel if it is set 1. Bit 0 masks IRQ0, Bit 1 masks IRQ1 and so forth. Masking an IRQ channel does not affect the other channels' operation.

5.15.2 SPECIAL MASK MODE

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

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

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

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

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

5.16 Reading the Interrupt Controller Status

The input status of several internal registers can be read to update the user information on the system. The following registers can be read via OCW3 (IRR and ISR or OCW1 [IMR]).

Interrupt Request register (IRR): 8-bit register which contains the levels requesting an interrupt to be acknowledged. The highest request level is reset from the IRR when an interrupt is acknowledged. (Not affected by IMR).

In-Service register (ISR): 8-bit register which contains the priority levels that are being serviced. The ISR is updated when an End of Interrupt Command is issued.

Interrupt Mask register (IMR): 8-bit register which contains the interrupt request lines which are masked.

The IRR can be read when, prior to the RD pulse, a Read Register Command is issued with OCW3 (RR = 1, RIS = 0).

The ISR can be read when, prior to the RD pulse, a Read Register Command is issued with OCW3 (RR = 1, RIS = 1).

There is no need to write an OCW3 before every status read operation, as long as the status read corresponds with the previous one; i.e., the Interrupt Controller ''remembers'' whether the IRR or ISR has been previously selected by the OCW3. This is not true when poll is used.

After initialization the Interrupt Controller is set to IRR.

For reading the IMR, no OCW3 is needed. The output data bus will contain the IMR whenever I/O read is active, the address is 021h or 061h (OCW1).

Polling overrides status read when P = 1, RR = 1 in OCW3.

6.0 NON-MASKABLE INTERRUPT PORTS

6.1 NMI Status and Control (Port 061h) (Read/Write)

Value on Reset = 00x00000 (Figure 6-1)

Parity Error from System Memory

Bit 7 is set if the system board drives PARITY# active. This Interrupt is enabled by setting bit 2 to ''0''. To reset the parity error, set bit 2 to ''1'' and then clear it to ''0''.

IOCHK# Error from Expansion Board Memory

Bit 6 is set if an expansion board drives IOCHK# active on the ISA/EISA bus. This interrupt is enabled by setting bit 3 to ''0''. To reset the interrupt, set bit 3 to 1 and then set it to ''0''.

Port—061h

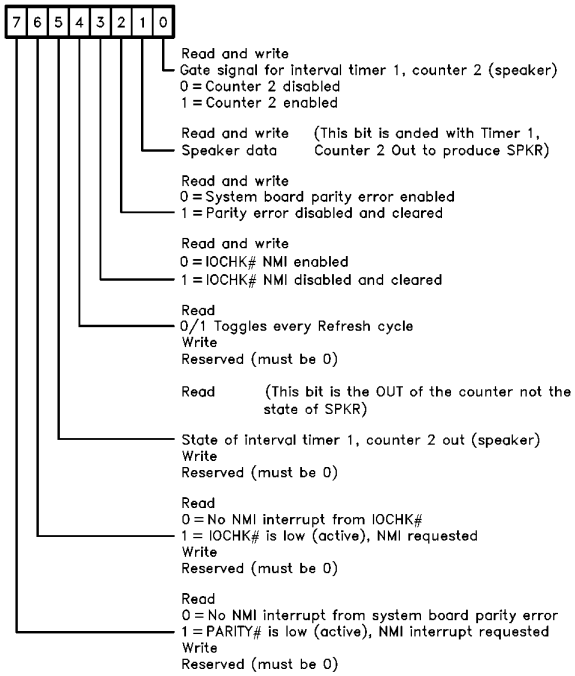


Figure 6-1

6.2 NMI Extended Status and Control (Port 0461h) (Read/Write)

Value on Reset = 00000000 (Figure 6-2)

This provides additional status and control.

Fail-Safe Timer Timeout

Bit 7 is set if the fail-safe timer count has expired before being reset by a software routine. This interrupt is enabled by setting bit 2 to ''1''. To reset the interrupt, set bit 2 to ''0'' then set it to ''1''.

Bus Timeout

Bit 6 is set if either a 64 BCLK or a 256 BCLK (Bus Timeout) occurs (refer to Section 4.1). The bus timeout interrupt is enabled by setting bit 3 to ''1'' or disabled by setting bit 3 to ''0''. To clear the bus timeout interrupt, set bit 3 to ''0'' and then set it to ''1''. The ISP drives RSTDRV active when a bus timeout occurs. Clearing the bus timeout status bit causes the ISP to negate RSTDRV.

Bit 4 indicates whether an 8 μs Bus Timeout occurred or not. For example, if Bit 6 = 1 and Bit 4 = 0, a 32 μs Bus Timeout occurred. If Bit 6 = 1 and Bit 4 = 1, an 8 μs Bus Timeout occurred.

Software Generated NMI

Bit 5 is set if an I/O write access occurred to port 0462h. This interrupt is enabled by setting 0461h bit 1 to ''1''. To reset the interrupt, set port 0461h bit 1 to ''0'' and then set it to ''1''.

Bus Reset

Bit 0 can be used to perform a system bus reset without resetting other devices in the system. A system bus reset is done by setting bit 0 to a ''1'', which drives the RSTDRV signal active on the ISA/EISA bus. Bit 0 should be set long enough for the system bus devices to be properly reset (8 BCLKs), and then bit 0 should be cleared to continue normal operation.

Bit 4 of Port 0461h has a new definition in the ISP B-stepping, which is shown in the Port 0461h Bit Map Table. The ISP stepping can be determined at system RESET:

Bit 4 = 1 (A-Step)

Bit 4 = 0 (B-Step)

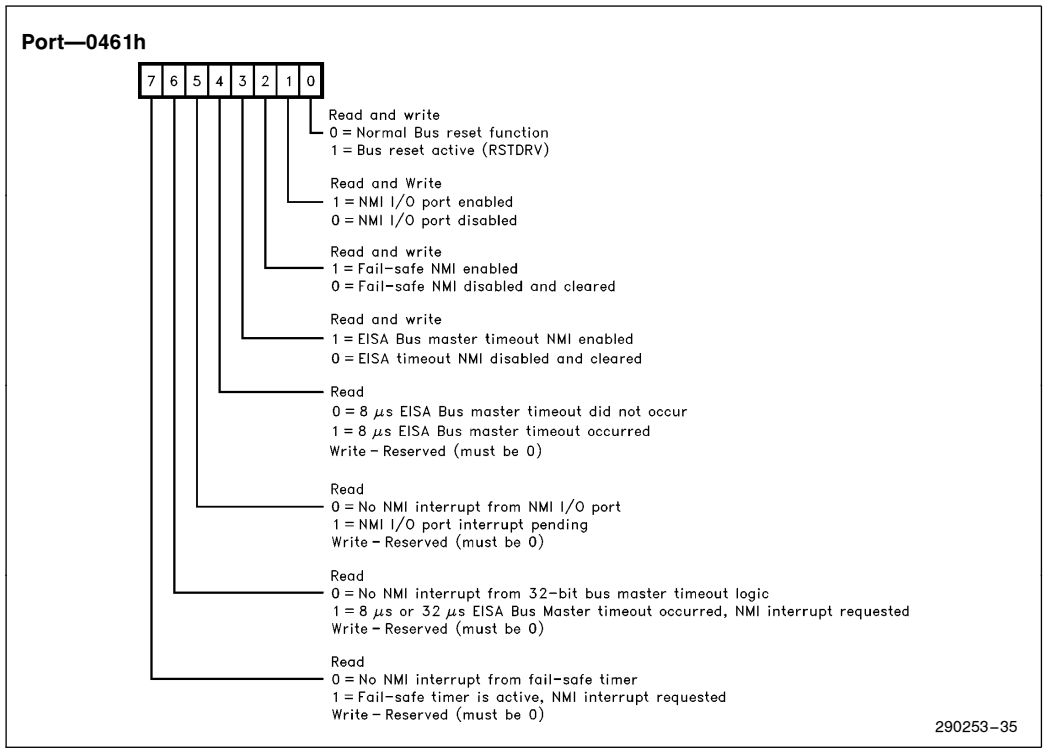


Figure 6-2

6.3 Software NMI Generation (Port 0462h) (Write Only)

A write to this port with any data will cause an NMI. This port provides a software mechanism to cause an NMI if interrupts are enabled. (Figure 6-3.)

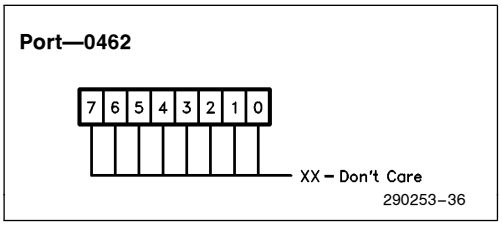


Figure 6-3

6.4 NMI Enable/Disable and Real-Time Clock Address (Port 070h)

The Mask register for the NMI interrupt is at I/O address 070h shown below. The most-significant bit enables or disables all NMI sources including IOCHK#, Fail Safe Timer, PARITY#, Bus Time Out, and the NMI Port. Write an 80h to port 70h to mask the NMI signal. This port is shared with the real-time clock. The real-time clock uses the lower six bits of this port to address memory locations. Writing to port 70h sets both the enable/disable bit and the memory address pointer. Do not modify the contents of this register without considering the effects on the state of the other bits. (Figure 6-4.)

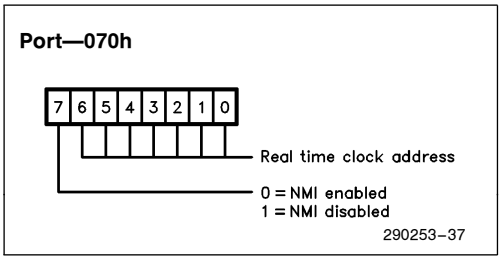


Figure 6-4

7.0 INTERVAL TIMER

EISA systems contain five counter/timers that are equivalent to those found in the 82C54 Programmable Interval Timer (NOTE: refer to the 82C54 Data Sheet for additional information on timer functions).

The following table shows the I/O address map of the interval timer counters:

I/O Port Address	Register Description
040h	Timer 1, System Timer (Counter 0)
041h	Timer 1, Refresh Request (Counter 1)
042h	Timer 1, Speaker Tone (Counter 2)
043h	Timer 1, Control Word Register
048h	Timer 2, Fail-Safe Timer (Counter 0)
049h	Timer 2, Reserved
04Ah	Timer 2, CPU Speed Control (Counter 2)
04Bh	Timer 2, Control Word Register

7.1 Programming the Interval Timer

The counter/timers are programmed by I/O accesses and are addressed as though they are contained in two separate 82C54 interval Timers. Timer 1 contains three counters, timer 2 contains two counters (EISA systems do not implement the middle counter of timer 2).

The interval timer is an I/O-mapped device. Several commands are available:

- Control Word Specifies:
  - — which counter to read or write
  - — the operating mode
  - — the count format (binary or BCD)
- Counter Latch latches the current count so that it can be read by the system. The countdown process continues.
- Read Back reads the count value, programmed mode, the current state of the OUT pins, and the state of the Null Count Flag of the selected counter.

The following Table lists the six operating modes for the interval counters.

Mode	Function
0	Out Signal on End of Count (=0)
1	Hardware retriggerable one-shot
2	Rate Generator (Divide by n Counter)
3	Square Wave Output
4	Software Triggered Strobe
5	Hardware Triggered Strobe

Because the timer counters wake up in an unknown state after power up, multiple refresh requests may be queued up. To avoid possible multiple refresh cycles after power up, program the timer counter immediately after power up.

Programming the interval timer is a simple process:

1. Write a control word.
2. Write an initial count for each counter.
3. Load the least and/or most significant bytes (as required by Control word bits 5, 4) of the 16-bit counter.

7.1.1 INTERVAL TIMER CONTROL WORD FORMAT

The Control Word specifies the counter, the operating mode, the order and size of the COUNT value, and whether it counts down in a 16-bit or binary-coded decimal (BCD) format. After writing the control word, a new count may be written at any time. The new value will take effect according to the programmed mode.

If a counter is programmed to read/write two-byte counts, the following precaution applies: A program must not transfer control between writing the first and second byte to another routine which also writes into that same counter. Otherwise, the counter will be loaded with an incorrect count. The count must always be completely loaded with both bytes. (See Figure 7-1.)

7.1.2 INTERVAL TIMER COUNTER LATCH COMMAND

The Counter Latch command latches the count at the time the command is received. This command is used to insure that the count read from the counter is accurate (particularly when reading a two-byte count). The count value is then read from each counter's Count register as was programmed by the Control register. (See Figure 7-2.)

Timer 1: Port 043h
Timer 2: Port 04Bh

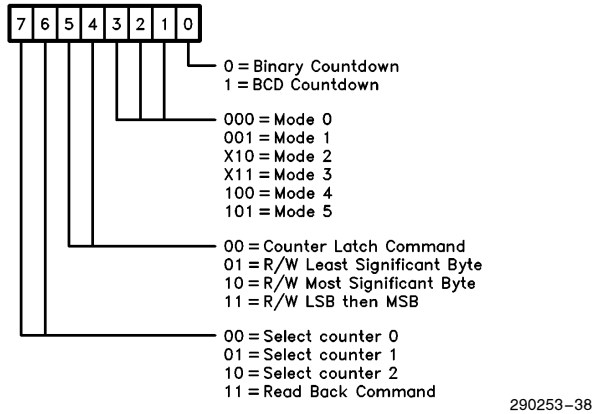


Figure 7-1. Control Word

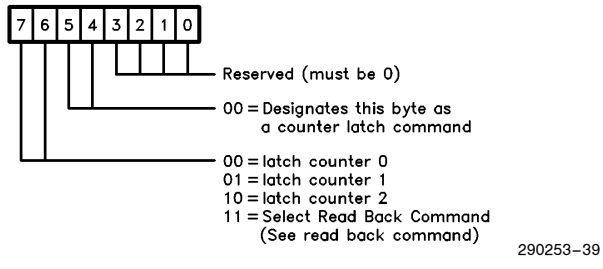


Figure 7-2

7.1.3 INTERVAL TIMER READ BACK COMMAND

The Read-Back command is used to determine the count value, programmed mode, and current states of the OUT pin and Null Count flag of the selected counter or counters. The Read-Back command is written to the Control Word register, which causes the current states of the above mentioned variables to be latched. The value of the counter and its status may then be read by I/O access to the counter address. Figures 7-3 and 7-4 show the format for the Read-Back command and the Status byte.

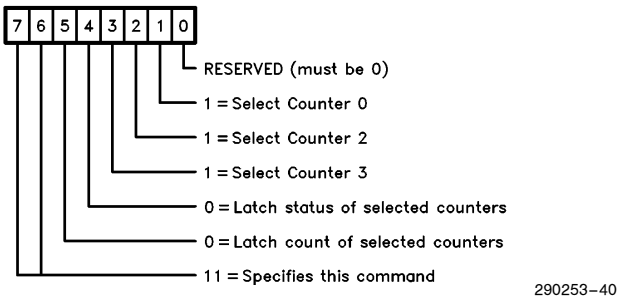


Figure 7-3. Current Read-Back Command Format

Timer 1: Port 040h, 041h, 042h
Timer 2: Port 048h, 04Ah

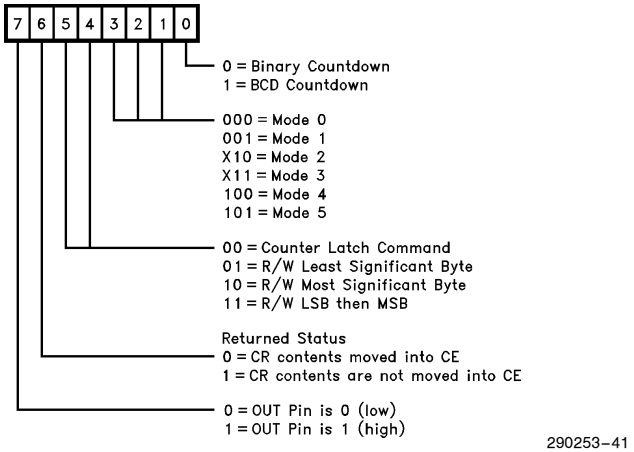


Figure 7-4. Status Byte Format

8.0 DETAILED SIGNAL DESCRIPTION

EISA Bus Cycle/BCLK timing definition points. (Figure 8-1.)

NOTE:
The following definition points are to be used as a reference when reading the detailed signal description.

T0 = rising edge of BCLK at the beginning of START# active.

T1 = falling edge of BCLK at the middle of START# active.

T2 = rising edge of BCLK at the beginning of CMD# active.

T3 = the first falling edge of BCLK after CMD# goes active.

T4A = the first rising edge of BCLK after CMD# goes active.

T4 = rising edge of BCLK at the end of a cycle (CMD# inactive edge).

T5 = the first falling edge of BCLK after CMD# goes inactive.

8.1 Signals Used during Arbitration

DREQ <3:0,7:5> INPUT (DMA REQUEST)

These input lines are used to request DMA service from the DMA subsystem or for a 16-bit ISA master to gain control of the system bus. The active level (high or low) is programmed in the Command registers. When the Command register bit 6 is programmed to 0, they are active high, otherwise they are active low. All inactive to active edges of DREQ are assumed to be asynchronous, they will be sampled on the falling edge of BCLK and used on the next rising edge of BCLK. The request must remain active until the appropriate DACK# goes active. Active to inactive edge sampling in Demand mode or in the case of a cascaded master will have various sampling points as defined below:

ISA Master—The DREQ line is sampled on the rising edge of BCLK, two BCLKs before the DACK# is inactivated. The DREQ is assumed to be asynchronous. The various address and control lines must be floated before DACK# goes inactive and the MASTER16# line must be released as DACK# goes inactive.

Compatible and Type A DMA—The DREQ line is sampled on the rising edge of BCLK, two BCLKs prior to the sampling of DRDY. This is 2.5 BCLKs before the address would be changed. The DREQ is assumed to be asynchronous. (Figure 8-2.)

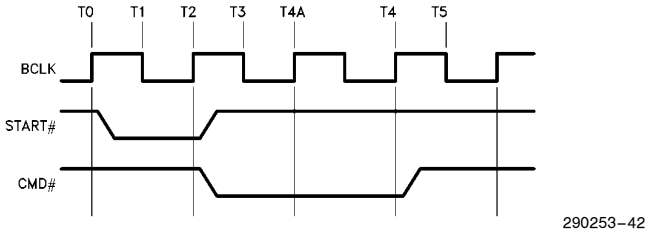
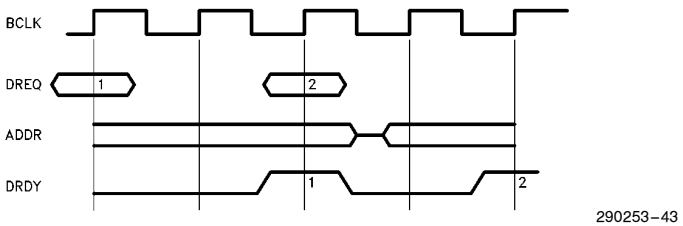


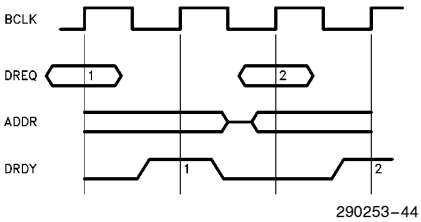
Figure 8-1



DREQ is sampled 2.5 BCLKs prior to changing the address

Figure 8-2

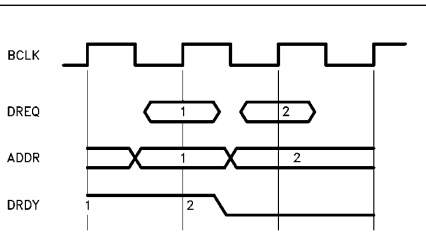
Type B DMA—The DREQ line is sampled on the rising edge of BCLK, 1 BCLK prior to the sampling of DRDY. This is 1.5 BCLKs before the address would be changed. The DREQ is assumed to be asynchronous. (Figure 8-3.)



DREQ is sampled 1.5 BCLKs prior to changing the address

Figure 8-3

Burst DMA—The DREQ line is sampled on the rising edge of BCLK, ½ BCLK after the address is changed on the bus. The DREQ must be synchronous to BCLK. (Figure 8-4.)



DREQ is sampled 0.5 BCLKs after changing the address

Figure 8-4

DACK# <3:0,7:5> OUTPUT
(DMA ACKNOWLEDGE)

These output lines indicate that a request for DMA service from the DMA subsystem has been recognized or that a 16-bit master has been granted the bus. The level of the DACK# lines when active may be programmed to be either high or low. This is accomplished by programming the DMA Command register. These lines should be used to decode the DMA slave device with the IORC# or IOWC# line to indicate selection. If used to signal acceptance of a bus master request, this signal indicates when it is legal to pull MASTER16# low. If the DMA controller has been programmed for a timing mode other than compatible mode, and another device has requested the bus, and a 4 μs time has elapsed, this line will be set inactive and the transfer stopped before the transfer is complete. In this case, the transfer will be restarted at the next arbitration period in which the channel wins the bus. Upon RST these lines are set inactive.

MREQ# <5:0> INPUT (MASTER REQUEST)

These inputs are slot specific signals used by EISA bus masters to request bus access. This signal, once set active, must remain active until the MACK# line indicates that the bus has been granted. The MREQ# line is to be negated on the falling edge of BCLK at or slightly before the end of a master transfer. The LA<>, BE<>#, M-IO#, and W/R lines should be floated on or before the rising edge of BCLK after MREQ# is negated. The end of the last bus cycle is determined by the arbiter via the activation of ST3 (CIP#) which is derived from CMD# in this case. The MREQ# signals are also activated on the falling edge of BCLK, they are always sampled on the rising edge of BCLK. These signals are synchronous with respect to BCLK and are active when low. After driving MREQ# active, the corresponding master must not drive MREQ# active again until 1.5 BCLKs after CMD# is driven inactive.

MACK# <5:0> OUTPUT
(MASTER ACKNOWLEDGE)

These outputs are slot specific signals used to acknowledge that an EISA bus master may use the bus that it has requested. This signal will go active from the rising edge of BCLK at which time the bus master may begin driving the LA<>, BE<>#, M-IO#, and W/R lines on the next falling edge of BCLK. MACK# will remain active until the rising edge of BCLK when MREQ# is sampled inactive. This line is sampled by EISA masters on the falling edge of BCLK. If another device has requested the bus, this line will be set inactive before MREQ# goes inactive. When the MACK# line goes inactive, the granted device has a maximum of 8 μs to release the MREQ# line and begin a final bus cycle. The MACK# signals will go active from the same rising edge of BCLK that EXMASTER# is activated. These signals are active when low. Upon reset they will be set inactive.

NOTE:
The ISP will deassert the MACK# signal a minimum of one BCLK after asserting it if another device (or refresh) is requesting the bus.

REFRESH# I/O OPEN COLLECTOR (REFRESH)

This signal is bidirectional. REFRESH# is used as an output during a refresh to indicate that a refresh cycle is in progress. It should be used to enable the SA<15:0> (or LA<15:2>) address to the row address inputs of all banks of dynamic memory so that when MRDC# (or CMD#) goes active, the entire system memory is refreshed at one time. Memory slaves must not drive any data onto the bus during refresh and should not add wait states since this would affect the entire system throughput. As an output, this signal is driven directly onto the EISA bus. This signal is an output only when the ISP is a master on the bus responding to an internally generated request for Refresh. Upon RST this pin will tri-state.

REFRESH# may be driven by an expansion bus adapter card acting as a 16-bit ISA bus master. As an input, it is assumed to be asynchronous with respect to BCLK. It will be sampled on the falling edge of BCLK and used on the following rising edge of BCLK.

DHLDA INPUT (HOLD ACKNOWLEDGE)

This signal indicates that the system has granted the ISP arbitration logic the use of the host bus. DHLDA is an asynchronous signal with respect to BCLK. It is sampled on the falling edge of BCLK and used on the next rising edge of BCLK. This signal is active high.

CPUMISS# INPUT (CPU CACHE MISS)

This signal from the host CPU or Cache controller subsystem, indicates that a CPU or Cache controller bus cycle is pending and that the System arbitration logic in the ISP should include the CPU as a contender in the next bus arbitration. This signal is asynchronous with respect to BCLK. It is sampled on the falling edge of BCLK and used on the next rising edge of BCLK. It is an active low signal. This signal should not be generated to the ISP when the CPU or cache own the bus (i.e., when HLDA is inactive).

DHOLD OUTPUT (CPU HOLD REQUEST)

This output signal is used to request control of the host or differentiation bus. In a simple system this would be making a request from the CPU, in a system with a cache, from the cache. This signal is synchronized to the requirements of the CPU or cache by the EBC. Upon RST this pin is set inactive.

EMSTR16# OUTPUT (ISA 16-BIT MASTER)

This output signal is used to tell the bus controller that an ISA 16-bit master has control of the bus. This will enable the bus controller to do the appropriate translation of signals. This signal is driven active 1 BCLK prior to the BCLK that the ISA Master DACK# signals are driven active, and driven inactive with the same BCLK that the ISA DACK# signals are changed on. This signal is active low. It is set inactive upon RST. EMSTR16# from the ISP must be ANDed with SPWROK from the EBC to generate EMSTR16# to the EBC so that EBC internal signals are initialized before START# is driven active in the first cycle.

EXMASTER# OUTPUT (EISA MASTER)

This output signal is used to tell the bus controller that a new EISA bus master, 16- or 32-bit, has control of the bus. This will enable the bus controller to do the appropriate translations of signals. This signal is driven active and inactive with the MACK# signals on the rising edge of BCLK except in the case of PREEMPT. In this case, EXMASTER# is negated on the same rising BCLK edge that CIP# (ST3) is sampled active. Upon RST this pin is set inactive.

8.2 Signals Used during DMA and Register Access

START# INPUT (START OF CYCLE)

This signal is connected directly to the EISA START#. It is used only in slave mode to indicate the start of a bus cycle. It is sampled on the rising edge of BCLK.

CMD# INPUT (COMMAND)

This signal is connected directly to the EISA CMD#. CMD# is used in slave mode to determine when to tri-state the data buffers after a read cycle. The data buffers will be floated when CMD# goes inactive. The leading edge of this signal is always synchronous, driven from a rising edge of BCLK. The trailing edge is asynchronous only in the case of an ISA bus master.

EOP TRI-STATED (END OF PROCESS)

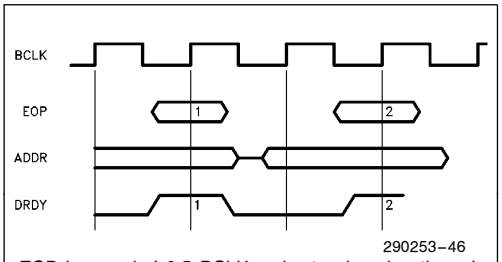
This pin is bidirectional, acting in one of three modes, and is directly connected to the TC line of the ISA/EISA bus. In the first mode, EOPIN, the pin is an input and can be used by a DMA slave to stop a DMA transfer. In the second mode, TCOUT, it is

used as a terminal count output by DMA slaves. An active pulse is generated when the byte counter reaches its last value. In the third mode, INTOUT, it indicates to an EISA programming master that a chaining buffer has been expired and a new chain buffer should be programmed. The timings are as follows:

Eopin Mode

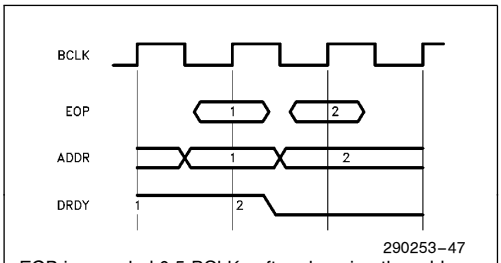
During DMA, using Compatible, Type A or Type B transfers, the EOP pin is sampled by the ISP on the rising edge of BCLK preceding the falling edge that address is changed. If it is sampled active, the address is tri-stated on the falling edge of BCLK, and the transfer is terminated. (Figure 8-5.)

When using Burst cycles, the EOP is sampled at the same time as the DREQ input, ½ BCLK after the address was changed. (Figure 8-6.)



EOP is sampled 0.5 BCLKs prior to changing the address

Figure 8-5



EOP is sampled 0.5 BCLKs after changing the address

Figure 8-6. Burst Mode EOP Timing

Tcout Mode

In this mode the EOP output behaves differently depending on whether or not the channel is programmed for Burst mode. If the channel is not programmed for Burst mode, the EOP output will go active 1.5 BCLKs after a new address has been output if the byte count expires with that transfer. In burst mode, EOP is driven active along with address since some transfers may only last 1 BCLK cycle. The EOP (TC) will be active until AEN# goes inactive for both Burst and non-Burst DMA (even after the address is tri-stated), except in the following two cases: Case 1—EOP (TC) will go inactive four BCLK cycles before AEN# goes inactive during an Autoinitialization or Buffer Chaining function. Case 2—EOP (TC) will go inactive one BCLK cycle before AEN# goes inactive if the DMA channel that is currently in use is configured for Type ''B'' timing.

Intout Mode

In this mode the EOP signal has the same behavior as the Chaining Interrupt to the host processor (IRQ13). If a chaining buffer is expired, EOP will go active on the falling edge of BCLK. Only the currently active channel's chaining interrupt will be reflected on this pin. Other channels with active chaining interrupts pending will not affect the EOP pin.

Whenever all the DMA channels are not in use, the EOP pin is kept in output mode and inactive.

BE<3:0># I/O TRI-STATED (BYTE ENABLES)

BE<3:0># are the byte enables of EISA bus. BE<2:0># are bidirectional and BE3# is an output only. In master mode BE<3:0># are outputs and are translated by the EBC to HBE<3:0># or ISA signals, whichever are appropriate. In Master mode the ISP will directly drive the EISA bus. The BE lines are always tri-stated on the T3 BCLK. The BE lines are re-enabled by the DMA on the falling edge of BCLK if DRDY is sampled active on the previous rising edge of BCLK.

In slave mode BE<2:0># are used in conjunction with HA<15:2> to address the ISP's internal registers. BE<2:0># are latched on the T2 edge of BCLK.

HA<31:2> I/O TRI-STATED (HOST ADDRESS BUS)

These signals are Address signals and are connected to the host bus. HA<31:20> and <15:2> are bidirectional, HA<19:16> are output only. In master mode they are outputs and are used in conjunction with BE<3:0># during DMA and Refresh cycles. In slave mode HA<15:2> and <31:20> are inputs. HA<15:2> in conjunction with BE<2:0># are used to address the internal ISP registers. HA<15:2> are latched with transparent latches by the trailing (rising) edge of DADS# (ST0). HA<31:20> are used to decode GT1M#. Upon RST these pins are tri-stated and placed in input Mode.

HW/R# I/O TRI-STATED (HOST WRITE/READ)

This line is bi-directional and connects to the host bus. In slave mode HW/R# is an input and is used to indicate a read cycle when low and a write cycle when high. When in slave mode, HW/R# is latched with transparent latches by the trailing (rising) edge of ST0 (DADS#). In master mode this pin is an output and will have the same timings as HA<31:2>. As an output, it is used during DMA or Refresh cycles by the EBC to propagate the appropriate write/read signals to the EISA Bus. This pin is placed in output mode only during DMA or Refresh cycles. Upon RST this pin is tri-stated and placed in input mode.

GT16M# OUTPUT TRI-STATED (>16 MBYTES)

GT16M# is directly connected to the EBC and indicates that the DMA address on HA<> is greater than 00FFFFFF (16 Mbytes). This signal is an output only when the DMA is the bus master, it is tri-stated otherwise.

The EBC uses this signal during DMA cycles to determine whether or not to generate ISA memory command signals. If it is not driven by another source, it should be pulled up with an external resistor to keep the node, which is an input to the EBC, from floating. This signal has timings similar to the HA<31:2> lines. Upon RST it will be tri-stated.

GT1M# OUTPUT (>1 MBYTE)

GT1M# is directly connected to the EBC and indicates when the address on the HA<> address lines is greater than 000FFFFFh (1 Mbyte). It is decoded by the ISP from the HA<31:20> bus and is used by the bus controller to control the SMRDC# and SMWTC# signals. Its timing is a combinatorial delay from HA<31:20> in all cycles. Upon RST its state depends on the state of the address bus.

RST INPUT (SYSTEM RESET)

This signal is connected directly to the EBC and indicates that the ISP should initialize all of its registers and state machines as well as driving the RSTDRV output active. This signal is asynchronous with respect to BCLK and must have a minimum active pulse width for eight BCLKs. 3.5 BCLKs are required from the inactive edge of RST to the T2 edge of the first slave cycle to the ISP.

BCLK INPUT (BUS CLOCK)

This signal is used as the main clock reference for the ISP and the EISA bus. BCLK is tied directly to the BCLK output of the EBC. The EBC divides the Host CPU Clock (HCLKCPU) by an appropriate number to generate BCLK. The normal frequency range of BCLK is 8.00 MHz to 8.333 MHz with a normal duty cycle of 50%. The high or low time can be stretched by the EBC for synchronization purposes, (REFER TO THE EBC DATA SHEET).

ST<3:0> I/O TRI-STATED WITH WEAK PULLUP (DMA STATUS)

These are synchronous bidirectional control signals between the ISP and the EBC. These pins are output during DMA and refresh cycles. They indicate what type of timing has been programmed for the current cycle and the size of the I/O device involved in the DMA transfer. The EBC takes ST0–ST3 and generates the appropriate command signals for the ISP. The command signals generated from ST0–ST3 are used to control the timing of the cycle. During cycles where the ISP is in slave mode, these pins are inputs. All of these pins have weak pullup resistors to sustain a high level when neither the EBC or ISP are driving the signals. Their definition is as follows:

Master Mode

ST1	ST0	DMA Cycle Timing
0	0	Compatible Cycle Timing
0	1	Type A Timing
1	0	Type B Timing
1	1	Burst Timing

These signals always transition on the falling edge of BCLK.

ST3	ST2	DMA Slave Size or Idle Indication
0	0	8-Bit
0	1	16-Bit
1	0	32-Bit
1	1	Idle

These signals will always go active on the falling edge of BCLK. They will go inactive on the falling edge of BCLK after the last DMA transfer, or on the rising edge of BCLK if a page break is detected (burst mode) or the transfer is about to end (burst mode).

Slave Mode

ST0 has the DMA address strobe function. This signal is used to latch the address and status information on the host bus. Specifically the signals latched with this signal are HA<15:2>, ST1 (M/IO#) and HW/R#. These signals have a setup and hold time around the rising edge of ST0. The leading edge of ST0 also has a setup time to BCLK at T2.

ST1 is the memory or I/O indicator pin (M/IO#), 1=Memory Cycle, 0=I/O Cycle. This signal is latched by the rising edge of ST0.

ST2 is the Interrupt Acknowledge pin (INTA#). When ST2=0, the cycle is an Interrupt Acknowledge. When ST2=1, the cycle is not an Interrupt Acknowledge. This signal is sampled by the ISP in slave mode at the T2 BCLK edge. It has a setup and hold time with respect to that BCLK edge so it must not be pipelined along with the HA lines. Only non-ISA masters can run interrupt acknowledge cycles. Only the host CPU has the control signals to run interrupt acknowledge cycles. ST2 must be tied high with a 1.2K pullup resistor.

ST3 is the cycle in progress pin (CIP#). This is an indication to the ISP System Arbiter that there is a bus cycle in progress. It is used to detect when the bus is free after an EISA master gives it up by driving MREQ# inactive. This signal is sampled on rising edges of BCLK.

These pins are output only when the DMA is the bus master or during Refresh cycles. Upon RST these pins are tri-stated and placed in input Mode.

DRDY I/O TRI-STATED WITH WEAK PULLUP (READY SIGNAL)

This pin is an output in slave mode and an input in master mode. In slave mode it is used by the EBC as an end of cycle indicator and is used to combinatorially drive CHRDY. DRDY is actively driven only when the ISP detects a slave cycle to one of its registers. At all other times it is sustained in a high state internally by a very weak pullup resistor. It is actively driven from the T2 edge of BCLK of an ISP cycle that requires more than a two BCLK wide CMD# pulse width when an ISA device is the bus master. DRDY will not be driven low when the bus master is either the host CPU or an EISA master. If DRDY is driven low, it will be kept low for four BCLKs after the T2 edge of BCLK. It will then be driven high off the rising edge of BCLK. The DRDY output buffer is enabled at T2 and disabled as a result of an inactive CMD# after T4A. DRDY will be asynchronously disabled upon the inactive edge of CMD#.

In master mode DRDY is used to indicate to the DMA controller that the current cycle is completed (DMA-READY) and that the DMA controller should pipeline addresses for Burst DMA transfers (START-PIPELINING). In master mode, DRDY, is sampled 2.5 BCLK periods after ST<3:2> are driven to a non-idle state from an idle one. This indicates START-PIPELINING. If DRDY is sampled high at this point, the DMA controller begins address pipelining. Pipelining is stopped at a page break or at the end of a transfer. DRDY is then sampled on the next BCLK rising edge and on all subsequent BCLK rising edges until ST<3:2> go to idle, to determine DMA-READY. If the DMA is pipelining addresses, DRDY will be sampled until active one last time after ST<3:2> go to the idle state (to complete the last DMA cycle). Upon RST this pin is tri-stated and must be tied high with a 2.4K pullup resistor.

D<7:0> I/O TRI-STATED (DATA LINES)

This bidirectional bus is the ISP's slave mode data bus. In master mode the data lines are not used. These pins are output only when CSOUT# is active during I/O read or Interrupt Acknowledge cycles. That is only when the ISP is being accessed as a result of an I/O read or Interrupt Acknowledge. These pins are inputs during an I/O write cycle to the ISP's internal registers. Upon RST these pins are tri-stated and placed in input mode.

This bus needs to be externally buffered to provide adequate drive to the EISA bus. In ISP master mode, this bus is not used. Upon RST these pins are tri-stated and placed in input Mode.

CSOUT# OUTPUT (SLAVE MODE SELECTED)

This output signal from the ISP indicates when the ISP is being accessed in slave mode. CSOUT# will go active from the rising edge of BCLK at T2 after a slave cycle to or from the ISP has been detected. It will not be active prior to that, i.e., between T0 and T2. CSOUT# is deactivated as a result of an inactive edge of CMD#, this is done asynchronously. CSOUT# will also be active during all interrupt acknowledge cycles. CSOUT# should be used to control the output enable of an EISA local data bus transceiver. This pin is always an output. Upon Reset it will be inactive (high).

AEN# OUTPUT (ADDRESS ENABLE)

This output signal, when inactive (high), indicates that the CPU or another EISA or ISA bus master has control of the bus. When active (low), the DMA controller has control of the bus. AEN# will also go active during Refresh cycles. AEN# is used to disable I/O devices which must not respond during a DMA cycle. This signal is further processed by the system board to form the slot specific AEN# signals. Upon RST this pin is set inactive.

8.3 Signals Used during an Interrupt Sequence

IRQ <15:3,1> INPUT (INTERRUPT REQUEST)

The active polarity of these inputs depends on the programming of the two ELCR registers. Only the EISA bus interrupts may be programmed for active low operation in the ELCR registers, all other interrupts will always be edge-triggered. If an interrupt is programmed for active low operation and shared with open collector drivers, an external pullup will need to be included to restore the inactive high level. Upon RST, the IRQ lines will be placed in edge-triggered mode.

The IRQ inputs must remain active until after the first falling edge of INTA(ST2#). If the IRQ input goes inactive before this time, a DEFAULT IR7 will occur when the CPU acknowledges the interrupt. (Refer to Section 5.14.7.)

INT OUTPUT (CPU INTERRUPT)

This signal is driven by the ISP's interrupt controller subsystem to signal the CPU that an Interrupt request is pending and needs to be serviced. It is asynchronous with respect to BCLK and is always an output. For i486 CPU systems, INT must be externally latched from the ISP until an Interrupt Acknowledge cycle occurs. INT will remain active until after the first INTA(ST2#) pulse. Upon RST the state of this pin is undefined.

8.4 Counter/Timer, NMI and Miscellaneous Utility Signals

OSC INPUT (COUNTER CLOCK)

This is a 14.31818 MHz input clock for use by the timers. This clock is divided by 12 and 48 to form the CLK input on several of the counters. It is asynchronous with respect to BCLK.

IOCHK# INPUT (I/O CHECK BUS ERROR)

This input signal comes from the ISA bus. It is used for parity errors on memory cards plugged into the bus, and for other high priority interrupts. The system board needs a pullup resistor to keep this signal inactive when it is not being driven active by some device. The active **level** of this signal will generate an NMI if this function is enabled in Port 061h. For example, if IOCHK# is active and Port 061h, bit 3 is set to 1 (inactive) no NMI will be caused, however, when Port 061h bit 3 is set to 0 (active) an NMI will be caused, providing that the IOCHK# input is still active and Port 070h bit 7 is enabled (0). This signal is asynchronous with respect to BCLK.

RSTDRV OUTPUT (SYSTEM BUS RESET)

This output signal is used by the EISA bus to disable and reset all installed adapters. More specifically, ISA/EISA bus masters and DMA devices must release and stop requesting the bus until reprogrammed by the system. Any drivers that are connected to the system bus must be tri-stated by this signal. In addition to being activated by RST, RSTDRV can be activated by a write to port 0461h or a bus timeout. A bus timeout will occur if an EISA bus master exceeds its allowable 8 μs time on the bus after being requested to get off by the removal of its MREQ#. Once the RSTDRV is activated by the bus timeout, the timed out master will release the bus allowing the host CPU to gain control of the system. A Bus Timeout will also occur upon the timeout of the 32 μs CMD# active timer. This output drives the EISA bus directly and has a 24 mA drive capability. RSTDRV will remain active as long as the source of the Bus Reset is active.

NOTE:
The 8x42 Keyboard Controller requires clocks (CLKKB) before the end of RSTKBD# (keyboard reset pulse) for keyboard reset to occur. To ensure that RSTKBD# has been driven inactive after the EBC has begun generating CLKKB; RSTDRV and RSTCPU are logically NANDed to generate RSTKBD# for the 8x42.

PARITY# INPUT (PARITY ERROR)

This signal is the main memory parity error input from the system board. The system board reports any parity errors from its memory system on this line. The active **edge** of this signal will generate an NMI if this function is enabled in Port 061h. For example, if PARITY# is active and Port 061h, bit 2 is set to 1 (inactive) no NMI will be caused, and when Port 061h, bit 2 is set to 0 (active) an NMI **will not** be caused even if Port 070h bit 7 is enabled (0). Another active edge on the PARITY# line would need to be detected in order to generate an NMI. This signal is asynchronous with respect to BCLK.

NMI OUTPUT (NON-MASKABLE INTERRUPT)

This signal from the NMI logic is used to force a non-maskable interrupt to the CPU. The CPU registers an NMI when it detects a rising edge on NMI. NMI will remain active until a read from the CPU to one of the NMI registers is detected by the ISP. This signal is set to low upon RST.

SPKR OUTPUT (SPEAKER DRIVE)

This signal is the output of timer 1, counter 2 and is ''ANDed'' with Port 061h bit 1 to provide Speaker Data Enable. This signal is to be used to drive an external speaker driver device, which, in turn drives the EISA system speaker. SPKR has a 24 mA drive capability. Upon reset, its output state is undefined.

SLOWH# OUTPUT (SLOW DOWN HOST CPU)

This output signal is from the CPU slowdown timer counter OUT, timer 2/counter 2. This counter is used to slow down the main CPU by pulse width modulation of its execution via the CPU's HOLD pin. When the first read is done to an I/O register in the 48h–4Bh range, the SLOWH# pin will be released to follow the output of the SLOWH# timer counter. Counter 2 is triggered by the refresh-request signal generated from timer 1/counter 1. Upon RST, SLOWH# will be set inactive (high).

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

RTCALE OUTPUT
(REAL TIME CLOCK LATCH ENABLE)

This active high output signal is provided by the ISP to latch the appropriate memory address into the Real Time Clock. A write to port 070h with the appropriate Real Time Clock memory address that will be written to or read from, will cause RTCALE to go active. RTCALE will go active from the rising edge of BCLK at T2 and will remain active for 1.5 BCLKs. The address is latched on the falling edge of RTCALE. Upon RST this pin is set inactive. (Figure 8-7.)

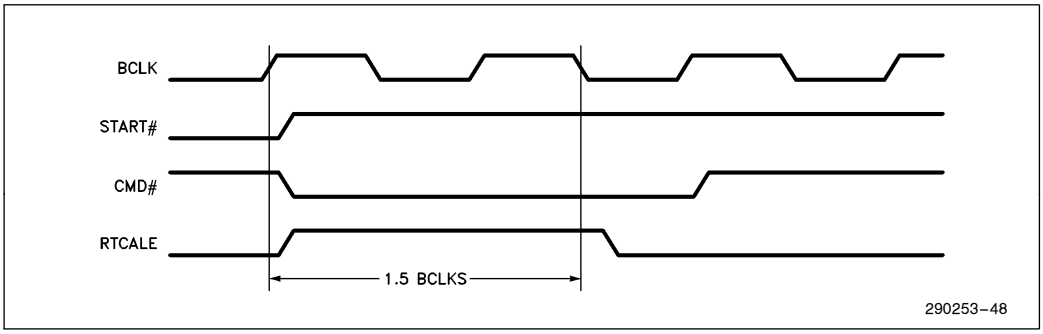


Figure 8-7

9.0 BASIC FUNCTION TIMING DIAGRAMS

The following general notes apply to all of the Basic Function timing diagrams:

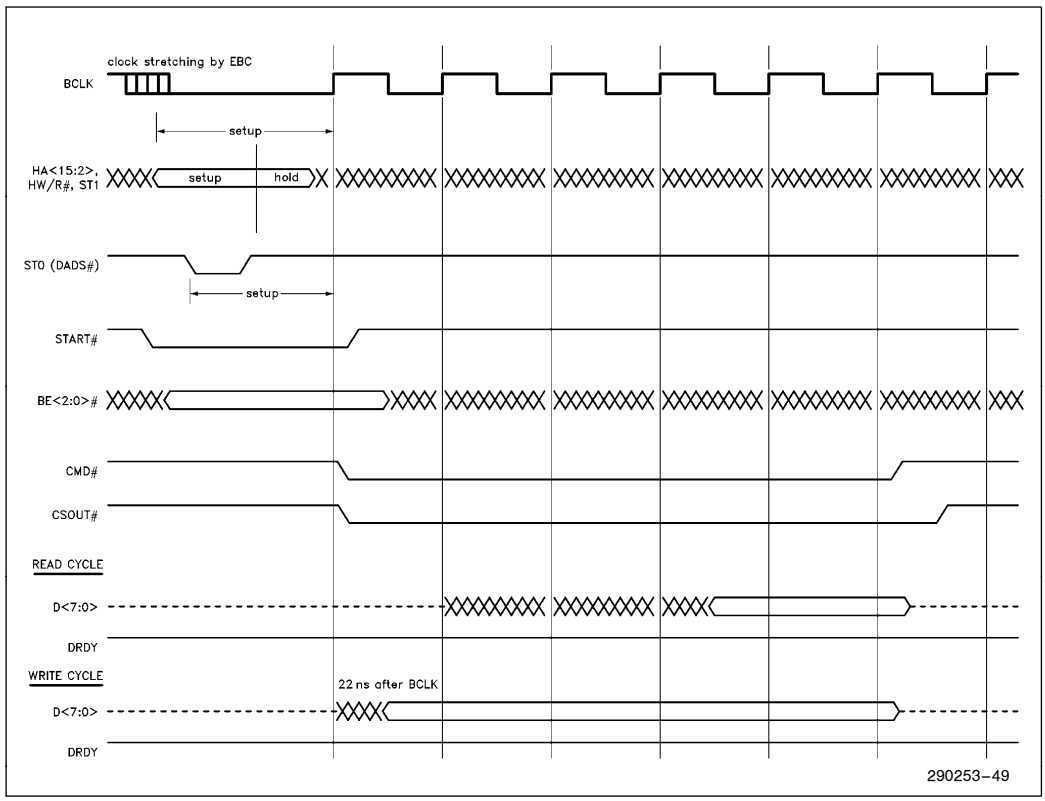
1) * = ISP Sampling point

2) x = Non-sampling points/don't care/Invalid Data

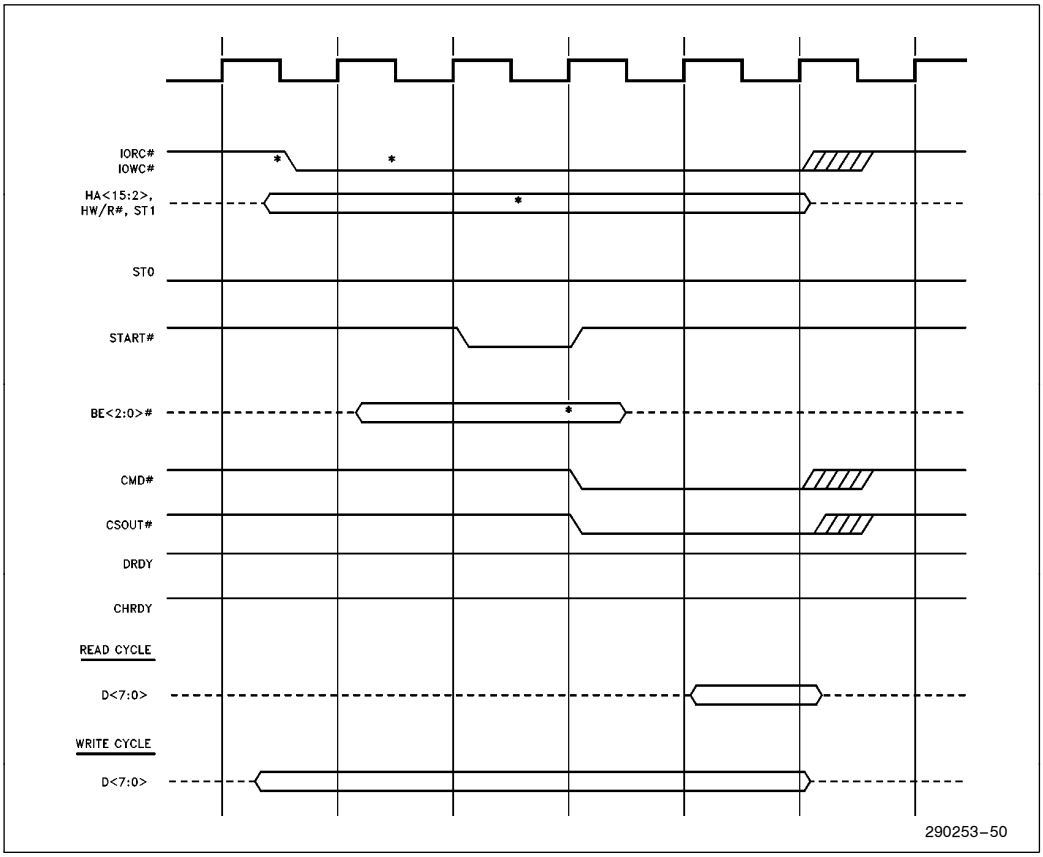
Refer to the 82358 EISA Bus Controller (EBC) data sheet for more timing diagrams.

9.1 Slave Mode

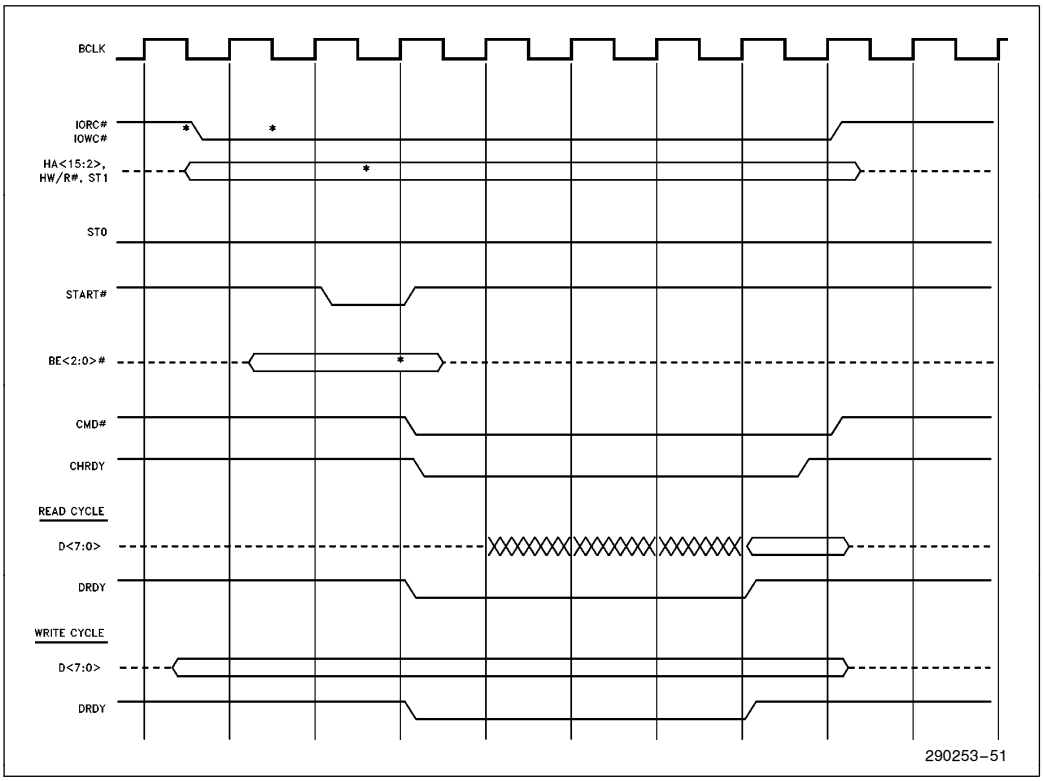
9.1.1 EISA MASTER OR HOST CPU CYCLE TO/FROM ISP



9.1.2 ISA MASTER SLAVE CYCLE TO/FROM THE ISP—SHORT CYCLE

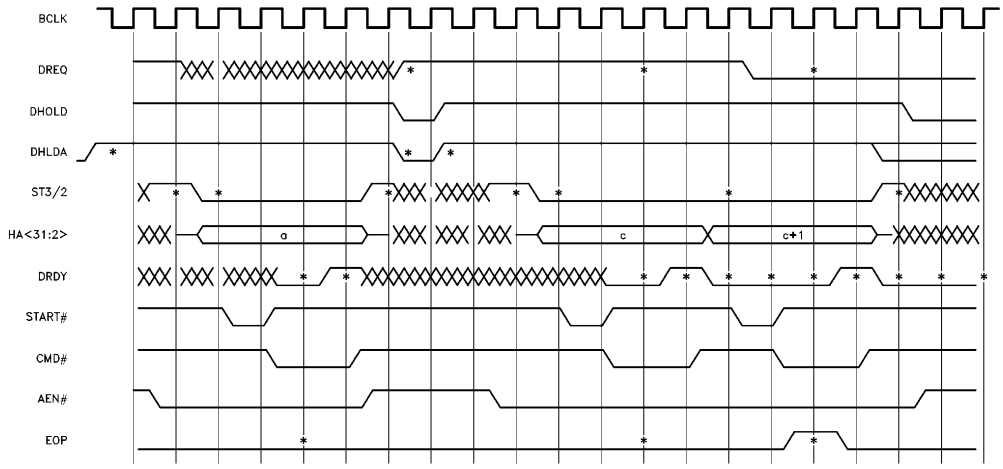


9.1.3 ISA MASTER SLAVE CYCLE TO/FROM THE ISP—LONG CYCLE



9.2 DMA Master Mode

9.2.1 NON BURST DMA CYCLES

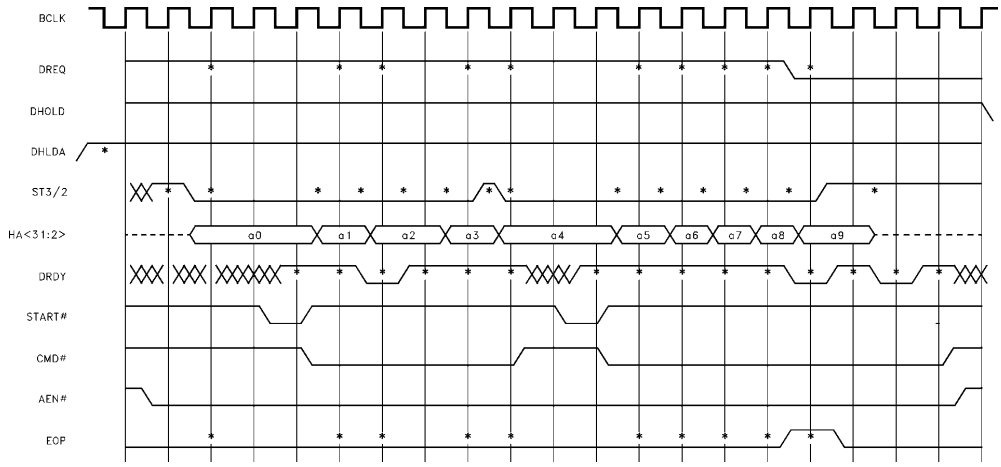


290253–52

Bus cycles illustrated:
a: Single Cycle DMA
c: Demand Mode DMA, cycle terminated with DRDY
c + 1: Demand Mode DMA, cycle terminated by either EOP or DREQ, Type B cycle

NOTE:
Type ''B'' Trailing Edge of AEN# is delayed one cycle.

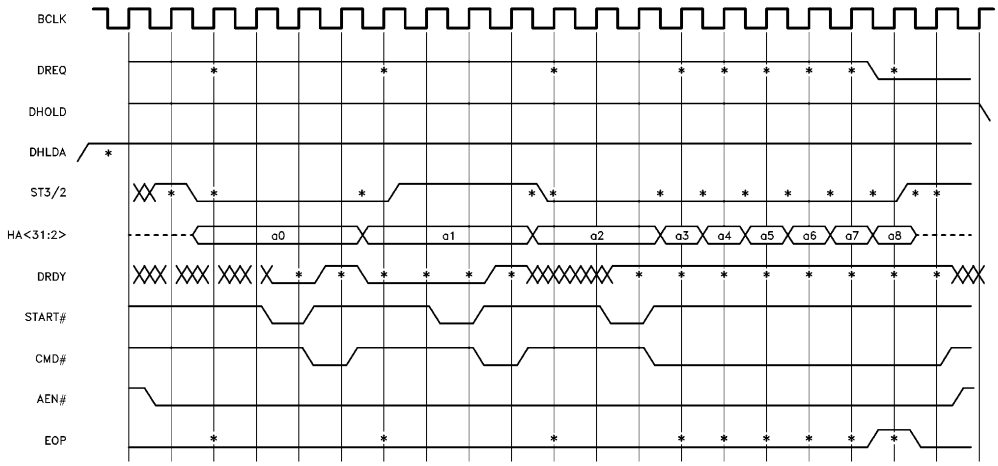
9.2.2 BURST DMA CYCLE DIAGRAM 1



290253–53

Bus Cycles illustrated:
A0: First DMA cycle of Burst, A1 address pipelined, 0 waitstate
A1: 2nd cycle of Burst, A2 address pipelined, 1 waitstate
A2: Continuation of Burst, A3 address pipelined, 0 waitstate
A3: End of Burst (A4 not in same page), A4 address pipelined, 0 waitstate
A4: Start of Burst, A5 pipelined, 0 waitstate
A5: A6 pipelined, 0 waitstate
A6: A7 pipelined, 0 waitstate
A7: A8 pipelined, 0 waitstate
A8: A9 pipelined, 1 waitstate
A9: Last DMA cycle, EOP sampled, DREQ sampled, 1 waitstate

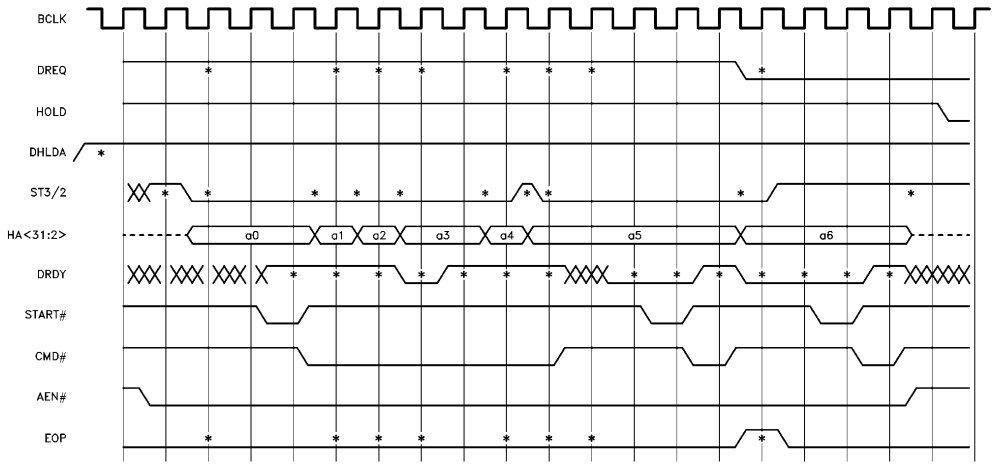
9.2.3 BURST DMA CYCLE DIAGRAM 2



290253–54

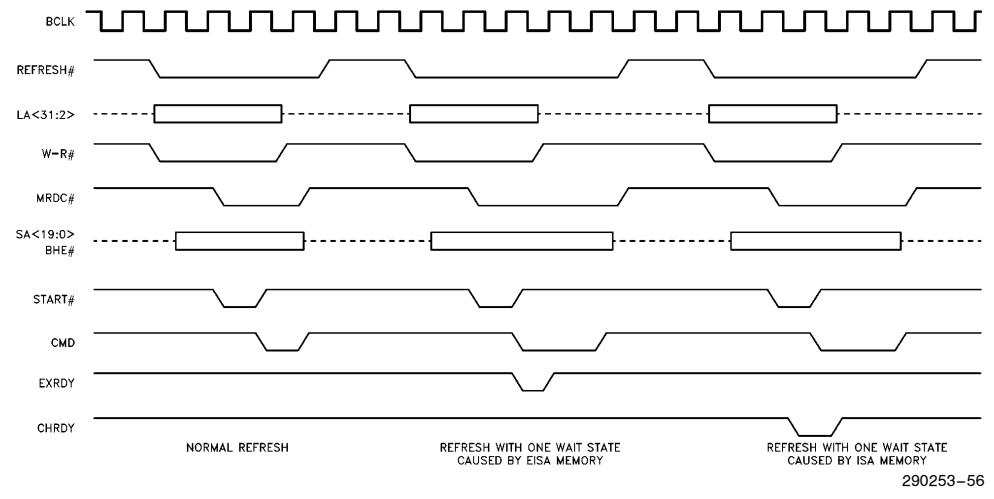
Bus Cycles illustrated:
A0: First DMA cycle, Burst Slave, non-Burst memory, 0 waitstate
A1: 1 waitstate, page break (A2 not in same page)
A2: Start of Burst, A3 address pipelined, 0 waitstate
A3: A4 pipelined, 0 waitstate
A4: A5 pipelined, 0 waitstate
A5: A6 pipelined, 0 waitstate
A6: A7 pipelined, 0 waitstate
A7: A8 pipelined, 0 waitstate
A8: Last DMA cycle, EOP sampled, DREQ sampled, 0 waitstate

9.2.4 BURST DMA CYCLE DIAGRAM 3



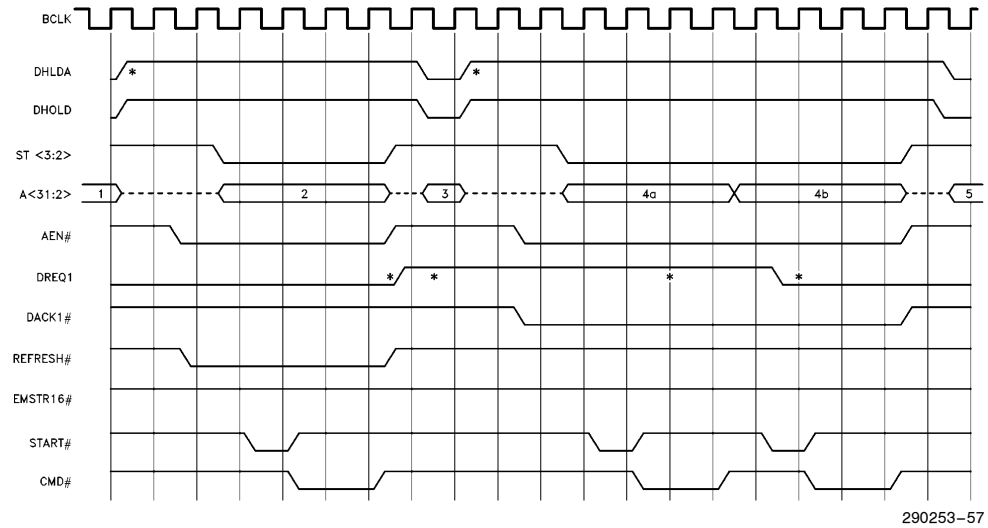
Bus Cycles illustrated:
A0: First DMA cycle, A1 pipelined, 0 waitstate
A1: A2 pipelined, 0 waitstate
A2: A3 pipelined, 0 waitstate
A3: A4 pipelined, 1 waitstate
A4: End of Burst (A5 not in same page), A5 pipelined, 0 waitstate
A5: Burst DMA slave, non-Burst Memory, 0 waitstate
A6: Last DMA cycle, EOP sampled, DREQ sampled, 1 waitstate

9.2.5 EISA REFRESH CYCLE



9.3 Arbiter Timing Diagrams

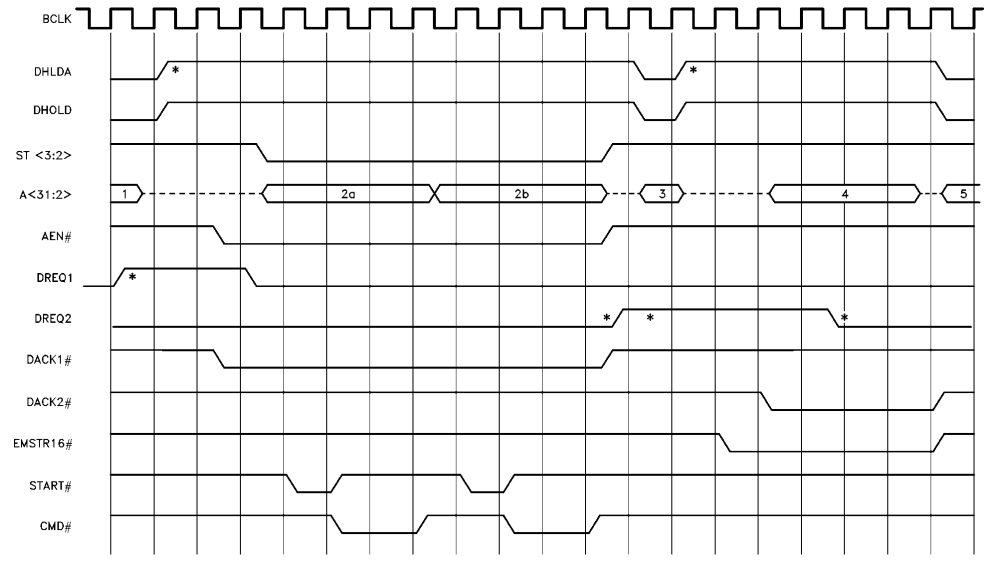
9.3.1 ARBITER TIMING DIAGRAM 1



290253–57

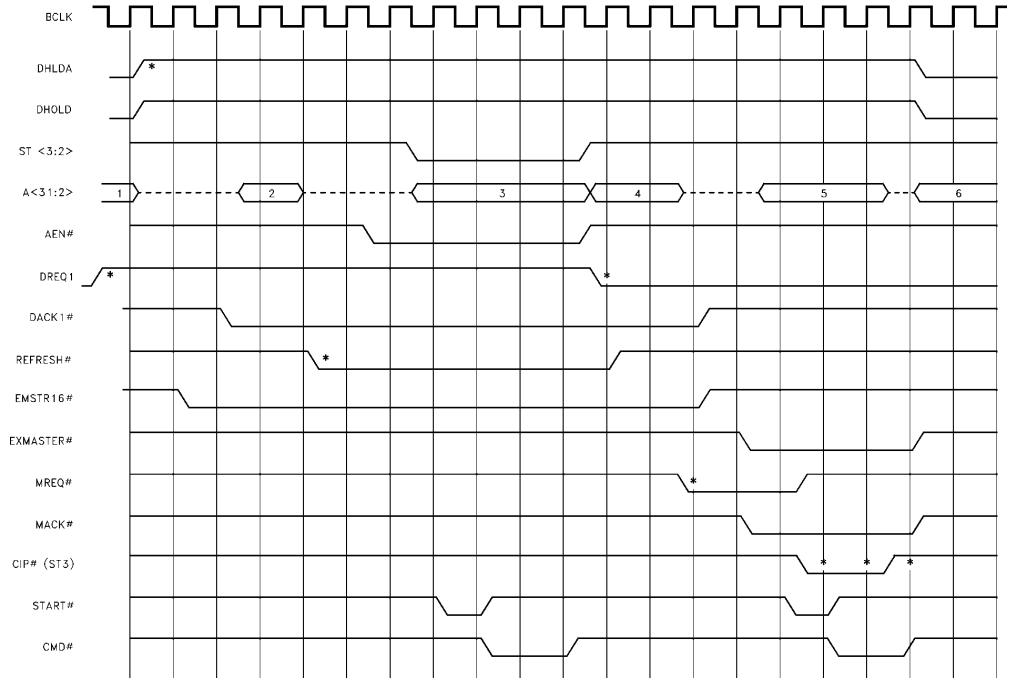
Cycles illustrated
1) CPU Cycle
2) Refresh Cycle
3) CPU Cycle
4) DMA Demand Mode Transfer, two Transfers, DREQ Inactivated
5) CPU Cycle

9.3.2 ARBITER TIMING DIAGRAM 2



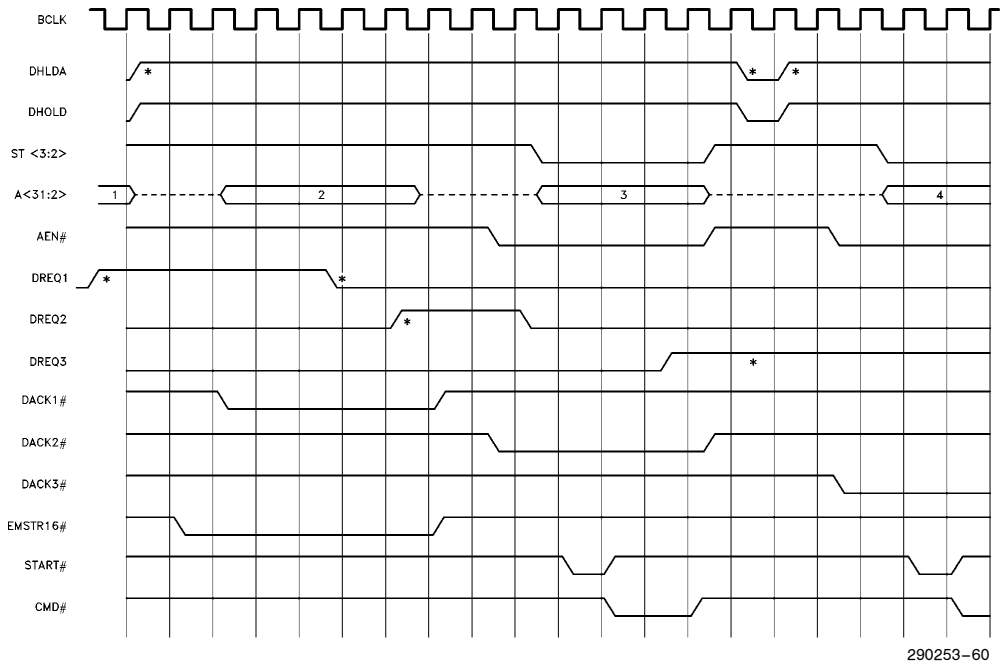
Cycles illustrated
1) CPU Cycle
2) DMA Block Mode, TC Terminated, two Transfers
3) CPU Cycle
4) ISA Master
5) CPU Cycle

9.3.3 ARBITER TIMING DIAGRAM 3



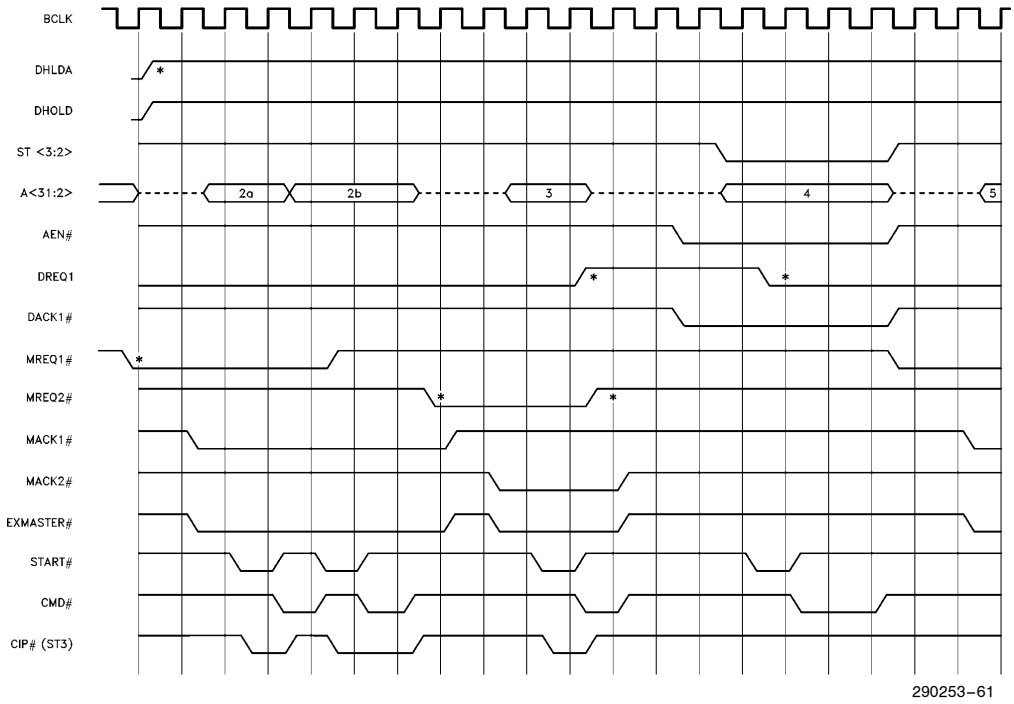
Cycles illustrated
1) CPU Cycle
2–4) ISA Master Cycle, Master Generates REFRESH#, ISA Master Cycle
5) EISA Master Cycle
6) CPU Cycle

9.3.4 ARBITER TIMING DIAGRAM 4



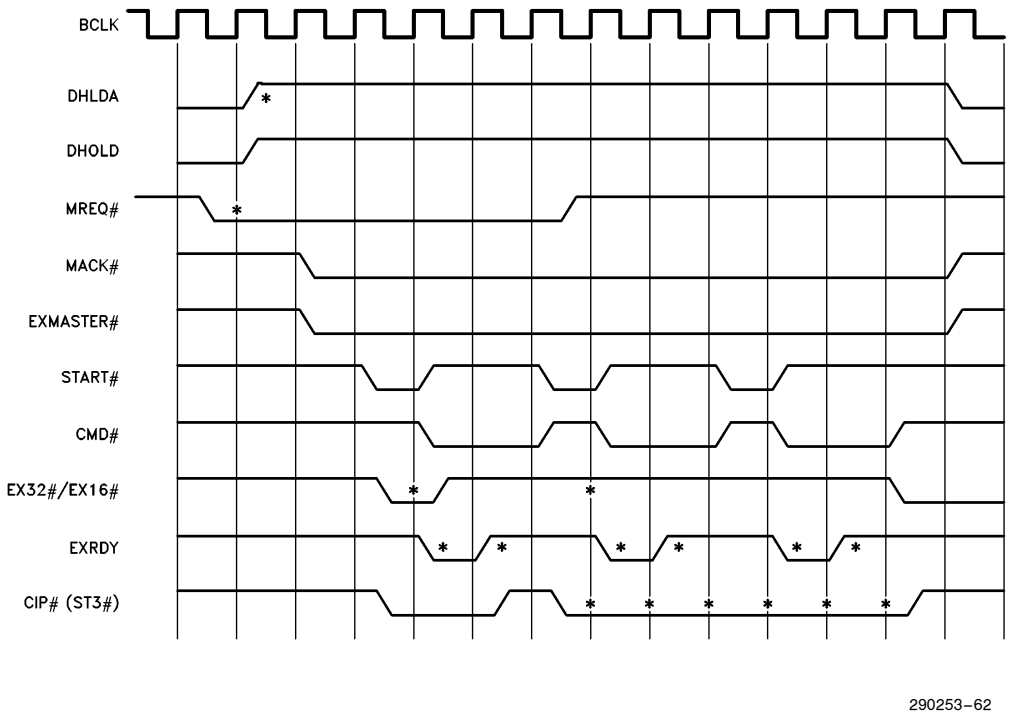
Cycles illustrated
1) CPU Cycle
2) ISA Master
3) Single Cycle DMA, Bus Returned to CPU
4) Demand Mode DMA

9.3.5 ARBITER TIMING DIAGRAM 5



Cycles illustrated
1) CPU Cycle
2) EISA Master1
3) EISA Master2
4) Demand Mode DMA, Type ‘‘A’’
5) EISA

9.3.6 ARBITER TIMING DIAGRAM 6—CIP# (ST3#) TIMING



Cycles illustrated
1) Matched EISA Bus Cycle
2) Mismatched EISA Bus Cycle, EISA Master Drops MREQ, Bus Controller Drops CIP# during Redrive

10.0 D.C. SPECIFICATIONS

NOTICE: This data sheet contains preliminary information on new products in production. The specifications are subject to change without notice. Verify with your local Intel Sales office that you have the latest data sheet before finalizing a design.

10.1 Maximum Ratings

Case Temperature Under Bias . . −65°C to + 110°C

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

Supply Voltages with
Respect to Ground . . . . . −0.5V to V_{CC} + 6.5V

Voltage On Any Pin . . . . . −0.5V to V_{CC} + 0.5V

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

10.2 D.C. Specification Tables

T_C = 0°C to +85°C, V_{CC} = 5V ±5%

T_A = 0°C to +70°C

Symbol	Parameter	Limits		Units	Test Conditions
		Min	Max		
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.5	V	
V_{ILC}	BCLK Input Low	−0.5	0.8	V	
V_{IHC}	BCLK Input High	V_{CC} − 0.8	V_{CC} + 0.5	V	
V_{OL1}	Output Low Voltage		0.45	V	I_{OL} = 5 mA
V_{OH1}	Output High Voltage	2.4		V	I_{OH} = −1 mA
V_{OL2}	Output Low Voltage		0.45	V	I_{OL} = 24 mA(1)
V_{OH2}	Output High Voltage	2.4		V	I_{OH} = −4 mA(1)
V_{OL3}	Output Low Voltage		0.45	V	I_{OL} = 4 mA(1)
V_{OH3}	Output High Voltage	2.4		V	I_{OH} = −250 μA(1)
I_{LI1}	Input Leakage		±15	μA	0V < V_{IN} < V_{CC}
I_{LI2}	Input Leakage		+15	μA	V_{IN} = V_{CC}(2)
I_{LI3}	Input Leakage	−45	−240	μA	V_{IN} = 2.4V(2)
I_{LO}	Output Leakage		±15	μA	0.45 < V_{IN} < V_{CC}
Cap	Cap. In, Out, I/O		12	pF	@ 1 MHz(3)
C_{CLK}	BCLK Cap.		20	pF	@ 1 MHz(3)
I_{CC}	V_{CC} Supply Current		200	mA	BCLK = 8.3 MHz V_{IL} = 0V V_{IH} = 4V

NOTES:
1. V_{OL2} and V_{OH2} apply only to the signals that directly drive the EISA bus and are not slot specific. These are: RSTDRV, REFRESH#, EOP, and BE<3:0>#.
V_{OL3} and V_{OH3} apply only to the signals that do not drive onto buses in the system. These are: INT, EMSTR16#, EXMASTER#, NMI, SLOWH#, GT16M#, GT1M#, DHOLD, AEN#, DRDY, ST<3:0>, and CSOUT#. All other outputs use V_{OL1} and V_{OH1}.
2. I_{L12} and I_{L13} apply only to those pins that include a weak sustaining pullup transistor. These are: ST<3:0>, DRDY and IRQ8#. All other IO and inputs should use I_{LI1}.
3. Sampled only.

11.0 A.C. SPECIFICATIONS

The A.C. specifications given in the following tables consist of output delays and input setup and hold requirements. The A.C. diagram's purpose is to illustrate the clock edges and specific signal edges from which the timing parameters are measured. The reader should not infer any other timing relationships from them. For specific information on timing relationships between the signals, refer to the appropriate functional and pin definition sections.

11.1 A.C. Characterization Tables

The following timing definition points are to be used as a reference when reading the A.C. Characteristic Tables.

EISA Bus Cycle/BCLK timing definition points.

T0 = Rising edge of BCLK at the beginning of START# active.

T1 = Falling edge of BCLK at the middle of START# active.

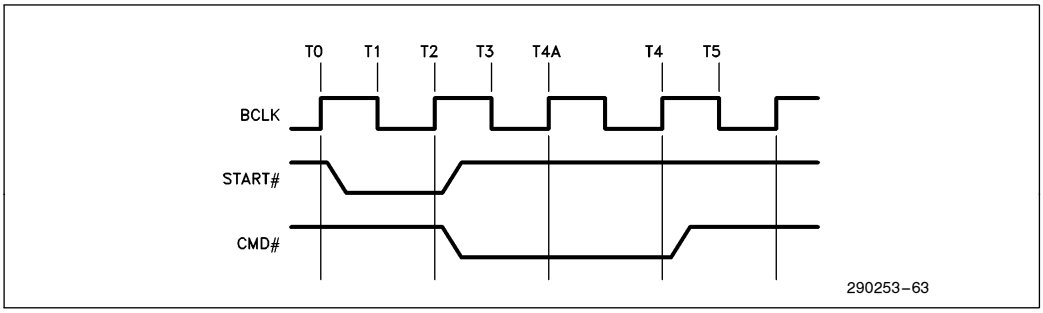
T2 = Rising edge of BCLK at the beginning of CMD# active.

T3 = The first falling edge of BCLK after CMD# goes active.

T4A = The first rising edge of BCLK after CMD# goes active.

T4 = Rising edge of BCLK at the end of a cycle (CMD# inactive edge).

T5 = The first falling edge of BCLK after CMD# goes inactive.



T_C = 0°C to +85°C, V_{CC} = 5V ±5%
T_A = 0°C to +70°C

Symbol	Parameter	Min	Max	Unit	Notes
CLOCK, OSC TIMING					
	BCLK Period				
T1	Period	120		ns	@ 1.5V
			500	ns	Not an ISP Slave Cycle
			170	ns	ISP Slave Cycle
T2a	High Time	55		ns	@ 2.0V
T2b	Low Time	55		ns	@ 0.8V
	OSC				
T77	Period	65	70	ns	For Asynchronous OSC
					@ 1.5V
T78a	High Time	20		ns	@ 2.0V
T78b	Low Time	20		ns	@ 0.8V
SLAVE MODE TIMING					
	HA<15:2>,ST<2:1>				
T3a	Setup Time	80		ns	To T2 BCLK
	ST0				
T3b	Setup Time	55		ns	To T2 BCLK
	HA<15:2>,ST1,HW/R#				
T4	Setup Time	30		ns	To ST0 Rising Edge
T5	Hold Time	10		ns	To ST0 Rising Edge
	BE<2:0>#,HW/R#				
T6a	(BE)Setup Time	60		ns	To T2 BCLK
T6b	(HW/R#)Setup Time	40		ns	To T2 BCLK
	BE<2:0>#,ST2				
T7	Hold Time	30		ns	
	START#				
T8	Setup Time	30		ns	
T9	Hold Time	0		ns	
	Data (Write)				
T10	Setup Time	15		ns	To T4a BCLK
T11	Hold Time	5		ns	To T4 BCLK
	Data (Read)				
T12	Delay Valid		60	ns	Prior To T4 BCLK
T13	Float		40	ns	From CMD#, Xcvr Must Meet EISA 35 ns(1)
	CSOUT#				
T14	Delay Active		35	ns	From BCLK Rising Edge
T15	Delay Inactive		40	ns	From CMD# Rising Edge
	DRDY				
T16	Delay Active/Inactive		45	ns	From BCLK Rising
T17	Float		45	ns	From CMD# Rising Edge.(1)
	CMD#				
T18	Setup Time	90		ns	Sync (Active/Inactive)
T19	Hold Time	0		ns	Sync (Active/Inactive)

Symbol	Parameter	Min	Max	Unit	Notes
DMA MASTER TIMING					
	HA<31:2>,HW/R#,GT16M#				
T22a	Delay Valid	1	35	ns	Initial Address, HW/R#, GT16#
T22b	Delay Valid	1	30	ns	Subsequent Addresses
T23	Float	1	30	ns	(1)
	ST<3:0>				
T24	Delay Active/Inactive	4	37	ns	
T25	Float	1	40	ns	(1)
	ST<3:2>				
T26	Delay Inactive	4	37	ns	Burst Cycle/Page Break
	DRDY				
T27	Setup Time	20		ns	
T28	Hold Time	15		ns	
	AEN#				
T29	Delay Active/Inactive		40	ns	
	BE<3:0>#				
T30	Delay Valid	2	45	ns	Direct to EISA to Meet Burst Timings
T31	Float		45	ns	Float from T3 BCLK.(1)
	EOP				
T32	Delay Active		40	ns	Non-Burst
T33	Delay Active/Inactive		40	ns	
T34	Float from DACK		30	ns	From DACK Active.(1)
T35	Setup Time	15		ns	Sync (Burst Mode)
T36	Hold Time	15		ns	Sync (Burst Mode)
T37	Setup Time	15		ns	Async (Non-Burst)
T38	Hold Time	15		ns	Async (Non-Burst)
ARBITER TIMING					
	DREQ				
T39	Setup Time	15		ns	Sync (Trailing Edge)
T40	Hold Time	15		ns	Sync (Trailing Edge)
T41	Setup Time	15		ns	Async (Leading/Trailing Edge)
T42	Hold Time	15		ns	Async (Leading/Trailing Edge)
	MREQ#				
T43a	Setup Time	17		ns	BCLK Rising
T43b	Setup Time	80		ns	BCLK Falling, 0.5 BCLKs after T43a
T44	Hold Time	15		ns	
	CPUMISS#				Asynchronous
T45	Setup Time	15		ns	
T46	Hold Time	15		ns	
	DHLDA				Asynchronous
T47	Setup Time	15		ns	
T48	Hold Time	15		ns	
	DACK#				
T49a	Delay Active/Inactive		50	ns	240 pF (ISA Masters)
T49b	Delay Active/Inactive		50	ns	240 pF (DMA Devices)
	MACK#				
T50	Delay Active/Inactive		40	ns	
	DHOLD				
T51	Delay Active/Inactive		40	ns	

Symbol	Parameter	Min	Max	Unit	Notes
ARBITER TIMING (Continued)					
T52	EMSTR16# Delay Active/Inactive		40	ns	
T53	EXMASTER# Delay Active/Inactive		40	ns	
	ST3 (CIP#)				
T54	Setup Time	25		ns	
T55	Hold Time	15		ns	
REFRESH TIMING					
	REFRESH#				
T56	Delay Active		55	ns	
T57	Float Inactive		55	ns	
	REFRESH#				Asynchronous
T58	Setup Time	15		ns	
T59	Hold Time	15		ns	
RESET TIMING					
	RST				Asynchronous
T60	Active Pulse Width	1000		ns	8 BCLKs
	RSTDRV				
T61	Delay Active		60	ns	Bus Timeout
T62	Delay Active		60	ns	Port/RST
T63	Delay Inactive		60	ns	
NMI TIMING					
	NMI				
T64	Delay Active/Inactive		60	ns	From BCLK Falling Edge
T65	Delay Act		200	ns	From OSC Rising Edge
	PARITY#, IOCHK#				Asynchronous
T66	Setup Time	15		ns	
T67	Hold Time	15		ns	
INTERRUPT TIMING					
	INT				
T68	Delay Active/Inactive		500	ns	From IRQ Active/Inactive
T69	Delay Active/Inactive		500	ns	From OSC Rising Edge
T70	Delay Active/Inactive		500	ns	From BCLK Falling Edge
COUNTER/TIMER TIMING					
	SPKR				
T71	Delay Active/Inactive		200	ns	From OSC Rising Edge
T72	Delay Active/Inactive		100	ns	From BCLK Falling Edge
	SLOWH#				
T73	Delay Active/Inactive		200	ns	
RTCALE, GTIM# TIMING					
	RTCALE				
T74	Delay Active		55	ns	
T75	Delay Inactive		40	ns	
	GT1M#				
T76	Delay Active/Inactive		50	ns	

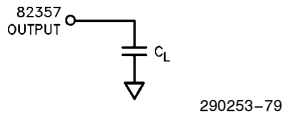
NOTE:
1. Sampled, not 100% tested.

A.C. TEST LOADS

C_L = 240 pF on RSTDRV, BE<3:0>#, REFRESH#, EOP

C_L = 50 pF on DHOLD, GT1M#, GT16M#, ST<3:0>, EMSTR16#, EXMASTER#, CSOUT#, INT, SLOWH#, NMI, DRDY, AEN#, SPKR, and RTCALE

C_L = 120 pF on all other pins (MACK<5:0>#, D<7:0>, HA<31:2>, DACK#)



290253–79

C_L includes all parasitic capacitances.

11.2 A.C. Characteristic Waveforms

In the following timing illustrations (10.2.1–10.2.10) ''SET'' refers to setup time and ''HOLD'' refers to hold time. The remaining timings are either output delay (valid, active/inactive edges) or float timings.

A.C. Testing: All non-clock Inputs are driven at 3V for a logic ''1'' and 0V for a logic ''0''. A.C. Timings referenced to a BCLK edge are measured from the 2V level on the clock to the 1.5V level on the signal under test. All signals referenced to a non-clock edge are measured from 1.5V to 1.5V; except as noted by the following:

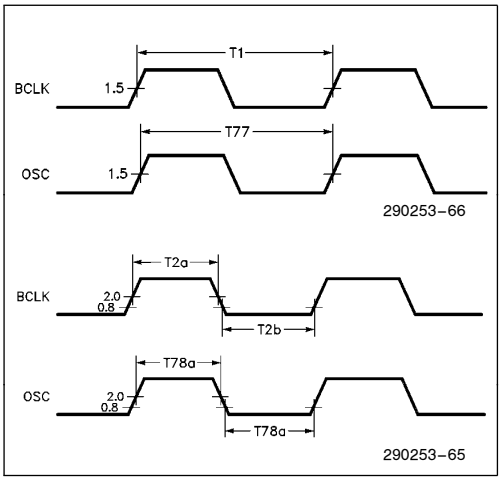
(1) BCLK high time (T2a) measurement is made at 2.0V

(2) BCLK low time (T2b) measurement is made at 0.8V

(3) OSC high time (T78a) measurement is made at 2.0V

(4) OSC low time (T78b) measurement is made at 0.8V

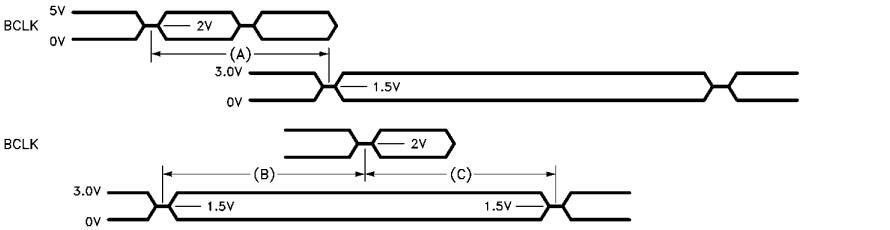
11.2.1 BCLK, OSC TIMING



290253–66

290253–65

A.C. TESTING INPUT, OUTPUT WAVEFORMS



290253–64

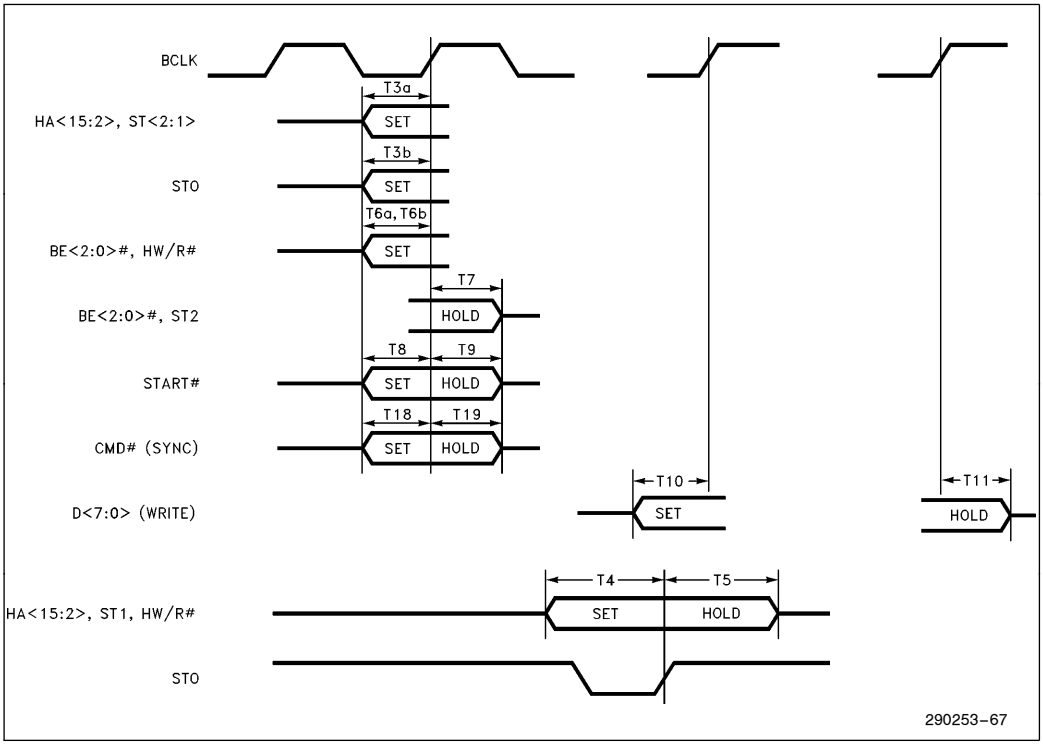
(A) Output delay spec referenced to either a BCLK rising or a BCLK falling edge.

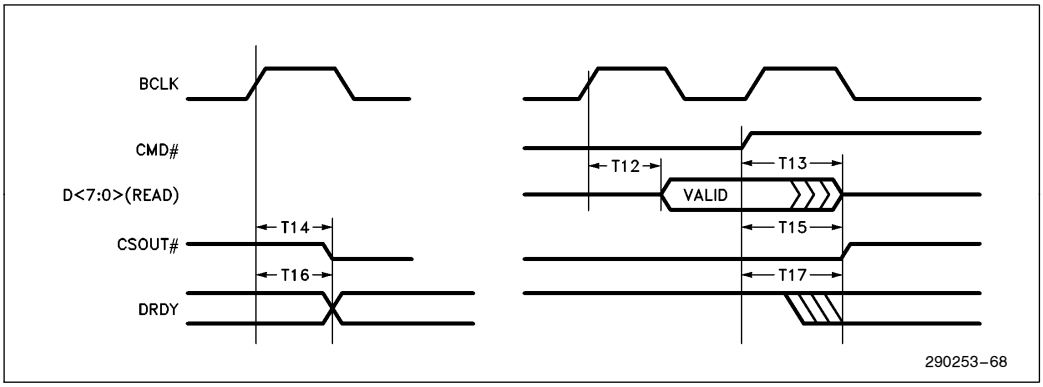
(B) Minimum input setup spec referenced to either a BCLK rising or a BCLK falling edge.

(C) Minimum input hold spec referenced to either a BCLK rising or a BCLK falling edge.

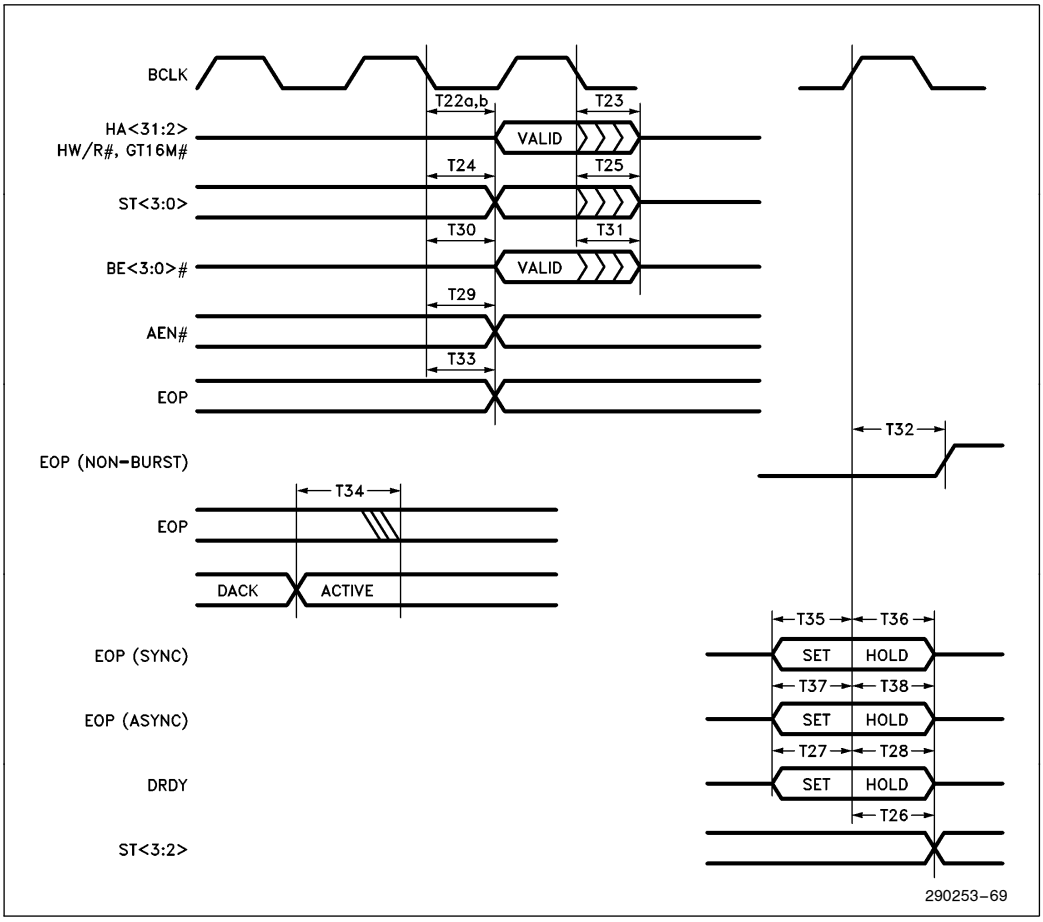
NOTE:
The input waveforms have a $t_r \leq 2.0$ ns from 0.8V to 2.0V.

11.2.2 SLAVE MODE TIMING

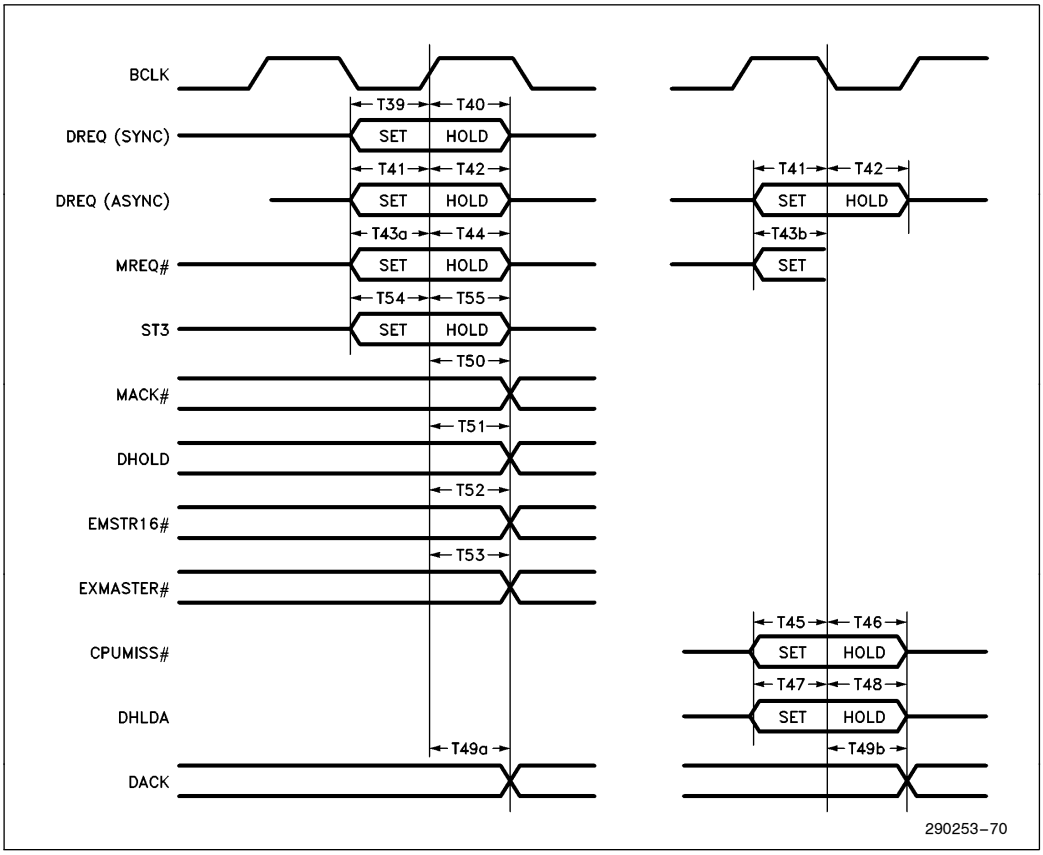




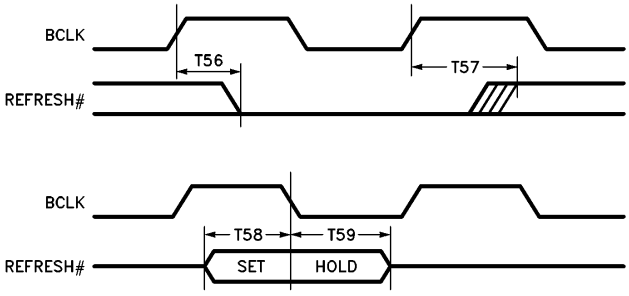
11.2.3 DMA MASTER TIMING



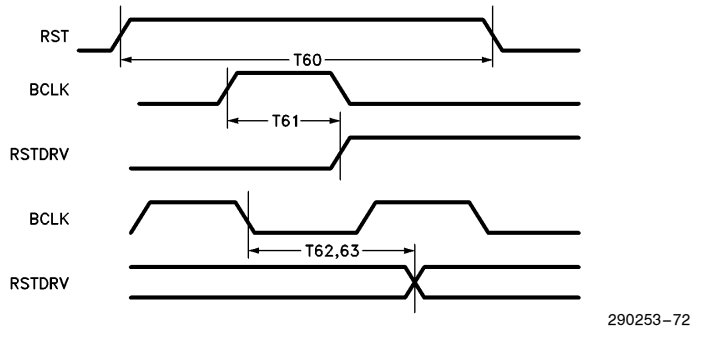
11.2.4 ARBITER TIMING



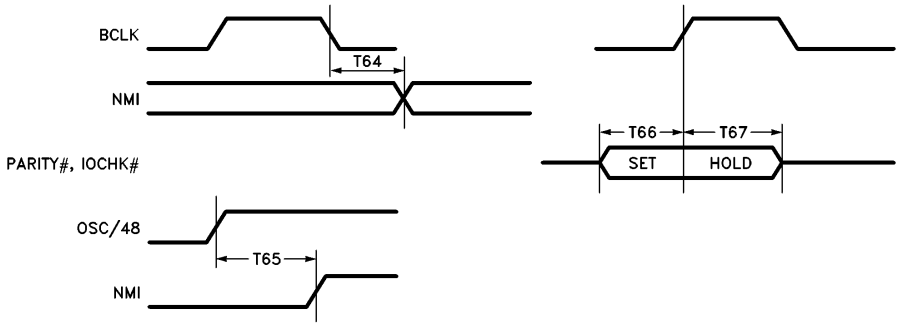
11.2.5 REFRESH TIMING



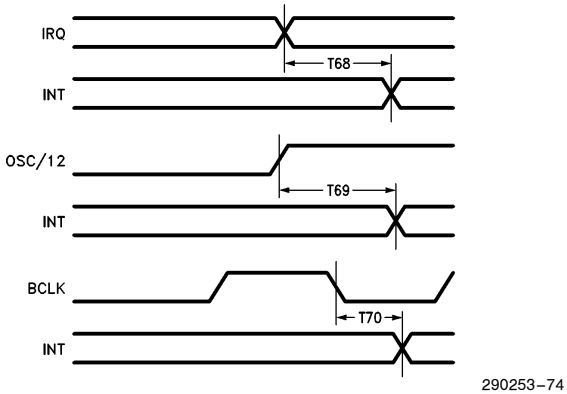
11.2.6 RESET TIMING



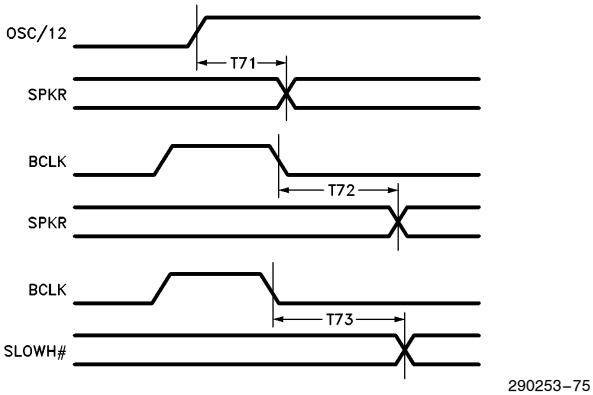
11.2.7 NMI TIMING



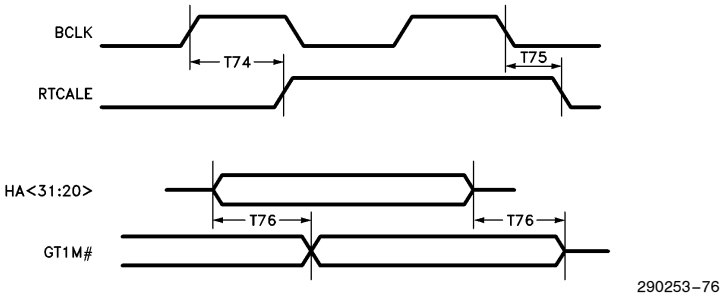
11.2.8 INTERRUPT TIMING



11.2.9 COUNTER/TIMER TIMING



11.2.10 RTCALE, GT1M# TIMING



12.0 ISP PIN AND PACKAGE INFORMATION

12.1 Signal Overview

Name	Type	Pin	Interface	Description
The following pins are used by the ISP for EISA system arbitration:				
DREQ<7:5>	I	100–102	EISA	DMA Request Lines
DREQ<3:0>	I	103–106	EISA	DMA Request Lines
DACK<7:5>#	O	83–85	EISA	DMA Acknowledge Lines
DACK<3:0>#	O	86–89	EISA	DMA Acknowledge Lines
MREQ<5:0>#	I	107–112	EISA	Master Request Lines
MACK<5:0>#	O	92–97	EISA	Master Acknowledge Lines
REFRESH#	I/O	131	EISA	Refresh Control (24 mA)
DHLDA	I	114	Host	CPU Hold Acknowledge
CPUMISS#	I	113	Host	CPU Cache Miss
DHOLD	O	98	EBC	CPU Hold Request
EMSTR16#	O	4	EBC	Early Master 16 Indication to the 82358 Bus Controller
EXMASTER#	O	5	EBC	EISA Bus Master Access Control
The following signals are used by the ISP for either DMA control or in slave mode during a register access:				
START#	I	20	EISA	Start Signal from the EISA Bus
CMD#	I	19	EISA	Command Signal from the EISA Bus
EOP	I/O	80	EISA	End of Process Input and Output
BE<3:0>#	I/O	70–67	EISA	Byte Enables (BE3 is Output Only)
HA<11:2>	I/O	55–64	Host	Address Bus
HA<14:12>	I/O	50–52	Host	Address Bus
HA<21:15>	I/O	35–41	Host	Address Bus (A16–A19 are Output Only)
HA<31:22>	I/O	23–32	Host	Address Bus
HW/R#	I/O	79	Host	Write/Read Status
GT16M#	O	11	EBC	Address Less Than 16 Mbytes
GT1M#	O	12	EBC	Address Less Than 1 Mbyte
RST	I	18	EBC	System Reset
ST<3:0>	I/O	72–75	EBC	Bidirectional Status Signals between the ISP/EBC
DRDY	I/O	78	EBC	Master Mode Ready from EBC and Slave Mode Ready to EBC Terminal Count Output (Directly Drives EISA Bus)
D<7:0>	I/O	42–49	Other	Data Bus
CSOUT#	O	9	Other	ISP Selected in Slave Mode
AEN#	O	82	Other	Address Enable Out from DMA
The following signals are used by the ISP for interrupt control:				
IRQ1	I	128	EISA	Interrupt Request Input
IRQ<15:3>	I	115–127	EISA	Interrupt Request Inputs
INT	O	3	Host	Interrupt Output to CPU

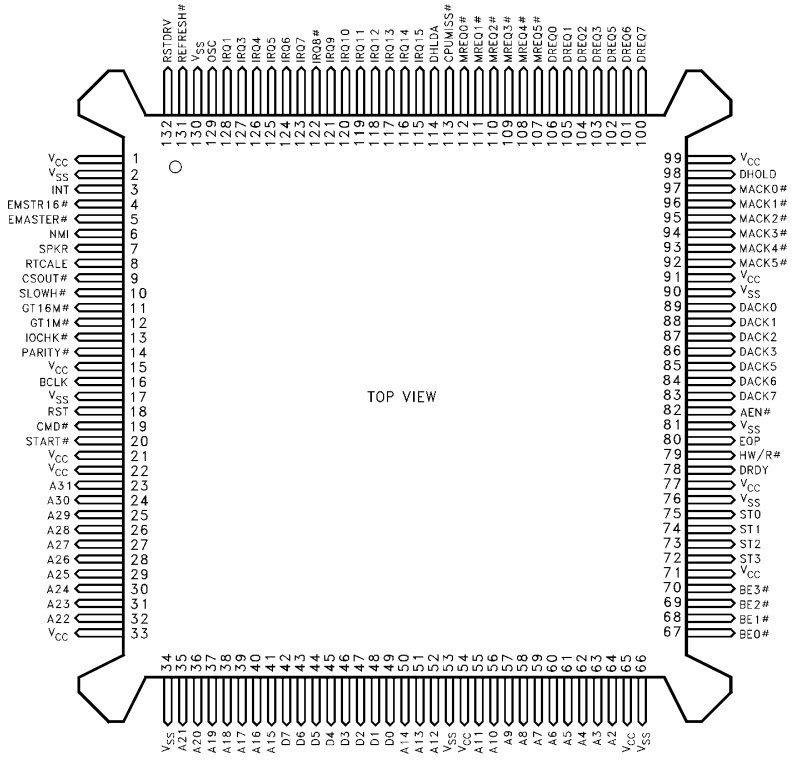
12.1 Signal Overview (Continued)

Name	Type	Pin	Interface	Description
The following signals are used by the ISP for the Timers, NMI Servicing, and Associated Logic:				
OSC	I	129	EISA	14.31818 MHz Clock for Timers
IOCHK#	I	13	EISA	Bus Error Signal
RSTDRV	O	132	EISA	System Bus Reset
PARITY#	I	14	Host	Main Memory Parity Error
SLOWH#	O	10	Host	Slow Down Timer to CPU
NMI	O	6	Host	From NMI Logic
SPKR	O	7	Other	Speaker Driver Output (24 mA Drive)
RTCALE	O	8	Other	Real Time Clock Address Latch Enable
The following pins are used for power:				
V_{CC}	1,15,21,22,33,54,65,71,77,91,99			Power
V_{SS}	2,17,34,53,66,76,81,90,130			Ground
The following pin is the clock input:				
BCLK	I	16	EBC	Operation Clock

Name = Pin Name, **Type** = Input/Output, **Pin** = Pin Location, **Interface** = EBC Interface, EISA Bus, Host Bus, or Other.

12.2 132-Pin Package

82357 Package Pin Out



290253–77

12.3 Device Pinout

Device Pinout—132 Lead PQFP

A Row		
Pin	Label	Type
1	V_{CC}	
2	V_{SS}	
3	INT	O
4	EMSTR16#	O
5	EXMASTER#	O
6	NMI	O
7	SPKR	O
8	RTCALE	O
9	CSOUT#	O
10	SLOWH#	O
11	GT16M#	O
12	GT1M#	O
13	IOCHK#	I
14	PARITY#	I
15	V_{CC}	
16	BCLK	I
17	V_{SS}	
18	RST	I
19	CMD#	I
20	START#	I
21	V_{CC}	
22	V_{CC}	
23	A31	IO
24	A30	IO
25	A29	IO
26	A28	IO
27	A27	IO
28	A26	IO
29	A25	IO
30	A24	IO
31	A23	IO
32	A22	IO
33	V_{CC}	

B Row		
Pin	Label	Type
34	V_{SS}	
35	A21	IO
36	A20	IO
37	A19	O
38	A18	O
39	A17	O
40	A16	O
41	A15	IO
42	D7	IO
43	D6	IO
44	D5	IO
45	D4	IO
46	D3	IO
47	D2	IO
48	D1	IO
49	D0	IO
50	A14	IO
51	A13	IO
52	A12	IO
53	V_{SS}	
54	V_{CC}	
55	A11	IO
56	A10	IO
57	A9	IO
58	A8	IO
59	A7	IO
60	A6	IO
61	A5	IO
62	A4	IO
63	A3	IO
64	A2	IO
65	V_{CC}	
66	V_{SS}	

C Row		
Pin	Label	Type
67	BE0#	IO
68	BE1#	IO
69	BE2#	IO
70	BE3#	O
71	V_{CC}	
72	ST3	IO
73	ST2	IO
74	ST1	IO
75	ST0	IO
76	V_{SS}	
77	V_{CC}	
78	DRDY	IO
79	HW/R#	IO
80	EOP	IO
81	V_{SS}	
82	AEN#	O
83	DACK7#	O
84	DACK6#	O
85	DACK5#	O
86	DACK3#	O
87	DACK2#	O
88	DACK1#	O
89	DACK0#	O
90	V_{SS}	
91	V_{CC}	
92	MACK5#	O
93	MACK4#	O
94	MACK3#	O
95	MACK2#	O
96	MACK1#	O
97	MACK0#	O
98	DHOLD	O
99	V_{CC}	

D Row		
Pin	Label	Type
100	DREQ7	I
101	DREQ6	I
102	DREQ5	I
103	DREQ3	I
104	DREQ2	I
105	DREQ1	I
106	DREQ0	I
107	MREQ5#	I
108	MREQ4#	I
109	MREQ3#	I
110	MREQ2#	I
111	MREQ1#	I
112	MREQ0#	I
113	CPUMISS#	I
114	DHLDA	I
115	IRQ15	I
116	IRQ14	I
117	IRQ13	I
118	IRQ12	I
119	IRQ11	I
120	IRQ10	I
121	IRQ9	I
122	IRQ8#	I
123	IRQ7	I
124	IRQ6	I
125	IRQ5	I
126	IRQ4	I
127	IRQ3	I
128	IRQ1	I
129	OSC	I
130	V_{SS}	
131	REFRESH#	IO
132	RSTDRV	O

13.0 ISP PORT ADDRESS / I/O DECODING

Address Bits					Register Type	Device
Port	FEDC	BA98	7654	3210		
0000h	0000	0000	000X	0000	rw	DMA1 CH-0 Base and Current Address
0001h	0000	0000	000X	0001	rw	DMA1 CH-0 Base and Current Count
0002h	0000	0000	000X	0010	rw	DMA1 CH-1 Base and Current Address
0003h	0000	0000	000X	0011	rw	DMA1 CH-1 Base and Current Count
0004h	0000	0000	000X	0100	rw	DMA1 CH-2 Base and Current Address
0005h	0000	0000	000X	0101	rw	DMA1 CH-2 Base and Current Count
0006h	0000	0000	000X	0110	rw	DMA1 CH-3 Base and Current Address
0007h	0000	0000	000X	0111	rw	DMA1 CH-3 Base and Current Count
0008h	0000	0000	000X	1000	rw	DMA1 status(r) command (w) register
0009h	0000	0000	000X	1001	w	DMA1 Write Request register (w) (reserved) (r)
000Ah	0000	0000	000X	1010	w	DMA1 Write single mask bit (w) (reserved) (r)
000Bh	0000	0000	000X	1011	w	DMA1 Write Mode register (w) (reserved) (r)
000Ch	0000	0000	000X	1100	w	DMA1 Clear byte pointer F/F (w) (reserved) (r)
000Dh	0000	0000	000X	1101	w	DMA1 Master Clear (w) (reserved) (r)
000Eh	0000	0000	000X	1110	w	DMA1 Clear Mask register (w) (reserved) (r)
000Fh	0000	0000	000X	1111	rw	DMA1 Write all mask register bits (w) DMA1 Read all mask register bits (r)
0020h	0000	0000	001X	XX00	rw	INT-1 control register
0021h	0000	0000	001X	XX01	rw	INT-1 mask register
0040h	0000	0000	010X	0000	rw	Programmable Interval Timer 1, 82C54 System Clock (Counter 0)
0041h	0000	0000	010X	0001	rw	Refresh Request (Counter 1)
0042h	0000	0000	010X	0010	rw	Speaker Tone (Counter 2)
0043h	0000	0000	010X	0011	w	Command Mode Register
0048h	0000	0000	010X	1000	rw	Programmable Interval Timer 2, 82C54 Failsafe Clock (Counter 0)
0049h	0000	0000	010X	1001	rw	(reserved)
004Ah	0000	0000	010X	1010	rw	CPU Speed Control (Counter 2)
004Bh	0000	0000	010X	1011	w	Command Mode Register
0061h	0000	0000	0110	0X01	rw	NMI Status (See NMI Interrupts)
0070h	0000	0000	0111	0XX0	w	NMI Enable Register (bit 7 = 0) (w) (reserved) (r)

NOTE:
The above reserved registers (9h–Eh) when read, will return an indeterminate value. Reserved register 49h can not be read or written. When read, 49h will return FFh.

13.0 ISP PORT ADDRESS / I/O DECODING (Continued)

Address Bits					Register Type	Device
Port	FEDC	BA98	7654	3210		
0080h	0000	0000	100X	0000	rw	DMA Page Register Reserved
0081h	0000	0000	100X	0001	rw	DMA Page Register CH-2 Page
0082h	0000	0000	100X	0010	rw	DMA Page Register CH-3 Page
0083h	0000	0000	100X	0011	rw	DMA Page Register CH-1 Page
0084h	0000	0000	100X	0100	rw	DMA Page Register Reserved
0085h	0000	0000	100X	0101	rw	DMA Page Register Reserved
0086h	0000	0000	100X	0110	rw	DMA Page Register Reserved
0087h	0000	0000	100X	0111	rw	DMA Page Register CH-0 Page
0088h	0000	0000	100X	1000	rw	DMA Page Register Reserved
0089h	0000	0000	100X	1001	rw	DMA Page Register CH-6 Page
008Ah	0000	0000	100X	1010	rw	DMA Page Register CH-7 Page
008Bh	0000	0000	100X	1011	rw	DMA Page Register CH-5 Page
008Ch	0000	0000	100X	1100	rw	DMA Page Register Reserved
008Dh	0000	0000	100X	1101	rw	DMA Page Register Reserved
008Eh	0000	0000	100X	1110	rw	DMA Page Register Reserved
008Fh	0000	0000	100X	1111	rw	DMA Page Register Refresh Page
00A0h	0000	0000	101X	XX00	rw	INT-2 control register
00A1h	0000	0000	101X	XX01	rw	INT-2 mask register
00C0h	0000	0000	1100	000X	rw	DMA2 CH-0 Base and Current Address
00C2h	0000	0000	1100	001X	rw	DMA2 CH-0 Base and Current Count
00C4h	0000	0000	1100	010X	rw	DMA2 CH-1 Base and Current Address
00C6h	0000	0000	1100	011X	rw	DMA2 CH-1 Base and Current Count
00C8h	0000	0000	1100	100X	rw	DMA2 CH-2 Base and Current Address
00CAh	0000	0000	1100	101X	rw	DMA2 CH-2 Base and Current Count
00CCh	0000	0000	1100	110X	rw	DMA2 CH-3 Base and Current Address
00CEh	0000	0000	1100	111X	rw	DMA2 CH-3 Base and Current Count
00D0h	0000	0000	1101	000X	rw	DMA2 status(r) command(w) register
00D2h	0000	0000	1101	001X	w	DMA2 Write Request register (w) (reserved) (r)
00D4h	0000	0000	1101	010X	w	DMA2 Write single mask bit (w) (reserved) (r)
00D6h	0000	0000	1101	011X	w	DMA2 Write Mode register (w) (reserved) (r)
00D8h	0000	0000	1101	100X	w	DMA2 Clear byte pointer F/F (w) (reserved) (r)
00DAh	0000	0000	1101	101X	w	DMA2 Master Clear (w) (reserved) (r)
00DCh	0000	0000	1101	110X	w	DMA2 Clear Mask register (w) (reserved) (r)
00DEh	0000	0000	1101	111X	rw	DMA2 Write all mask register bits (w) DMA2 Read all mask register bits (r)

NOTE:
Reserved registers (0084h–0086h and 008Ch–008Eh) can be read or written. Reserved registers (00D2h–00DCh) when read, will return an indeterminate value.

13.0 ISP PORT ADDRESS / I/O DECODING (Continued)

Address Bits					Register Type	Device
Port	FEDC	BA98	7654	3210		
0400h	0000	0100	0000	0000	rw	(reserved)
0401h	0000	0100	0000	0001	rw	DMA1 CH-0 Base/Current count high
0402h	0000	0100	0000	0010	rw	(reserved)
0403h	0000	0100	0000	0011	rw	DMA1 CH-1 Base/Current count high
0404h	0000	0100	0000	0100	rw	(reserved)
0405h	0000	0100	0000	0101	rw	DMA1 CH-2 Base/Current count high
0406h	0000	0100	0000	0110	rw	(reserved)
0407h	0000	0100	0000	0111	rw	DMA1 CH-3 Base/Current count high
0408h	0000	0100	0000	1000	rw	(reserved)
0409h	0000	0100	0000	1001	rw	(reserved)
040Ah	0000	0100	0000	1010	rw	DMA1 Set Chaining mode (w) Interrupt Status (r)
040Bh	0000	0100	0000	1011	rw	DMA1 Ext Write Mode register (w) (reserved) (r)
040Ch	0000	0100	0000	1100	ro	Chain Buffer Control Register
040Dh	0000	0100	0000	1101	ro	Stepping Level register
040Eh	0000	0100	0000	1110	rw	ISP Test Register (reserved)
040Fh	0000	0100	0000	1111	rw	ISP Test Register (reserved)
0461h	0000	0100	0110	0001	rw	Extended NMI and reset control
0462h	0000	0100	0110	0010	wo	NMI I/O interrupt port (casual)
0464h	0000	0100	0110	0100	ro	Last 32-bit bus master granted (L)
0480h	0000	0100	1000	0000	rw	(reserved)
0481h	0000	0100	1000	0001	rw	DMA High Page Register CH-2 Page
0482h	0000	0100	1000	0010	rw	DMA High Page Register CH-3 Page
0483h	0000	0100	1000	0011	rw	DMA High Page Register CH-1 Page
0484h	0000	0100	1000	0100	rw	(reserved)
0485h	0000	0100	1000	0101	rw	(reserved)
0486h	0000	0100	1000	0110	rw	(reserved)
0487h	0000	0100	1000	0111	rw	DMA High Page Register CH-0 Page
0488h	0000	0100	1000	1000	rw	(reserved)
0489h	0000	0100	1000	1001	rw	DMA High Page Register CH-6 Page
048Ah	0000	0100	1000	1010	rw	DMA High Page Register CH-7 Page
048Bh	0000	0100	1000	1011	rw	DMA High Page Register CH-5 Page
048Ch	0000	0100	1000	1100	rw	(reserved)
048Dh	0000	0100	1000	1101	rw	(reserved)
048Eh	0000	0100	1000	1110	rw	(reserved)
048Fh	0000	0100	1000	1111	rw	DMA High Page Register Refresh
04C2h	0000	0100	1100	0010	rw	(reserved)
04C6h	0000	0100	1100	0110	rw	DMA2 CH-5 Base/Current count high
04CAh	0000	0100	1100	1010	rw	DMA2 CH-6 Base/Current count high
04CEh	0000	0100	1100	1110	rw	DMA2 CH-7 Base/Current count high

NOTE:
When read from or written to, all of the above reserved registers (except for 40Ah and 40Bh) will not respond and CSOUT# will not be asserted. Reserved registers 40Ah and 40Bh when read, will return an indeterminate value.

13.0 ISP PORT ADDRESS / I/O DECODING (Continued)

Address Bits					Register Type	Device
Port	FEDC	BA98	7654	3210		
04D0h	0000	0100	1101	0000	rw	INT-1 Edge Level Control Register
04D1h	0000	0100	1101	0001	rw	INT-2 Edge Level Control Register
04D2h	0000	0100	1101	0010	rw	(reserved)
04D3h	0000	0100	1101	0011	rw	(reserved)
04D4h	0000	0100	1101	0100	rw	DMA2 Set Chaining mode (w) Chaining mode (r)
04D5h	0000	0100	1101	0101	rw	(reserved)
04D6h	0000	0100	1101	0110	rw	DMA2 Ext Write Mode register (w) (reserved) (r)
04D7h	0000	0100	1101	0111	rw	(reserved)
04D8h	0000	0100	1101	1000	rw	(reserved)
04D9h	0000	0100	1101	1001	rw	(reserved)
04DAh	0000	0100	1101	1010	rw	(reserved)
04DBh	0000	0100	1101	1011	rw	(reserved)
04DCh	0000	0100	1101	1100	rw	(reserved)
04DDh	0000	0100	1101	1101	rw	(reserved)
04DEh	0000	0100	1101	1110	rw	(reserved)
04DFh	0000	0100	1101	1111	rw	(reserved)
04E0h	0000	0100	1110	0000	rw	DMA CH-0 Stop Reg Bits <7:2>
04E1h	0000	0100	1110	0001	rw	DMA CH-0 Stop Reg Bits <15:8>
04E2h	0000	0100	1110	0010	rw	DMA CH-0 Stop Reg Bits <23:16>
04E3h	0000	0100	1110	0011	rw	(reserved)
04E4h	0000	0100	1110	0100	rw	DMA CH-1 Stop Reg Bits <7:2>
04E5h	0000	0100	1110	0101	rw	DMA CH-1 Stop Reg Bits <15:8>
04E6h	0000	0100	1110	0110	rw	DMA CH-1 Stop Reg Bits <23:16>
04E7h	0000	0100	1110	0111	rw	(reserved)
04E8h	0000	0100	1110	1000	rw	DMA CH-2 Stop Reg Bits <7:2>
04E9h	0000	0100	1110	1001	rw	DMA CH-2 Stop Reg Bits <15:8>
04EAh	0000	0100	1110	1010	rw	DMA CH-2 Stop Reg Bits <23:16>
04EBh	0000	0100	1110	1011	rw	(reserved)
04ECh	0000	0100	1110	1100	rw	DMA CH-3 Stop Reg Bits <7:2>
04EDh	0000	0100	1110	1101	rw	DMA CH-3 Stop Reg Bits <15:8>
04EEh	0000	0100	1110	1110	rw	DMA CH-3 Stop Reg Bits <23:16>
04EFh	0000	0100	1110	1111	rw	(reserved)
04F0h	0000	0100	1111	0000	rw	(reserved)
04F1h	0000	0100	1111	0001	rw	(reserved)
04F2h	0000	0100	1111	0010	rw	(reserved)
04F3h	0000	0100	1111	0011	rw	(reserved)
04F4h	0000	0100	1111	0100	rw	DMA CH-5 Stop Reg Bits <7:2>
04F5h	0000	0100	1111	0101	rw	DMA CH-5 Stop Reg Bits <15:8>
04F6h	0000	0100	1111	0110	rw	DMA CH-5 Stop Reg Bits <23:16>
04F7h	0000	0100	1111	0111	rw	(reserved)
04F8h	0000	0100	1111	1000	rw	DMA CH-6 Stop Reg Bits <7:2>
04F9h	0000	0100	1111	1001	rw	DMA CH-6 Stop Reg Bits <15:8>
04FAh	0000	0100	1111	1010	rw	DMA CH-6 Stop Reg Bits <23:16>
04FBh	0000	0100	1111	1011	rw	(reserved)
04FCh	0000	0100	1111	1100	rw	DMA CH-7 Stop Reg Bits <7:2>
04FDh	0000	0100	1111	1101	rw	DMA CH-7 Stop Reg Bits <15:8>
04FEh	0000	0100	1111	1110	rw	DMA CH-7 Stop Reg Bits <23:16>
04FFh	0000	0100	1111	1111	rw	(reserved)

NOTE:
When read from or written to, reserved registers (4D2h, 4D3h, 4D5h, and 4D7h–4DFh) will not respond and CSOUT# will not be asserted. Reserved registers (4E3h, 4EFh, 4EBh, 4EFh–4F3h, 4F7h, 4FBh, and 4FFh when read, will return a value of FFh. Reserved register 4D6h when read, will return an indeterminate value.

14.0 THERMAL SPECIFICATION

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

The PQFP case temperature should be measured at the center of the top surface opposite the pins, as shown in Figure 14-1 below.

Process Name: CHMOSIV

Temperature Rise Vs. Time:

Time (sec)	Temperature Rise Unit Power (°C/w)
0	0
1	3.7
2	6.0
3	7.3
5	9.5

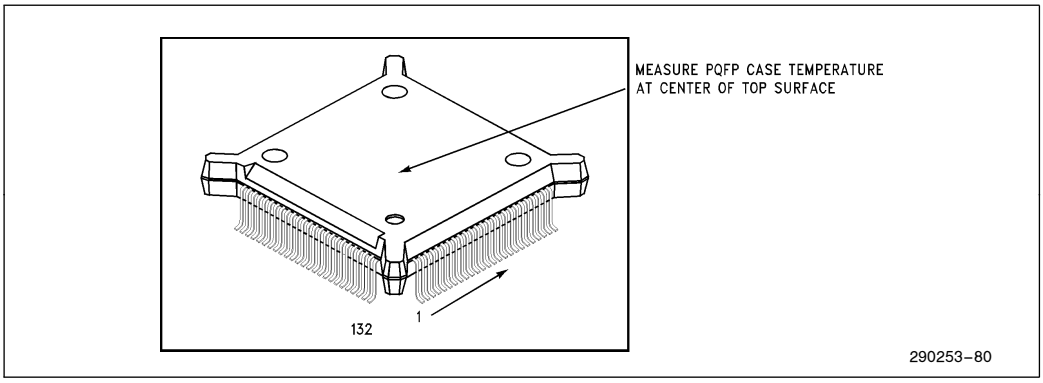


Figure 14-1. Plastic Quad Flat Pack (PQFP)

Table 14-1. 82357 PQFP Package Thermal Characteristics

Thermal Resistance—°C/Watt					
Parameter	**Air Flow Rate (Ft/Min)**				
	0	**200**	**400**	**600**	**800**
θ Junction to Case	7	7	7	7	7
θ Case to Ambient	22	17.5	14.5	12	10

NOTES:
1. Table 14-1 applies to the 82357 PQFP plugged into a socket or soldered directly on to the board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$.

15.0 REVISION HISTORY

This data sheet contains updates and improvements between versions -003 and -004. A revision summary is listed here for your convenience.

82357 Integrated System Peripheral (ISP)

Highlights Reference was added to 82350DT.

Section 3.7.1 Text for I/O writes was revised.

Section 11.1 t10 is 15 ns minimum. t30 is 2 ns min. t43a is 16 ns minimum.

Note 1 was deleted from t22b, and added to t23.

Section 12.4 Add thermal specification.

This data sheet contains updates and improvements between versions -004 and -005. A revision summary is listed here for your convenience.

82357 Revision Summary

Section 1.0	In Figure 1-1 ISP System Interface, the REFRESH# pin was corrected to indicate I/O function.
Section 2.3	In 2nd paragraph in left-handed column, EXMASTER16# was changed to EMSTR16#.
Section 7.1.3	In Figure 7-4, Status Byte format, Returned Status bit changed. Bit 6 should be 0=CR contents are moved into CE. 1=CR contents not moved into CE.
Section 8.1	In CPUMISS# pin description, deleted the line, ''This line may be tied active in a cache-less system.'' Added ''This signal should not be generated to the ISP when the CPU or cache own the bus (i.e., when HLDA is inactive).''
	In EXMASTER# signal description, the last sentence changed to read ''This signal is driven active and inactive with MACK# on the rising edge of BCLK, except in the case of PREEMPT. In this case, EXMASTER# is negated on the same rising BCLK edge that CIP# (ST3) is sampled active.'' The last sentence was not changed.
Section 9.1.2, 9.1.3	Figure 9.1.2 IORD# and IOWR# were changed to IORC# and IOWC#, and ST0 was changed to be LOW (i.e., 0) during entire diagram.
Section 9.3	Corrected CHRDY to deassert after the falling edge during the eighth BCLK.
Section 11.1	T13 had Note 1 added. t43a changed from 16 ns to 17 ns min.
Section 13.0	040CH device description changed to ''chain buffer control register'' (3) 04E0h should reference bits <7:2> **NOT** <7:3>.

82357 Revision Summary

The following changes have been made since revision 005:

Section 8.1 REFRESH# paragraph heading corrected to read ''REFRESH# I/O OPEN COLLECTOR (REFRESH)''.

Section 10.2 Test Conditions for Symbol V_{OL3} has been corrected to $I_{OL} = 4$ mA(1).

Section 14.0 Table 14-1 note 2 has been corrected to read $\theta_{JA} = \theta_{JC} + \theta_{CA}$.