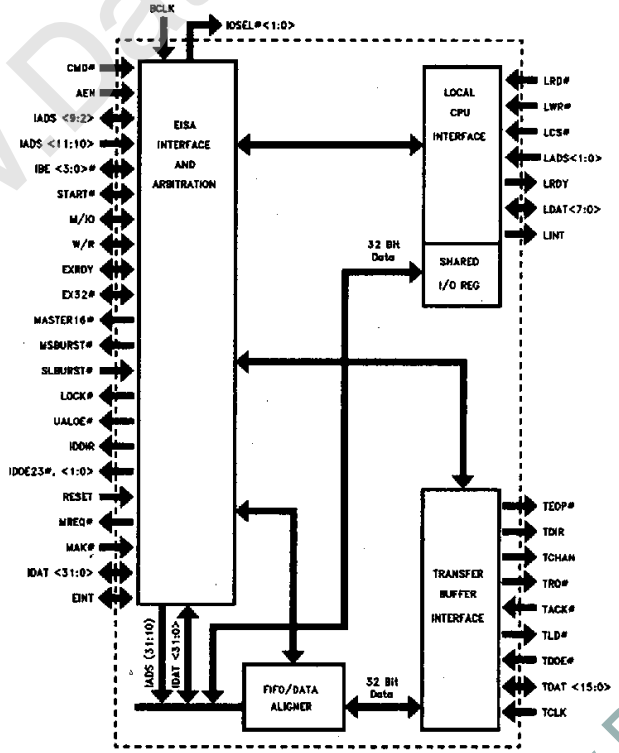
82355
BUS MASTER INTERFACE CONTROLLER (BMIC)

- Designed for use in 32-Bit EISA Bus Master Expansion Board Designs
  - — Integrates Three Interfaces (EISA, Local CPU, and Transfer Buffer)
- Supports 16- and 32-Bit Burst Transfers
  - — 33 Mbytes/Sec Maximum Data Transfers
- Supports 32-Bit Non-Burst and Mismatched Data Size Transfers
- Supports 32-Bit EISA Addressability (4 Gigabyte)
- Two independent Data Transfer Channels with 24-Byte FIFOs
  - — Expansion Board Timing and EISA Timing Operate Asynchronously
- Supports Peek/Poke Operation with the Ability to Access Individual Locations in EISA Memory or I/O space
- Automatically Handles Misaligned Doubleword Data Transfers with No Performance Penalty
- Supports Automatic Handling of Complete EISA Bus Master Protocol
  - — EISA Arbitration/Preemption
  - — Cycle Timing and Execution
  - — Byte Alignment
  - — 1K Boundary Detection
- Supports Local Data Transfer Protocol Similar to Traditional DMA
- Supports a General Purpose Command and Status Interface
  - — Local and EISA System Interrupt Support
  - — General Purpose Information Transfers
  - — Set-and-Test-Functions in I/O Space (Semaphore Function)
  - — Supports the EISA Expansion Board ID Function
- Supports Decode of Slot Specific and General I/O Addresses
- 132-Pin JEDEC PQFP Package

  (See Packaging Specification Order #240800, Package Type NG)

82355 Internal Block Diagram



290255-1

May 1994
Order Number: 290255-008

82355 Bus Master Interface Controller (BMIC)

CONTENTS

1

CONTENTS

PAGE

CONTENTS

1.0 INTRODUCTION

The 82355 Bus Master Interface Controller (BMIC) is a highly integrated Bus Master designed for use in 32-Bit EISA Bus Master expansion board designs and supports all of the enhancements defined in the EISA specifications required for EISA bus master applications. The BMIC provides a simple, yet, powerful and flexible interface between the functions on the expansion board and the EISA bus. With the help of external buffer devices, the BMIC provides all EISA control, address, and data signals necessary to interface to the EISA bus.

The primary function of the 82355 is to support 16- and 32-bit burst data transfers between functions on the EISA expansion board and the EISA bus. Data transfer rates of up to 33 Mbytes/sec are supported (the fastest transfer rate available on an EISA bus). The following logic on the BMIC supports efficient burst transfers:

- Arbitration logic, for gaining control of the EISA bus
- Two transfer-address and byte counters
- Two data FIFOs, which allow expansion board and EISA bus timing to operate asynchronously
- Data shifters, which align data to specific byte boundaries
- A transfer buffer interface, for the data transfers on the expansion board
- General-purpose command and status interface logic
- Local processor interface, to allow programming by an on-board processor
- EISA slave interface, to allow communication with the EISA system

The BMIC greatly simplifies the design of EISA expansion boards. With the 82355, a board can be implemented with simple logic similar to that used in traditional ISA DMA designs. The EISA standard allows designs with 32-bit data and address buses, burst transfers, and automatic handling of the full EISA bus master protocol.

To maximize system throughput, the 82355 BMIC incorporates three fully concurrent interfaces: EISA interface, Transfer Buffer interface, and Local Processor interface. The EISA interface incorporates two 24-byte FIFOs, and implements the full EISA protocol. The Transfer Buffer interface is optimized for high speed static RAM buffers, and can operate at a maximum frequency of 20 MHz. The Local Processor interface supports a generic slave interface, and allows the local processor to fully program the BMIC for operation. Local processors are supported with the ability to access individual locations in system memory or I/O space; this peek-and-poke feature allows the expansion board to communicate easily with other devices in the system. All three interfaces can operate simultaneously, thus maximizing overall system performance.

Address-generation support for the data transfer buffer logic on the expansion board is provided on-chip. The transfer logic on the expansion board can use a high-speed asynchronous transfer clock. The BMIC handles all synchronization with the EISA bus. A FIFO within the BMIC eliminates performance degradation on burst transfers caused by synchronization delays. The BMIC also provides a set of programmable address comparators that drive external chip selects on the expansion board to assist local devices in decoding I/O address ranges.

1.1 BMIC Terminology/Definitions

EISA BUS MASTER—A 32- or 16-bit device that uses the extended part of the EISA bus to generate memory or I/O cycles.

Downshifting Bus Master—A "downshifting" master is a 32-bit master which can convert to a 16-bit master "on the fly". The BMIC will only downshift from a 32-bit master to a 16-bit master if programmed for burst mode (refer to Section 4.2.1).

EISA READ—A data transfer (burst, non-burst (two BCLK), or mismatched) from system to the expansion board across transfer channel 1.

EISA WRITE—A data transfer (burst, non-burst (two BCLK), or mismatched) from the expansion board to system memory across one of the two transfer channels.

I/O ADDRESS DECODE SUPPORT—Refers to slot specific or general I/O address decoding.

Slot Specific Address Decoding—Refers to the decoding of unique addresses allocated to EISA slot specific expansion boards. These addresses are: X000h–X0FFh, X400h–X4FFh, X800h–X8FFh, and XC00h–XCFFh, where X represents the EISA slot number. EISA slot number "0" is reserved for the EISA system board.

General I/O Address Decoding—Refers to the decoding of addresses allocated to ISA expansion boards. These addresses are: 0100h–03FFh.

LOCAL PROCESSOR—A processor located on the expansion board.

SYSTEM CPU—Processor located on the motherboard.

SYSTEM MEMORY—Memory located on the EISA bus or motherboard.

TRANSFER INTERRUPTION—A transfer interruption is defined as an occurrence resulting in a break in a transfer caused by one of the following conditions: A FIFO pause, a FIFO stall, a channel preemption, a channel clear or suspension, a 1K page break, or a transfer complete (EOP).

FIFO Pause—This is a condition where the EISA bus does not provide or take data at a rate fast enough to keep up with the expansion board transfer buffer logic. During an EISA read, this condition is defined as an empty FIFO. During an EISA write, this condition is defined as a full FIFO. A FIFO pause is considered a preferred condition and under normal operations should occur frequently. A FIFO pause will result in the BMIC negating TRQ# until the FIFO becomes not full during an EISA write or not empty during an EISA read.

FIFO Stall—This is a condition where the transfer buffer logic on the expansion board does not provide or take data at a rate fast enough to keep up with the EISA bus. During an EISA read, this condition is defined as a full FIFO. During an EISA write, this condition is defined as an empty FIFO. Under normal operations, a FIFO stall is expected to be a rare and exceptional event. For additional information regarding a FIFO stall, refer to Section 6.2.

Channel Clear—A channel clear results in the immediate termination of the current transfer and the flushing of the channel's corresponding FIFO. A channel clear is initiated by setting the CFGCL bit in the corresponding channel's Configuration register to a 1. For additional information regarding channel clear, refer to Section 8.2.4.2.

Channel Suspension—This temporarily prevents a channel from proceeding with a transfer. A transfer can be temporarily suspended by setting the CFGSU bit in the corresponding channel's Configuration register to a 1.

Channel Preemption—The BMIC can be preempted from the EISA bus by the 82357 (ISP). The 82357 negates MAK# indicating to the BMIC that it must finish the current bus cycle and relinquish control of the EISA bus by negating MREQ# within 64 BCLK periods. The BMIC is programmable to relinquish the bus within 0, 32, or 64 BCLKs from the negation of MAK# (refer to Section 4.4.2).

1K Page Break—The temporary termination of a burst, non-burst (two BCLK), or mismatched data transfer due to a 1K page address boundary crossing (refer to Section 4.2.2).

Transfer Complete (EOP)—End of process due to the transfer byte count being exhausted or a channel being cleared (channel clear). A transfer complete (EOP) will result in the BMIC asserting TEOP# with the last cycle (refer to Section 5.4).

TRANSFER BUFFER LOGIC—Logic located on the expansion board used to support the transfer and storage of data during BMIC EISA master mode transfers between the expansion board and system memory.

The transfer buffer logic interfaces to the Transfer Buffer Interface of the BMIC. Refer to Section 5.2 for additional information regarding transfer buffer logic.

2.0 BMIC INTERFACE ILLUSTRATION

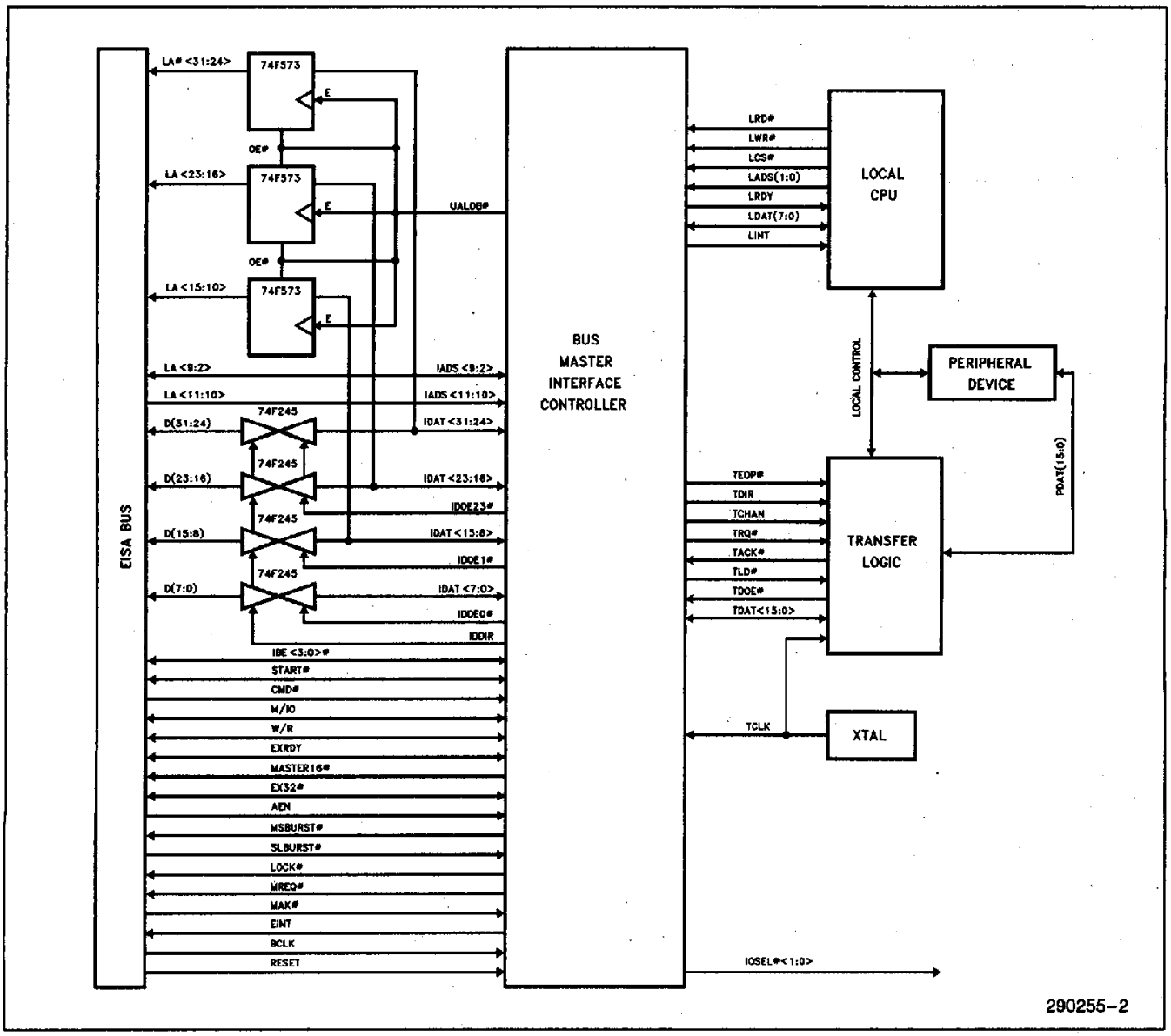


Figure 2-1. BMIC System Interface

3.0 FUNCTIONAL OVERVIEW

The following is a brief discussion of the functional blocks and features of the 82355. The EISA interface, Transfer Buffer interface, FIFO/Data Aligner, and Local interface each have a corresponding detailed section later in this data sheet.

3.1 EISA Master and EISA Slave Operations

In EISA slave mode, the 82355 monitors the EISA address lines <11:2> for general I/O address decoding, slot-specific address decoding, and Shared register accessing. During slave mode operations, all internal registers are accessible through the Local Processor interface, and all Shared registers are accessible through either the Local Processor interface or the EISA interface of the BMIC.

In EISA master mode, the 82355 becomes the master of the EISA bus. It may perform burst, non-burst (two BCLK), mismatched, or Peek/Poke data transfers at this time. During master mode operations, all internal registers are accessible through the Local Processor interface of the BMIC.

The arbiter portion of the BMIC determines which mode the device is in, performs the EISA arbitration, and provides the control signals necessary to regulate the slave and master activities internal to the chip. In slave mode, the arbiter also mediates between the EISA side and the local side during Shared register accesses.

The following is a table of the functions that can be performed during master and slave operations:

	Shared Reg. Accessing	Local CPU Only Reg. Accessing	EISA I/O Address Decoding	Data Transfers
EISA Slave Mode	YES (1, 2)	YES	YES	NO
EISA Master Mode	YES (2)	YES	NO	YES

NOTE:
Shared Reg. Accessing refers to the registers that are accessible through either the EISA interface or Local Processor interface.
Local Processor Only Reg. Accessing refers to the registers that are accessible through the Local Processor interface only.
EISA I/O Address Decoding refers to either general or slot specific I/O decoding support for the expansion board.
Data Transfers refer to either burst, non-burst (two BCLK), mismatched, or peek/poke data transfers.
YES = Can Be Performed
NO = Can Not Be Performed
1 = EISA interface
2 = Local interface

3.2 82355 Internal Architecture Description

The 82355 contains four blocks of control logic. The EISA interface block, Transfer Buffer interface block, FIFO/Data Aligner block, and the Local Processor interface block.

3.2.1 EISA INTERFACE BLOCK

The EISA interface block provides the following functions:

— generates the 32-bit EISA address for burst, non-burst (two BCLK), and peek/poke data transfers

— generates the EISA control signals necessary to implement an EISA 16-bit or 32-bit bus master, and a 32-bit EISA slave

— generates the control signals necessary to enable and disable the external buffer devices

— performs the EISA arbitration and provides the internal control signals required to regulate the slave and master activities of the BMIC

— integrates the registers necessary for the above operations as well as the registers required to provide the configuration and status of the data transfers between the EISA bus and the memory buffer on the expansion board

The EISA memory address range of the 82355 covers the 4 Gigabytes and supports the detection of 1K page address boundaries during burst, non-burst (two BCLK), and mismatched data cycles to and from system memory.

During slave mode, the EISA interface also supports slot specific and general I/O address decode necessary for Shared Register accesses and general decode as required by the expansion board. The shared register addresses are mapped into the slot specific I/O range (C80h-C9Fh).

The EISA interface block contains 43 registers necessary to execute the above functions. A detailed description of the registers and their functions can be found under Register Description (Sections 8.1 and 8.2).

3.2.2 TRANSFER BUFFER INTERFACE BLOCK

The Transfer Buffer interface block provides the group of signals that are required to perform 16-bit data transfers to and from the memory buffer on the expansion board. The protocol used is similar to that found in standard DMA designs. The interface includes a 16-bit data bus (TDAT), seven control signals and a transfer clock (TCLK). The transfer clock can run completely asynchronous to the EISA BCLK signal.

The Transfer Buffer interface block also provides a 16-bit transfer start address which is generated at the beginning of all new data transfers to and from the memory buffer on the expansion board. The 16 TDAT data lines are used to transfer the address.

The Transfer Buffer interface block contains eight registers. A detailed description of the registers and their functions can be found under Register Description (Section 8.2).

3.2.3 FIFO/DATA ALIGNER BLOCK

The FIFO/Data Aligner block is used to isolate and simplify the timing relationships between the EISA bus and the bus master expansion board. This allows the transfer buffer logic and EISA bus timing to operate asynchronously. The FIFO provides the data channel between the EISA bus and the expansion board during BMIC master data transfers and the Data Aligner provides the byte alignment and assembly necessary for the EISA bus.

There are two dual-port, six doubleword wide (24 byte) FIFOs on-board, one per transfer channel. The data is written into the FIFO from either the EISA bus side or the expansion board side, depending on the direction of the transfer. The transfer direction is controlled by a bit in the Transfer Base Count register set.

3.2.4 LOCAL PROCESSOR INTERFACE BLOCK

The Local Processor interface block provides the interface between the BMIC and the local processor. If a local processor is not present, the processor interface can be connected to the ISA bus. The Local Processor interface block is based on an 8086 style slave mode and provides an 8-bit data path for BMIC programming. All of the BMICs internal registers are accessible through this interface.

The Local Processor interface block contains a group of Shared registers used to support general-purpose command and status interactions between the system CPU or EISA bus master and the local processor. In addition to the command/status registers, the CPU interface includes a set of ID registers for EISA expansion board ID support, and a set of Peek/Poke data registers used to hold the data during peek/poke operations.

The local interface portion of the BMIC also contains three 8-bit registers which are used by the local processor to access all of the BMICs internal registers. These registers are mapped into the local processor interface and include a local status register, local data register, and a local index register (refer to Section 3.2.6.1).

The Local Processor block contains 31 registers. A detailed description of the registers and their functions can be found under Register Description (Sections 8.1 and 8.2).

3.2.5 DATA TRANSFER TYPES

The BMIC supports four types of data transfers on the EISA bus: Burst, non-burst (two BCLK), peek/poke or locked exchange, and mismatched. For all of the above transfer types, the addressed slave device can negate EXRDY if wait state timing is required (each wait state is one BCLK).

The primary function of the BMIC is to support 16- and 32-bit burst data transfers between functions on the expansion board and the EISA memory. If the addressed memory is not capable of supporting burst transfers, the BMIC will run either 32-bit non-burst (two BCLK) cycles or, with the support of the 82358 EISA Bus Controller, run mismatched data cycles.

The burst cycle type provides a continuous sequence of one BCLK read or write cycles to and from 16- or 32-bit EISA memory. Burst cycles can not be used with I/O devices or ISA devices (slaves or masters).

The non-burst cycle type provides a continuous sequence of two BCLK read or write transfers to and from 32-bit EISA memory. The BMIC will only respond as a 32-bit master when configured for two BCLK transfers (refer to Section 4.2.1).

The peek/poke and locked exchange feature allows local processor accesses to and from individual I/O space or system memory locations on the EISA bus. The BMIC responds as a 32-bit master and generates two BCLK cycles when configured for peek/poke transfers (refer to Section 4.3). A locked exchange transfer consists of six BCLKs (peek followed by a poke). A peek/poke data transfer has the same timings as a non-burst (two BCLK) data transfer.

The mismatched cycle type provides a means of communicating with 8- or 16-bit EISA or ISA devices. In the event the I/O or memory slave device that has been addressed requires a data size translation, the BMIC will back-off the bus and allow the 82358 EISA Bus Controller to perform the necessary data size translations (refer to Section 4.2.1). The BMIC will generate mismatched cycles as required for all data transfers (burst, non-burst, peek, poke, or locked-exchange).

The following table identifies the BMIC cycle types, master sizes, slave types accessible (memory-I/O), and BCLKs per cycle.

Transfer Type	BMIC Master Size		Slave Type Accessible		BCLKs per Cycle
	16-Bit	32-Bit	I/O	Memory	
Burst	X	X		X	1
Mismatched		X		X	*
Non-Burst		X		X	2
Mismatched		X		X	*
Peek/Poke		X	X	X	2
Mismatched		X	X	X	*
Locked Exchange		X	X	X	6
Mismatched		X	X	X	*

*Depends on slave type/size (EISA/ISA, I/O/Memory, 8-bit/16-bit)

For all of the above transfer types, the addressed slave device can negate EXRDY if wait-state timing is required (each wait-state is one BCLK).

3.2.6 REGISTER ACCESSING

The BMIC provides three distinct groups of registers; the Shared register set, the Local Processor Only register set, and the Index register set. The Shared register set is used by the system CPU or EISA bus master and the local processor for general-purpose command and status interactions and expansion board ID support. The Local Processor only registers are used by the local processor to program the BMIC and provide status for data transfers across the EISA bus and Transfer Buffer interface. The Local Processor Only register set also provides address range decode support for slot specific and general I/O address ranges of interest to the expansion board. The Index register set is used by the local processor as a means of accessing all of the above registers through an indexing scheme.

The Shared register set is accessible through either the EISA interface or the Local Processor interface, the remaining two register sets are accessible through the Local Processor interface only. In the case of contention between the EISA bus and the local processor accessing a Shared register simultaneously, the local processor on the expansion board will have initial priority. Consecutive multiple accesses to the BMIC's shared registers result in a rotational arbitration between the EISA bus and the local processor.

3.2.6.1 Register Accessing through the Local Processor Interface

Register accessing on the local side of the BMIC is accomplished using an indexing procedure. The local interface portion of the BMIC contains two 8-bit registers which are used by the local processor to access all of the BMIC's internal registers. These registers are mapped into the Local Processor interface and include a local data register and a local index register. The registers are selected using the two local address lines (LADS<1:0>). The BMIC's internal register set is read by writing the address of the register to be accessed into the local index register. The register contents are then read through the Local Data register. To write to one of the BMIC's internal registers, the local processor must first write the address of the register to be accessed into the local index register, same as a read, then write the new data value to the Local Data register.

An optional auto-increment mode is supported by the BMIC, which automatically increments the index register after each register read or write. This allows for efficient programming of the register set by using byte string moves. If the Local Index register is given a local index address with bit (7) set high, the local index address will automatically increment each time the Local Data register is read or written.

The Local Status/Control register is directly mapped into the Local Processor interface and is also accessible using the two address lines (LADS<1:0>).

3.2.6.2 Register Accessing through the EISA Interface

The shared registers are mapped directly into the EISA slot-specific I/O space XC80–XC9F. The EISA address lines <11:2> and the byte enables <3:0> are used for decode during shared register accesses.

A standard slave read or write access to the BMIC consists of two BCLKs + one wait-state (one wait-state = one BCLK period). During a slave cycle where the EISA access loses the internal register access through arbitration to the local processor, the cycle will consist of two BCLKs + two wait-states. The BMIC will negate EXRDY for one BCLK for each wait-state required.

3.2.7 INTERRUPTS

The BMIC provides two interrupt request lines, one for the EISA side (EINT), and one for the local side (LINT). The EISA interrupt (EINT) can be programmed for either edge or level-triggered operations. During edge-triggered operations the EINT signal will transition from a low level to a high level. In level-triggered mode, the EISA interrupt signal is an active low open collector output. The local interrupt signal (LINT) can be programmed for either active low or active high level operations and will default to active low operation upon reset. The LINT signal is not an open collector output during active low operations and will require external logic if interrupts need to be tied together on the local side. The EINT and LINT modes of operation are programmed through the Global Configuration register.

3.2.7.1 Interrupt Sources

Several events can trigger each of the two interrupt request signals, and the events can be enabled or disabled on an individual or global basis (refer to Sections 8.1.1.3 and 8.2.2). The system CPU or EISA bus master can only be interrupted by an I/O write from the local processor to the BMIC EISA System Doorbell register. However, the local processor can be interrupted by several sources which are listed below:

— An I/O write from the system CPU or EISA bus master to the BMIC Local Doorbell register.

— The completion of a data transfer on one of the transfer channels.

3.2.7.2 Interrupt Handling

To prevent the BMIC from allowing undetected interrupts from occurring, when servicing an interrupt initiated by the BMIC, all additional interrupts must be disabled prior to reading the Local or EISA System Doorbell Status registers. The interrupts are disabled by writing to the Local or EISA System Doorbell Enable registers, depending on the source of the interrupt.

This is required due to the nature of the interrupt mechanism of the BMIC. All interrupt sources have an edge triggered nature internal to the BMIC, with each event being 'OR'ed together. Additional interrupt sources occurring after the first interrupt will set their appropriate bit in the Status register, but they will not generate an external interrupt until the initial event has been cleared. Thus if the Status register was read first, and another interrupt occurred after this read, the second interrupt would remain undetected in the status register until another event occurred. Disabling of the interrupts prior to reading the status register will prevent this from occurring.

4.0 EISA INTERFACE

4.1 EISA Interface Signals

The BMIC provides a complete interface to the EISA bus and supplies all of the control signals, data lines, and address lines necessary to implement a 16- or 32-bit EISA bus master and a 32-bit EISA slave. This includes a 32-bit data path, a 32-bit address path, and 20 EISA control signals. The BMIC also provides five control signals used to enable and disable the external data buffers and address latches, as shown in Figure 2-1.

The BMIC uses four 74F245 external bidirectional buffers to drive and receive the 32 EISA data and three 74F573 external latches to latch and drive the upper 22 EISA address lines. The external data buffers and address latches should be comprised of "F" or "AS" type logic to meet EISA speed requirements.

The upper 22 EISA addresses are multiplexed through the 22 upper EISA data lines of the BMIC. They are latched externally by the 74F573's. EISA address lines <11:2> and byte enable lines <3:0> are tied directly to the EISA bus. Address lines 10 and 11 are input directly to the BMIC for slave mode address decode. During EISA master operation, lines 10 and 11 are driven indirectly through the external latches.

As a slave, the BMIC receives address lines IADS<11:2> and byte enable lines IBE<3:0># for I/O address decode. Address lines <11:2> are used for slot specific decode and address lines <9:2> are used for general I/O address decode. Address lines <11:2> along with IBE<3:0># are used by the BMIC during Shared register accesses. Address lines <31:12> are not used by the BMIC in slave mode.

The following address lines are used during I/O decoding as shown:
Slot specific I/O address decoding (expansion board)—IADS<11:2>
Slot specific I/O address decoding (shared registers)—IADS<11:2>/IBE<3:0>#
General I/O address decoding (expansion board)—IADS<9:2>

All of the BMIC EISA control signals function as defined in the EISA bus specification. The signals are used to support the following cycles:

BMIC as a Master

	(Cycle Type Performed)			
Master Type	**Burst**	**Non-Burst**	**Mismatched**	**Peek/Poke/Locked Exchange**
32-Bit	X	X	X	X
16-Bit	X			

BMIC as a Slave

1. Responds to EISA shared register accesses as 32-bit slave.
2. Responds to slot specific and general I/O accesses (refer to Section 4.8).

4.2 Transfer Channels

The BMIC contains two identical independent transfer channels which are configurable to run either burst or non-burst (two BCLK) cycles to and from system memory. The BMIC will automatically run non-burst (two BCLK) or mismatched cycles if the memory the BMIC has addressed cannot run burst cycles. Mismatched cycles will be run if data size translation is required.

Channel 0 must be used for EISA READ operation only. Channel 1 can be used for both EISA READ and EISA WRITE operations.

Each channel has three sets of registers to regulate data transfers. These are the Base register group, the Current register group, and the Data Status/ Control register group. This implementation of a triple register set allows a processor to begin programming the next transfer on the channel while the current transfer is being executed.

The Base register set contains seven 8-bit registers. These registers are programmed by the local processor when a transfer is required across one of the channels. Four Transfer Channel Base Address registers are combined to form the starting 32-bit EISA address to be used during the transfer. The remaining three registers are the Transfer Channel Base Count Registers. The Base Count registers are combined to determine the number of transfers (in bytes) to be performed. The number of bytes which can be transferred ranges from 1 byte to 4 Mbytes. The most significant bit of the Transfer Channel Base Count register group is used to control the start of the transfer and the second most significant bit is used to control the direction of the transfer (refer to Section 8.2.3.3).

The Current register set contains seven registers each of which corresponds to a Base register. These registers are loaded from the Base registers. The Transfer Channel Current Address registers contain the 32-bit real-time EISA memory address. The Transfer Channel Current Count registers contain the number of bytes remaining to be transferred on the channel. The current register set is readable by the local processor. However, there are possible coherency problems involved with reading multiple bytes while the current registers are being updated during a transfer. To avoid these problems, a channel's transfer should be temporarily suspended (using the channel's Configuration Register) before trying to read the channel's current register set.

The Status/Control register set contains three registers: the Transfer Channel Strobe register, Transfer Channel Configuration register, and the Transfer Channel Status register. The Transfer Channel Strobe register is used to initiate the transfer of data from the Base register set to the associated Current register set. A transfer request for that channel will be generated following the Current register load. The Transfer Channel Configuration register is used to program the mode of the transfer. The Transfer Channel Status register provides current FIFO and transfer channel status.

To initialize a transfer over either of the two transfer channels, the following steps must be completed:

1. Verify that the Base registers for the desired transfer channel are available.

   The Transfer Channel Base Address and Base Count registers must be available before they can be programmed. This is determined by the status of bits 0 and 1 in the Local Status/Control register. A "1" in either of the two bits indicates that the corresponding channel is currently running a transfer and the Base registers are busy. A "0" indicates that the Base registers are free and available for programming. In the event that the Base registers are not available, the local processor must wait until the data transfer executing on the requested channel has completed, at which time bits "0" or "1" (depending on which channel was programmed) in the Local Status/Control registers will be reset to 0. Programming the Base registers during a Base register Busy state, is illegal and will corrupt the Base register data of the pending transfer. Programming the Transfer Configuration register during a cycle in progress may cause the termination of the transfer, depending on which bit in the register was changed.
2. Program the transfer channel's associated Transfer Base register set with the desired transfer information (Base registers must be available).
3. Initiate the Base register to Current register load and schedule a transfer request by writing to the channel's Transfer Strobe register.

   If a transfer is in progress on the requested channel and a write to the associated channel's Strobe register is done, the Base to Current register load will take place immediately after the data transfer on the requested channel has completed.

4.2.1 BURST AND NON-BURST MODES OF OPERATION

The BMIC can be programmed for burst or non-burst (two BCLK) data transfers to and from EISA memory. This is determined by a write to the Channel Configuration Register.

If burst mode is enabled, the BMIC will look for the SLBURST# signal at the beginning of the transfer to determine if the slave device that was addressed is

capable of running burst cycles. If the slave device does not respond with an active SLBURST# signal, the BMIC will not activate the MSBURST# signal and will proceed with either non-burst (two BCLK) bus cycles or mismatched cycles.

In burst mode, the BMIC can respond as a 16- or 32-bit master. The BMIC informs the system of this capability by driving MASTER16# low from the same BCLK rising edge that START# is asserted. MASTER16# will remain low for one BCLK. The BMIC will automatically "downshift" from a 32- to a 16-bit master if the EX32# signal is sampled inactive and the SLBURST# signal is sampled active at the beginning of a transfer. If EX32# and SLBURST# are sampled active at the beginning of the transfer, the BMIC will proceed with a 32-bit burst transfer.

In non-burst mode, the BMIC will respond as a 32-bit master. The BMIC will look for the EX32# signal at the beginning of the transfer to determine if the system memory it has addressed has the same bus width. If the EX32# signal is not returned (mismatched cycle indicated), the BMIC will "back-off" the bus by floating START#, IBE#<3:0>, and IDAT<31:0> to allow the 82358 EISA Bus Controller to take control of the transfer. The EISA Bus Controller will then proceed to assemble or disassemble the data as needed. The EISA Bus Controller will return the EX32# signal after the mismatched cycle is complete, indicating to the BMIC that a new address can be placed on the bus. If the EX32# signal is sampled active at the beginning of the transfer, the BMIC will proceed with a 32-bit non-burst (two BCLK) transfer.

4.2.2 1K PAGE ADDRESS BOUNDARY DETECTION

During burst, non-burst (two BCLK), and mismatched data cycles, the BMIC provides the support to detect 1K page address boundary crossings. If the BMIC detects that the current cycle is about to cross a 1K page boundary, the transfer will be temporarily terminated on the next cycle. The BMIC will then arbitrate between restarting the transfer on the current channel, selecting the second channel, doing a peek/poke cycle, or preempting the channel (refer to Section 4.4 for information regarding BMIC arbitration).

Example: Transfer = 32-bit transfer and page address boundary is at location 400h = 1024

1. The BMIC detects that the current cycle is about to cross a 1K page address boundary—current address (3FCh = 1020).
2. Address after BMIC has executed the current cycle (400h = 1024).
3. Transfer is temporarily terminated (interrupted).
4. BMIC will now arbitrate between restarting the transfer on a new page, selecting the second channel, doing a peek/poke cycle, or preempting the channel.

4.3 Peek/Poke, Locked Exchange Transfers

To allow the local processor to communicate with other devices in the main system, the BMIC allows the local processor to execute individual I/O or memory cycles over the EISA bus. These cycles can be thought of as being similar to "peek" and "poke" statements in the Basic programming language. These cycles may be reads, writes, or locked exchanges in 8-, 16-, 24-, or 32-bit values. All cycles must be contained within a single doubleword.

The Peek/Poke operation requires the following set of registers: Four 8-bit Peek/Poke Address registers which are combined to provide the 32-bit Peek/Poke address; One 8-bit Peek/Poke Control register which contains the bits defining whether the cycle is I/O or memory, peek (read)/poke (write) or locked exchange, and which byte enables are to be active during the cycle; and four 8-bit Peek/Poke Data registers which are used to hold the data for the Peek/Poke cycle. During all peek/poke or locked exchange cycles, byte enables IBE<3:0># are derived from bits 0–3 in the Peek/Poke Control register set. The lower two bits of the Peek/Poke Address register are ignored. Peek, poke, or locked exchange cycles will not be generated for illegal combinations of byte enables (i.e., 1111, 1010, 0110, 0101, 0100, 0010).

To do an individual write cycle (poke), the local processor must first write to the Peek/Poke Address register set to specify the 32-bit memory address or the 16-bit I/O address. It must then write the data to be transferred into the Peek/Poke Data register set. The data must be placed in the appropriate byte positions in the Data register set so that it goes out on the correct byte lanes during a 32-bit bus master transfer.

Once the appropriate data and address have been programmed, the local processor must write to the Peek/Poke Control register to specify the cycle type and initiate the cycle. After this write to the Peek/Poke Control register, bit 2 in the Local Status/Control register will be set to a 1 by the BMIC to indicate that a peek/poke request is pending and that the peek/poke registers are busy. When the poke cycle has finished executing on the EISA bus, the Peek/Poke status bit 2 in the Local Status/Control register will return to normal (0).

To do an individual read cycle (peek), the local processor must write to the Peek/Poke Address registers, then to the Peek/Poke control register to initiate the read cycle. The Peek/Poke status bit 2 in the Local Status/Control register will be set high by the BMIC and remain active until the peek cycle finishes on the EISA bus. The local processor can then read the data from the Peek/Poke data registers.

NOTE:
When running consecutive peek transfers, the data must be read from the Peek/Poke data registers before each new peek transfer is generated. The BMIC will read the data off the EISA bus from all four byte lanes regardless of which Byte enables (IBE<3:0>#) are active. (Although all bytes are read, the value of the byte enables are important to the system and must be programmed for the peek transfer).

When a locked exchange cycle is requested by the local processor, a peek cycle is scheduled first and then immediately followed by a poke cycle. The LOCK# signal is active during the locked exchange cycle to indicate to the system that no other accesses to the addressed location can be made.

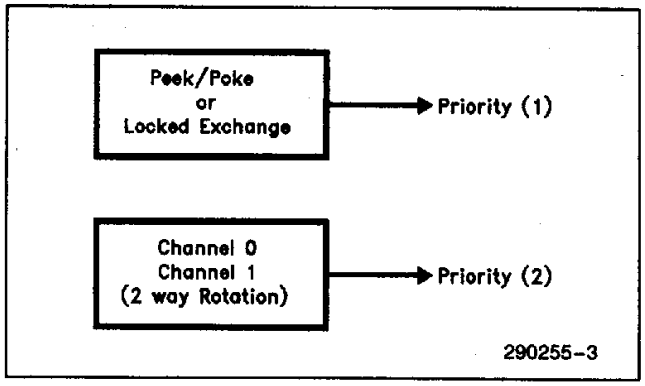
Whenever the BMIC is commanded to do an EISA POKE cycle, the BMIC will assert the MREQ# signal low normally, transfer up to four bytes of data, and release the bus by de-asserting MREQ# high. A potential problem exists, however, when the slave device extends the cycle by de-asserting EXRDY low. If the slave holds this signal low past the time that the BMIC is forced to release MREQ# high (it has been preempted while waiting for the slave to assert EXRDY high), then the BMIC will drive MREQ# back low again immediately after this cycle ends if there is another transfer pending (TBI, PEEK, POKE or LOCKED-EXCHANGE). Note that according to the EISA spec, MREQ* signal description "A bus master must wait at least two BCLKs after releasing the bus before reasserting its MREQx*". To adhere to EISA specifications, it is required that LOCKED-EXCHANGE cycles be used in lieu of POKE cycles.

Any consecutive Peek/Poke or Locked exchange transfers must be initiated only after the previous Peek/Poke or Locked exchange has been completed. This can be accomplished by making sure that bit 2 of the local status/control register is set to a zero before initiating the transfer.

4.4 Arbitration

4.4.1 EISA/BMIC ARBITRATION

The BMIC will begin master mode operation any time a transfer request is pending. If more than one transfer request is pending, the BMIC will service them in the following order. Peek/Poke cycles have the highest priority access to the EISA bus followed by the two data channels. Once the BMIC has gained control of the EISA bus, the BMIC will first perform any peek, poke, or locked exchange transfers that may be pending. If there are no peek, poke, or locked exchange transfers pending, the BMIC will run data transfers initiated by either of the two transfer channels. The two transfer channels have equal priority with respect to each other and are serviced in an alternating fashion. The priorities and assignments are as follows:



The BMIC will maintain ownership of the EISA bus until it has serviced all outstanding data transfer requests or it is preempted from the bus by the removal of the MAK# signal. The BMIC can be configured to relinquish the EISA bus immediately, 4 μs, or 8 μs after a preempt is received. If the BMIC has completed all outstanding data transfer requests prior to the time-out of the preempt timer, it will give up the bus. If the BMIC finishes one task prior to the time-out of the preempt timer, it will start on the next pending transfer request unless the request is a peek, poke, or locked exchange cycle. The BMIC will not start a set of peek, poke, or locked exchange cycles after the MAK# signal has been removed. If a transfer is cut-off due to a preempt timer time-out, the BMIC, upon regaining access to the EISA bus and following its internal arbitration priority scheme, will continue the transfer that was preempted at the point the transfer was cut-off.

When a channel is interrupted for any reason, 1K page break, FIFO stall, channel clear, channel suspend, or transfer complete, the BMIC may immediately relinquish the EISA bus depending on the state of the CFGFF bit in the Channel Configuration register set.

NOTE:
During a FIFO pause, the CFGFF bit in the associated Channel's Configuration register is ignored. The function of the CFGFF bit, as related to the above channel interruptions, is as follows:

If the CFGFF bit = 1, the BMIC will immediately relinquish control of the EISA bus upon the detection of any of the above interruptions. This will occur

regardless if there are additional data transfer requests pending. If there are additional data transfer requests pending, the BMIC will reassert MREQ# a minimum of two BCLKs later to reacquire the EISA bus. The BMIC will follow the arbitration priority scheme outlined above when servicing a data transfer request after a transfer interruption has occurred.

If the CFGFF bit = 0, the BMIC retains ownership of the EISA bus upon detection of a FIFO stall or 1K page break as long as a preempt timer timeout has not occurred. If there are additional data requests pending, the BMIC will immediately perform the pending transfer and then rearbitrate for the EISA bus to complete the interrupted transfer. If there are no additional data requests pending, the BMIC will relinquish ownership of the EISA bus only after the current transfer interruption has been serviced and completed.

4.4.2 BMIC PREEMPT TIMER

The BMIC can be preempted from the EISA bus by the 82357 (ISP). The 82357 negates MAK#, indicating to the BMIC that it must finish the current bus cycle and relinquish control of the EISA bus by negating MREQ# within 64 BCLK periods (8 μs).

The BMIC provides a programmable preempt timer which can be programmed to relinquish the bus within 3, 32, or 64 BCLKs. The preempt timer is programmable through the Global Configuration register.

The following diagrams illustrate the latest the BMIC will start a new transfer after MAK# has been negated.

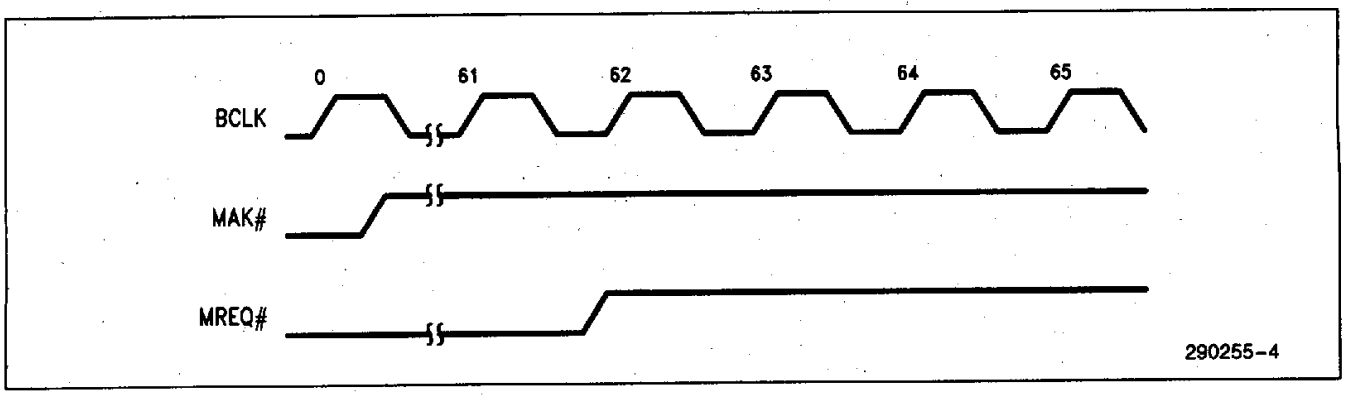
Depending on the type of transfer started, the BMIC will respond as follows:

Assumptions:

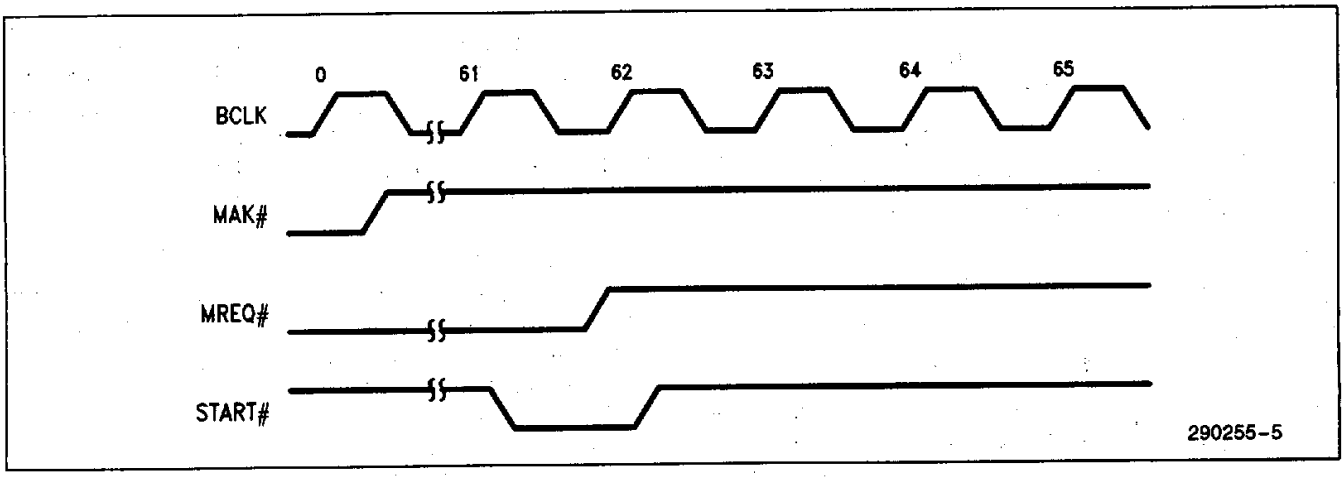
1. The 82357 has negated the MAK# signal at BCLK zero.
2. The preempt timer is programmed to relinquish the EISA bus within 64 BCLKs after the negation of MAK#.
3. Let X = programmed value of preempt delay (in BCLKs).

BMIC Response:

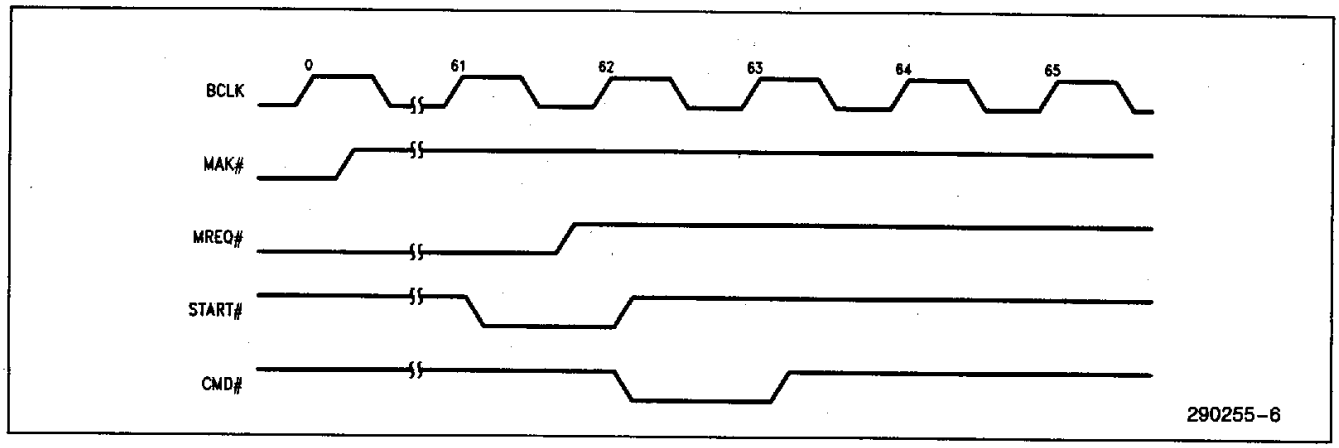
For all transfers, the BMIC will negate MREQ# within (X-2.5) BCLK periods following the MAK# transition to an inactive state (BCLK 61.5).



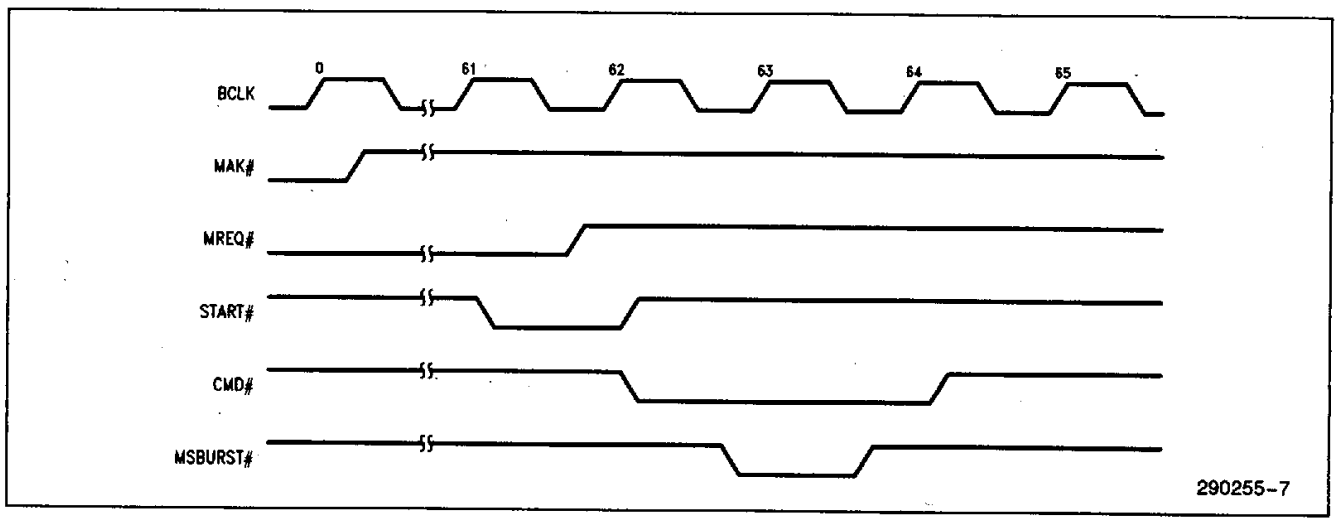
For all transfers, the BMIC may assert START# on any of the first X-3 rising edges of BCLK following the MAK# transition to an inactive state (BCLK 61).



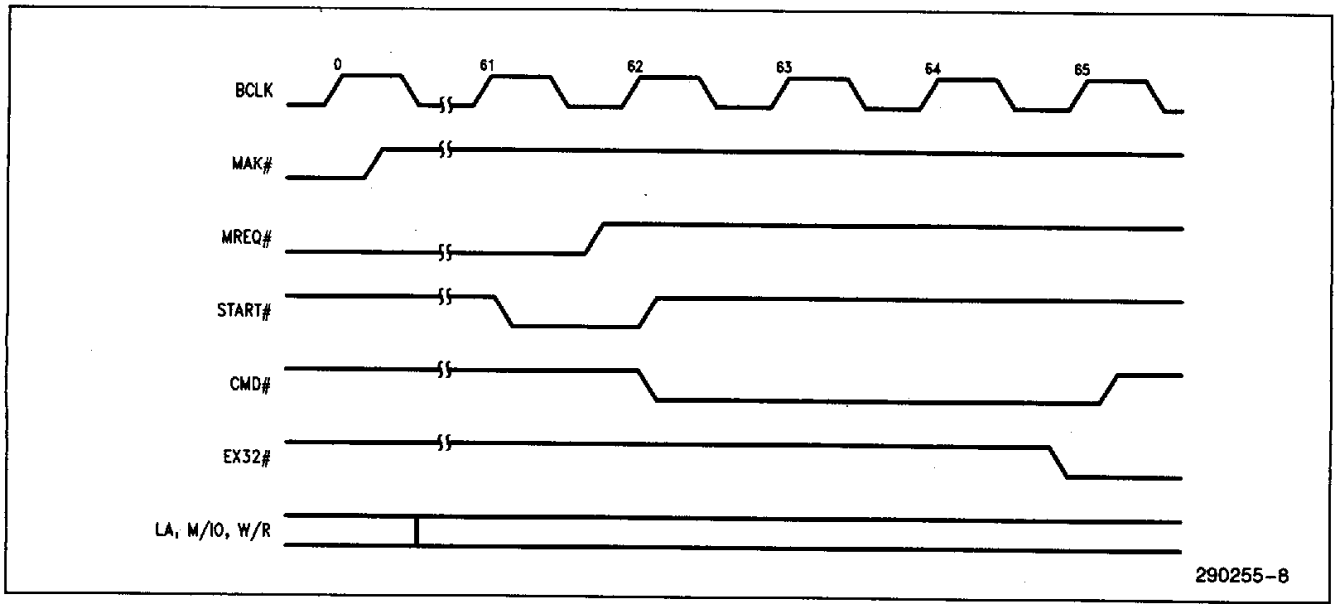
If the last cycle is a non-burst two BCLK cycle, CMD# will become inactive within (X-1) BCLK periods from the inactive transition of MAK# (BCLK 63), this is assuming that EXRDY is active.



If the last cycle is a burst EISA cycle, the BMIC will negate MSBURST# within (X-0.5) BCLK periods from the inactive transition of MAK# (BCLK 63.5). The last CMD# will go inactive within X BCLK periods from the deassertion of MAK# (BCLK 64). This is assuming EXRDY is active.



If the last cycle is mismatched, cycle completion will be controlled by the system. The BMIC will drive the LA address, M/IO, and W/R signals until the falling edge of BCLK after the last CMD# inactive transition.



4.5 EISA Address Incrementer

The Transfer Channel Current Address register set for each channel functions as an address incrementer and is used to generate and track the address of the data during transfers. The register set increments the address according to the number of bytes being transferred during that cycle. The transfer is automatically aligned on doubleword boundaries. The two least significant bits of the starting 32-bit address (A0 and A1) are used to determine the initial address increment value.

For 32-bit transfers, the BMIC provides an initial address increment of 1, 2, 3 or 4 depending on the value of address lines A<1:0>. After the initial increment, the BMIC increments the address by 4 until the last cycle is detected.

The following example illustrates the BMIC address incrementer during a 32-bit master mode transfer.

		EISA Address			
		A3	A2	A1	A0
Start Address	FFFFF001h	0	0	0	1
Initial Increment	FFFFF004h	0	1	0	0
(Incremented by 3)					
All Increments Following	FFFFF008h	1	0	0	0
(Incremented by 4)	FFFFF00Ch	1	1	0	0

The starting address A<1:0> is 01, this means that the initial increment must be 3 in order to align the next increments on doubleword boundaries. The subsequent increments will be by 4 until the last cycle is detected.

For 16-bit transfers, the BMIC provides an initial address increment of 1 or 2 depending on the status of address lines A<1:0>. After the initial increment, the BMIC increments the address by two until the last cycle is detected.

The following example illustrates the BMIC address incrementer during a 16-bit master mode transfer.

		EISA Address			
		A3	A2	A1	A0
Start Address	FFFFF001h	0	0	0	1
Initial Increment	FFFFF002h	0	0	1	0
(Incremented by 1)					
All Increments Following	FFFFF004h	0	1	0	0
(Incremented by 2)	FFFFF006h	0	1	1	0

The starting address A<1:0> is 01, this means that the initial increment must be 1 in order to align the next increments on singleword boundaries. The subsequent increments will be by 2 until the last cycle is detected.

NOTE:
The BMIC internally assembles 32-bit dwords. When a 16-bit burst transfer is preempted, the transfer will stop on a doubleword boundary.

4.6 EISA Byte Decrementer

The Transfer Channel Current Count register set for each channel contains the intermediate value of the byte count during the transfer and is used as the byte decrementer. The decrementer's function is partially based upon the address incrementer. In the above 32-bit incrementer example, the byte count would be decremented by 3 on the first cycle. After the initial decrement, the channel's Current Count register set is decremented by 4 until the last cycle is detected. In the above 16-bit incrementer example, the byte count would be decremented by 1 on the first cycle. After the initial decrement, the channel's Current Count register set is decremented by 2 until the last cycle is detected. Note that the Current Count register does not decrement entirely to zero. Instead, it retains the value of the number of bytes transferred during the last cycle.

4.7 EISA Address Incrementer/Byte Decrementer Illustration

The following table illustrates the various states of (A0, A1) vs the transfer byte-count and the initial address during a 32-bit transfer.

Byte Count	Starting Address	Next Address	Initial Increment	Number of Bytes Left	Last Cycle	Number of Cycles Left
1	XXX 0000	NA	NA	0	Yes	0
	XXX 0001	NA	NA	0	Yes	0
	XXX 0010	NA	NA	0	Yes	0
	XXX 0011	NA	NA	0	Yes	0
2	XXX 0000	NA	NA	0	Yes	0
	XXX 0001	NA	NA	0	Yes	0
	XXX 0010	NA	NA	0	Yes	0
	XXX 0011	XXX 0100	1	1	No	1
3	XXX 0000	NA	NA	0	Yes	0
	XXX 0001	NA	NA	0	Yes	0
	XXX 0010	XXX 0100	2	1	No	1
	XXX 0011	XXX 0100	1	2	No	1
4	XXX 0000	NA	NA	0	Yes	0
	XXX 0001	XXX 0100	3	1	No	1
	XXX 0010	XXX 0100	2	2	No	1
	XXX 0011	XXX 0100	1	3	No	1
5	XXX 0000	XXX 0100	4	1	No	1
	XXX 0001	XXX 0100	3	2	No	1
	XXX 0010	XXX 0100	2	3	No	1
	XXX 0011	XXX 0100	1	4	No	1
6	XXX 0000	XXX 0100	4	2	No	1
	XXX 0001	XXX 0100	3	3	No	1
	XXX 0010	XXX 0100	2	4	No	1
	XXX 0011	XXX 0100	1	5	No	2
7	XXX 0000	XXX 0100	4	3	No	1
	XXX 0001	XXX 0100	3	4	No	1
	XXX 0010	XXX 0100	2	5	No	2
	XXX 0011	XXX 0100	1	6	No	2
8	XXX 0000	XXX 0100	4	4	No	1
	XXX 0001	XXX 0100	3	5	No	2
	XXX 0010	XXX 0100	2	6	No	2
	XXX 0011	XXX 0100	1	7	No	2
9	XXX 0000	XXX 0100	4	5	No	2
	XXX 0001	XXX 0100	3	6	No	2
	XXX 0010	XXX 0100	2	7	No	2
	XXX 0011	XXX 0100	1	8	No	2
10	XXX 0000	XXX 0100	4	6	No	2
	XXX 0001	XXX 0100	3	7	No	2
	XXX 0010	XXX 0100	2	8	No	2
	XXX 0011	XXX 0100	1	9	No	3

NOTES:
1. "X" = Don't Care
2. If the "byte count" is less than or equal to the "initial increment", then the current cycle = the first cycle = the last cycle.
3. If the number of bytes left is less than or equal to 4, then the next cycle = the last cycle.
4. For information regarding byte alignment, refer to Section 6.3.1.

4.8 I/O Address Range Decode Support

The BMIC provides on-board decoder logic, two I/O select pins (IOSEL<1:0>#), and a set of 8-bit I/O Decode Range registers to support both general I/O decode and expansion board slot specific I/O decode. The BMIC also uses the AEN signal when decoding I/O locations.

The set of I/O Decode registers include two I/O Decode Range Base Address registers and two I/O Decode Range Control registers (refer to Section 8.2.6). The I/O Decode registers are used to define the address ranges of interest to the bus master expansion board. Each IOSEL#<1:0> pin has an associated Control and Base register along with an associated address range as defined by the I/O Decode register set.

Through the I/O Decode Range Control register set, the BMIC can be programmed to respond to a select I/O address range as either an 8-bit or 32-bit EISA device. The only control signal provided by the BMIC to the EISA bus during an I/O decode is the EX32# signal. The output state of the EX32# pin on the BMIC will indicate the elected response (low = 32-bit EISA, high = 8-bit EISA). The Control register set controls the size of the I/O decode range, the I/O decode type (slot specific or general I/O), and the I/O decode address latching. The I/O address can be latched by the CMD# signal (de-pipelined) or merely decoded. By latching the I/O address, the associated IOSEL# line will remain active a minimum of 5 ns from the rising edge of CMD#.

The IDOEs do not go active during an IOSEL cycle outside the shared register access space.

The I/O decode range size depends on the value of bits <4:0> in the Control register. Each of these bits masks a corresponding address comparison bit in the Base register. If no bits are masked in the Control register, the BMIC will decode a doubleword address. The bits are masked as follows:

I/O Control Register	I/O Base Register Bit Masked	EISA Address Bit Masked
Bit 0	Bit 0	IADS2
Bit 1	Bit 1	IADS3
Bit 2	Bit 2	IADS4
Bit 3	Bit 3	IADS5
Bit 4	Bit 4, 5	IADS<7:6>

The I/O Decode Range Base Address register contains the address range that is used during the I/O decode address comparison. The following table gives the bits in the I/O Base Address Register and the EISA Address that are used during the comparison:

I/O Base Address Register	(EISA Address Bits)	
	Slot Specific	General I/O
Bit 0	IADS2	IADS2
Bit 1	IADS3	IADS3
Bit 2	IADS4	IADS4
Bit 3	IADS5	IADS5
Bit 4	IADS6	IADS6
Bit 5	IADS7	IADS7
Bit 6	IADS10	IADS8
Bit 7	IADS11	IADS9

If bit 6 in the I/O Decode Range Control register is programmed for General I/O decode, and the two most significant bits in the I/O Decode Range Base Address register are programmed to 0 (IADS<9:8>), I/O decoding for that range will be disabled. This is done to ensure that the I/O address does not conflict with the slot specific address range or the EISA system board address range. The following table summarizes the EISA system I/O address mapping:

I/O Address Range (HEX)	I/O Range Reserved for
0000–00FF	EISA/ISA System Board
0100–03FF	General I/O (ISA Expansion Board)
0400–04FF	ISP (82357)
0500–07FF	General I/O (Alias of 0100h–03FFh)
0800–08FF	EISA System Board
0900–0BFF	General I/O (Alias of 0100h–03FFh)
0C00–0CFF	EISA System Board
0D00–0FFF	General I/O (Alias of 0100h–03FFh)
Slot Specific Range where X = Slot Number	
X000–X0FF	Slot (X)
X100–X3FF	General I/O (Alias of 0100h–03FFh)
X400–X4FF	Slot (X)
X500–X7FF	General I/O (Alias of 0100h–03FFh)
X800–X8FF	Slot (X)
X900–XBFF	General I/O (Alias of 0100h–03FFh)
XC00–XCFF	Slot (X) (BMIC Registers 0C80h–0CAFh)
XD00–XFFF	General I/O (Alias of 0100h–03FFh)

The following is an example of the BMIC programmed for slot specific decode:

I/O Decode Range 0 Control register programmed for (EFh)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
(EFh)	1	1	1	0	1	1	1	1

— Bit 7—Respond as a 32-bit EISA slave
— Bit 6—Slot specific decode enabled
— Bit 5—Slot specific address latched by CMD#
— Bit 4—Compare I/O Decode Range 0 Base Address Bits (5) and (4) with EISA address signals IADS7 and IADS6 respectively
— Bit 3—Mask I/O Decode Range 0 Base Address Bit (3)
— Bit 2—Mask I/O Decode Range 0 Base Address Bit (2)
— Bit 1—Mask I/O Decode Range 0 Base Address Bit (1)
— Bit 0—Mask I/O Decode Range 0 Base Address Bit (0)

I/O Decode Range 0 Base Address register programmed for (2-h)

	IADS11	IADS10	IADS7	IADS6	IADS5	IADS4	IADS3	IADS2
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
(2-h)	0	0	1	0	—	—	—	—

EISA slot specific address range decoded—X080h through X0BFh where X represents the expansion board slot number

```
      (X080h)                                (X0BFh)
xxxx000010(00) (00)00throughxxxx000010(11) (11)11
           ↑   ↑                                ↑   ↑
                    EISA Address
                     Bits Masked
                    (IADS<5:2>)
```

Byte enables IBE<3:0># and EISA address lines IADS<1:0> are not used during either slot specific or general I/O decode. During slot specific I/O decode, EISA address lines IADS<9:8> must be 0 to ensure that the I/O address does not conflict with the ISA general I/O address range (0100h–03FFh).

IOSEL0# and EX32# will be driven low by the BMIC if addresses X080h through X0BFh are present on the EISA bus.

AEN is used as part of the decode and must be negated low when a response from the BMIC is required.

5.0 TRANSFER BUFFER INTERFACE

5.1 Transfer Buffer Interface Signals

The Transfer Buffer Interface portion of the BMIC provides the signals essential for interfacing to the expansion board as required for EISA-to-expansion board and expansion board-to-EISA burst data transfers. The Transfer Buffer Interface is designed to interface to a high speed transfer buffer and simple logic similar to that used in traditional DMA designs. This interface includes a 16-bit data bus, one clock input, and seven control signals.

The 16-bit data lines TDAT<15:0> are used by the BMIC to transfer the data to and from the transfer buffer logic on the expansion board during transfers. The data is word aligned. The BMIC automatically assembles the words received from the expansion board into 32-bit dwords for 32-bit transfers over the EISA bus. The data lines are also used by the BMIC to transport internally generated transfer start and real-time addresses to the external logic for use during data transfers (refer to Section 5.3).

The clock input (TCLK) controls the transfer rate between the BMIC and the external transfer buffer logic. The TCLK can be asynchronous to the BCLK.

The seven control signals include:

— Transfer Request (TRQ#): an output to request data transfers over the Transfer Buffer interface.
— Transfer Acknowledge (TACK#): An input to acknowledge data transfers. The TACK# signal may be used by the transfer buffer logic to add wait states to the data cycle.
— Data Transfer Direction (TDIR): An output to inform the external transfer buffer logic as to the direction of the current transfer (EISA read or EISA write).
— Transfer Channel Select (TCHAN): An output to indicate which of the two channels is currently active.
— Transfer Address Counter Load (TLD#): An output to load the current transfer start address to an external address counter, depending on the expansion board application.
— Transfer Data Output Enable (TDOE#): An input that unconditionally disables the BMIC from driving the TDAT<15:0> lines. With this signal, the BMIC can be prevented from driving the TDAT<15:0> lines while the local processor accesses the transfer buffer logic on the expansion board. No handshaking is required, so throughput is increased.
— Transfer End-of-Process (TEOP#): TEOP# is a status output pin that signals the end of a data transfer to the external transfer buffer logic.

NOTE:
Refer to Section 9.4 for additional information regarding the above signals.

5.2 External Transfer Buffer Logic

The Transfer Buffer interface is designed for high speed devices, such as SRAM based designs, or FIFOs. The Transfer Buffer interface data path is 16 bits wide. This requires the transfer clock (TCLK) to run at a speed of 16 MHz to 20 MHz to maintain the EISA maximum data rate of 33 Mbytes/sec. The fast cycle times required on the data Transfer Buffer interface can be implemented in the controlled environment found locally on the expansion board. If two BCLK transfers are used on the EISA side (16 Mbytes/sec), the timing requirements for the transfer buffer can be relaxed, and lower cost implementations can be utilized.

If the transfer buffer controller does dynamic arbitration for the transfer buffer between the BMIC and the peripheral device(s) on the expansion board, the peripheral device accesses should be short enough so that the BMIC's data FIFO can handle the interruption to its data flow without stalling the EISA transfer.

Examples of transfer buffer architecture implementations that could be interfaced to the BMIC include:

— A FIFO implementation which is large enough to buffer the difference in throughput rates between the peripheral device on the expansion board and the EISA Bus. See Section 5.2.1.

— A small high-speed DMA like device that generates addressing for a SRAM based transfer buffer.

— A controller implementation for dual-ported SRAM for high transfer buffer bandwidth.

— A page or nibble-mode dynamic-RAM controller implementation for large, low cost transfer buffers.

— For graphics systems, the frame buffer itself can be used for the transfer buffer with a non-linear address generator for transferring windows in the screen image.

5.2.1 FIFO IMPLEMENTATION

During EISA writes, the BMIC will overread the transfer buffer (read data beyond the number of bytes to be transferred) by a maximum of 28 bytes. These overread bytes may contain valid data (back to back transfers) which will be lost. The data loss can be avoided through software or hardware. The software solution avoids back to back transfers. This implies that there is data for only one transfer in the FIFO at any given time.

The hardware solution requires an external 22-bit Byte Counter and a Flip-Flop. The terminal count of the Byte Counter is used to SET the Flip-Flop which disables BMIC reads to the FIFO. The BMIC will continue to read (overread) "stale" data. The BMIC TEOP# output signal is used to RESET the Flip-Flop enabling BMIC reads to the external FIFO.

5.3 Transfer Interface Start Address Generation

The BMIC provides four 8-bit Transfer Buffer Interface (TBI) registers, two Base and two Current registers, which can be programmed with 16-bit transfer start addresses. Each transfer channel has an associated Base and Current register pair. The Base registers contain the start address and the Current registers provide the real-time address used to track the current transfer. The Current registers will increment by one each time a 16-bit word is transferred across the Transfer Buffer interface.

The 16-bit start address is transferred across the TDAT<15:0> lines to the transfer buffer logic at the beginning of all new data transfers (i.e., each time the TBI Base register set contents are transferred to the TBI Current register set). The contents of the TBI Base registers are transferred to the TBI Current registers after a write to the associated channel's Transfer Strobe register is completed (refer to Section 4.2). The BMIC provides a load signal (TLD#) which can be used to latch the start address into an external address counter for use by the transfer buffer logic.

The BMIC can also be programmed to generate the transfer address each time the associated channel regains the bus, in which case, the address will be the real-time address. By programming the CFGEA bit in the Channel Configuration register to a "1", the start address will be transferred to the transfer buffer logic at the beginning of all new transfers and the real-time address will be transferred each time the associated channel regains the bus. If the CFGEA bit is set to a "0", the transfer start address will be transferred at the beginning of all new transfers and the real-time address will not be transferred.

NOTE:
The TBI Current register set is readable by the local processor. However, there are possible coherency problems involved with reading multiple bytes while the current registers are being updated during a transfer. To avoid these problems, the channel's transfer should be temporarily suspended (using the channel's Configuration Register) before trying to read the channel's TBI Current register set.

5.4 Transfer Buffer Interface Timing Example

Figures 5-1 and 5-2 illustrate the start up and conclusion of a transfer cycle across the Transfer Buffer interface and should be used as a reference when reading the following text.

1. At the start of a data transfer TCHAN and TDIR change to their new values prior to the falling edge of TLD# to set up the cycle. TCHAN and TDIR will not change states as long as TRQ# is asserted.
2. TLD# is asserted until acknowledged by TACK#. The transfer address is transferred to the external logic each time the TBI Base register contents are transferred to the TBI Current register set (new transfer) and, if programmed, each time the current channel regains the bus.
3. The new address is loaded using the TDAT bus during TLD# at point (A). The TDAT bus should be turned on by asserting TDOE# during TLD# if the internal start address is required. Once the external channel address and direction are set up, the data transfer can begin.
4. Data transfer requests are signaled by TRQ# being asserted (low). TRQ# will remain active until the data transfer is completed or a transfer interruption occurs (refer to Section 1.0) followed by TACK# active. During an EISA write, there will be a one TCLK delay between TLD# deasserting and TRQ# asserting as denoted by point (D) in Figure 5-2. This is to allow time for the external buffers to change direction after the TLD# has been completed.
5. Each word transfer to or from the BMIC is acknowledged by the TACK# signal. If TACK# is active at the rising edge of TCLK, one word will be transferred. If TACK# is not active at the rising edge of TCLK, the word that is currently being transferred will be inhibited and a wait state will

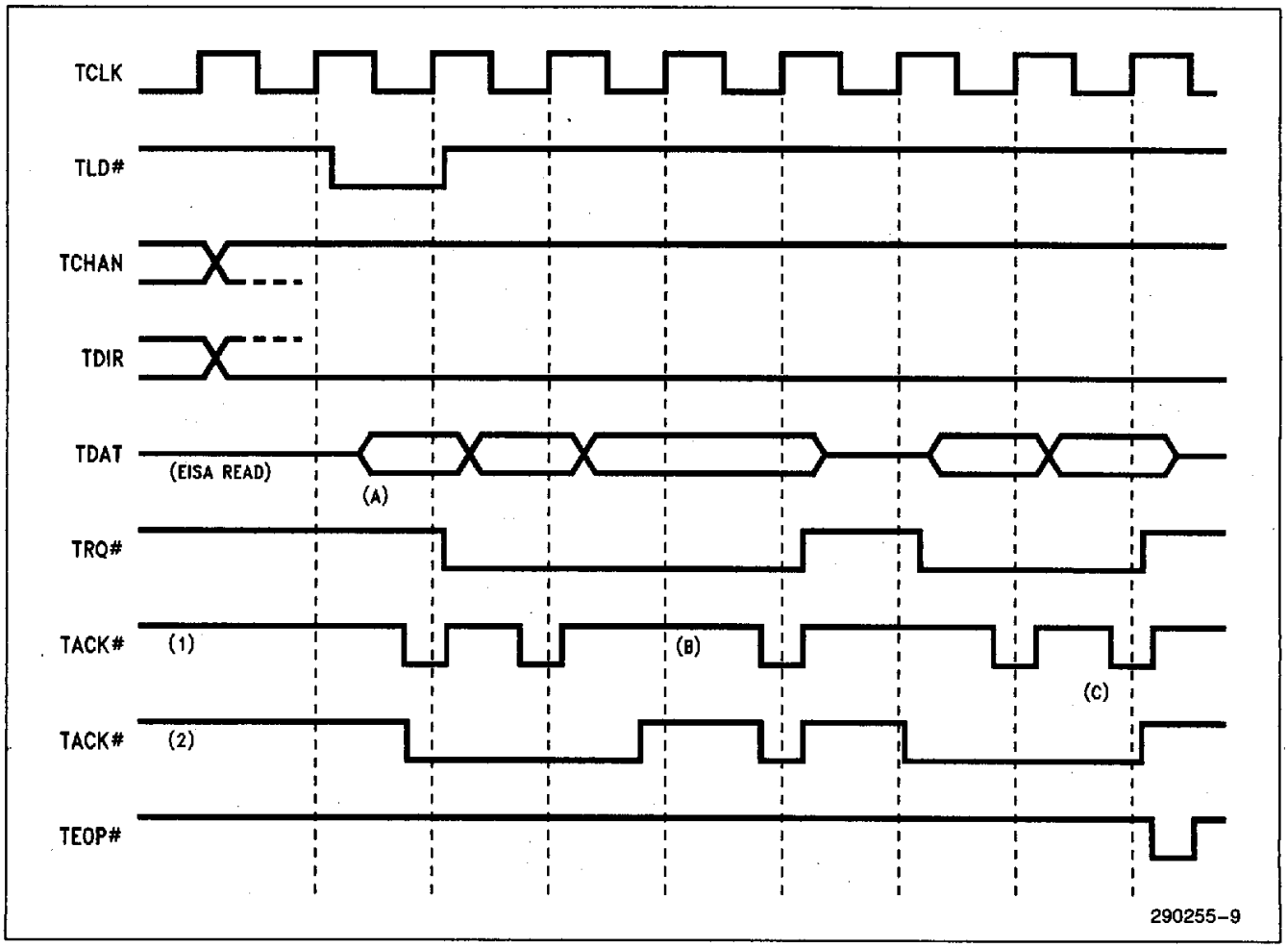


Figure 5-1. Transfer Buffer Interface Timing (EISA READ)

be inserted. This is shown at point (B) in Figure 5-1. Such a wait may be needed when the external transfer buffer logic is arbitrating between the BMIC and the I/O subsystem on the expansion board. Wait states may also be inserted by stretching TCLK at point (C) in both of the figures. Clock stretching is possible as long as the one to one ratio of TCLK to BCLK is not violated.

NOTE:
A long TCLK stretch time will hang the Transfer Buffer interface. Also, TCLK must be running during the time TRQ# is inactive in order for the Transfer Buffer interface to function properly.

As indicated above, TACK# must be stable at the rising edge of TCLK. However, TACK# can assume any convenient pattern at other times. As shown by the first pattern, TACK# (1) pulses low at the TCLK edge that data is transferred. This pattern is particularly useful when TCLK wait-states are desired as indicated at point (B) in Figure 5-1. The alternate pattern (TACK#2) is useful during TCLK stretching since TACK# is always low during TRQ# as shown at point (C). This is effective since the transfer clock edge timing is controlled by the amount TCLK is stretched.

6. TEOP# is asserted at the end of a transfer by the BMIC.

   The BMIC will indicate end-of-process by asserting TEOP# shortly after the negation of the last CMD# in the transfer. During an EISA write transfer, the BMIC will assert TEOP# a maximum of two TCLKs after CMD# is negated. During an EISA read transfer, the BMIC will assert TEOP# typically eight TCLKs after the negation of CMD#. In either case (EISA read or EISA write), the TEOP# signal is delayed from the rising edge of TCLK.

NOTE:
The BMIC will assert the expansion board interrupt signal (LINT) at the end of a transfer, if so programmed in the Transfer Channel Configuration register.

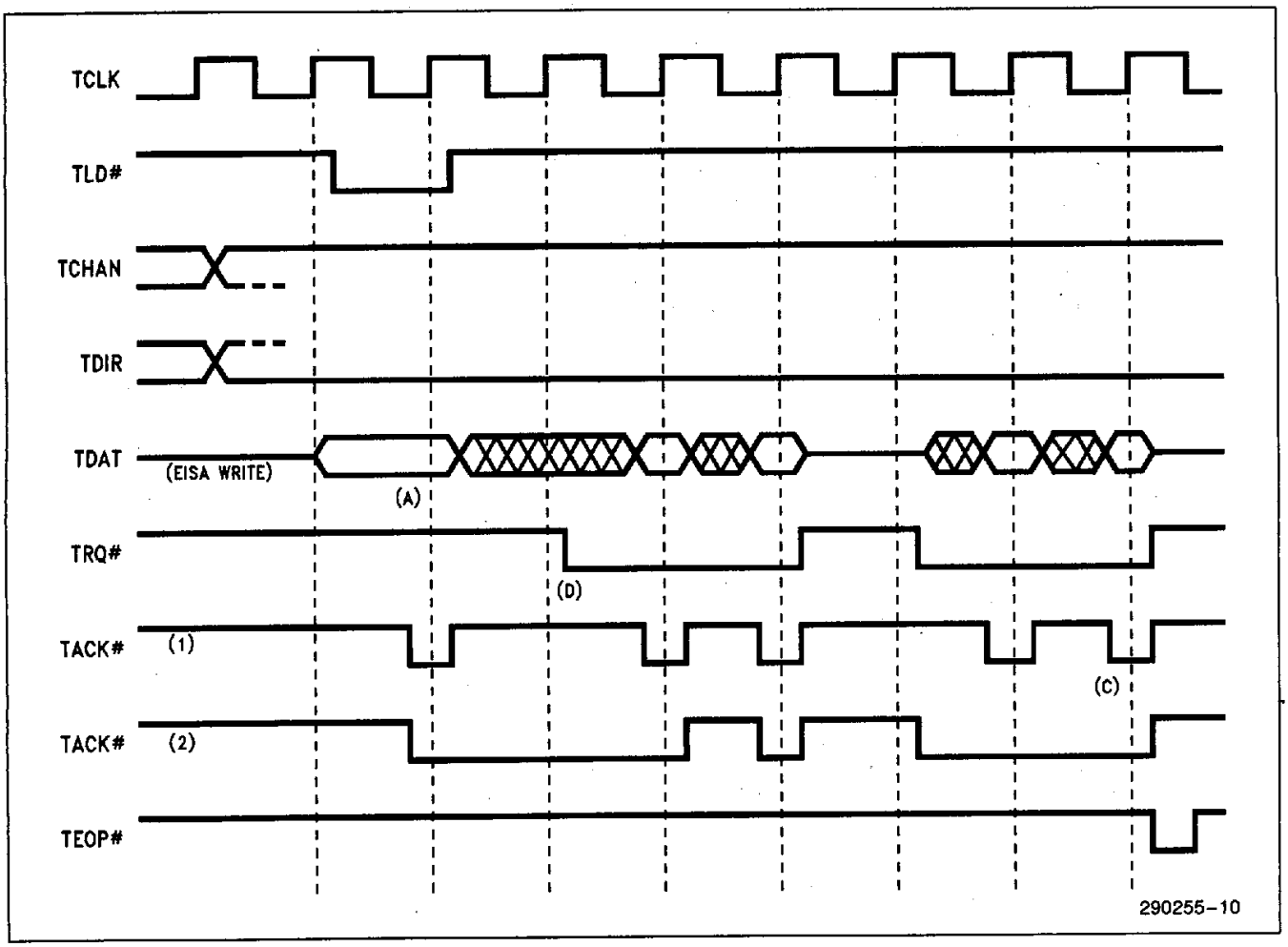


Figure 5-2. Transfer Buffer Interface Timing (EISA WRITE)

6.0 FIFO/DATA ALIGNER

6.1 FIFO/Data Aligner

The BMIC uses two identical FIFOs, one per transfer channel, and a common data aligner for data transfers between system memory and the bus master expansion board. The primary function of the FIFO/Data Aligner Unit is to help isolate and simplify the timing relationships between the EISA bus and the devices on the expansion board.

The FIFO allows the timing on the expansion Board side of the BMIC to be based on a locally generated clock signal. This transfer clock (TCLK) can be independent of the EISA BCLK signal that governs EISA bus timing. The FIFO also provides latency protection for wait states generated on either the EISA bus or expansion board.

The Data Aligner arranges the 16-bit data from the external transfer buffer to any arbitrary byte alignment in system memory. The data aligner also performs the assembly and disassembly of the EISA data during the transfer. The TDAT data assembly and disassembly is done by the Transfer Buffer interface portion of the BMIC.

6.2 FIFOs

Each FIFO on-board the BMIC is 24 bytes in size. The transfer data is written into the FIFOs from either the expansion board or the EISA bus side, depending on the direction of transfer. The data is written into the FIFO as doublewords during the transfer. However, if the data is not doubleword aligned, partial FIFO loads will be done at the beginning or end of a transfer depending on the byte count, address programmed and the direction of the transfer.

The condition of the FIFOs can be determined by a read to the Transfer Channel Status register set. A read to this register will indicate whether the FIFOs are stalled or active. A FIFO stall is defined as a FIFO that is full during an EISA read or empty during an EISA write. In either case, the transfer buffer logic is unable to keep up with the EISA device. If a FIFO stall occurs, the transfer will be stopped and the BMIC will either service the transfer request with the highest priority or relinquish the EISA bus to the system. The BMIC will relinquish the bus to the system if the CFGFF bit in the channel's corresponding Configuration register is set to a 1.

6.3 Data Aligner

6.3.1 EISA BYTE ALIGNMENT

The BMIC automatically handles the byte alignment for the EISA bus in the case of misaligned doubleword boundaries and assumes no performance penalty. The BMIC will do any partial doubleword transfers as required at the beginning and the end of all transfers. The two least significant bits of the 32-bit transfer start address (A1 and A0) are used to provide the byte alignment for both EISA read and EISA write transfers. The following tables illustate the BMIC's byte alignment approach during 32- and 16-bit transfers:

In the following tables "—" represents no data transferred and the digits represent the data items being transferred. The byte alignment for an EISA read is identical to that of an EISA write.

EISA Write (32-bit/12-byte Transfer) and (16-bit/6-byte Transfer)

		(32-Bit)				(16-Bit)			
A1	**A0**	**Output Data to EISA Bus**				**Output Data to EISA Bus**			
Byte Lane	**→**	**3**	**2**	**1**	**0**	**3**	**2**	**1**	**0**
0	0	03	02	01	00	—	—	01	00
		07	06	05	04	—	—	03	02
		11	10	09	08	—	—	05	04
0	1	02	01	00	—	—	—	00	—
		06	05	04	03	—	—	02	01
		10	09	08	07	—	—	04	03
		—	—	—	11	—	—	—	05
1	0	01	00	—	—	—	—	01	00
		05	04	03	02	—	—	03	02
		09	08	07	06	—	—	05	04
		—	—	11	10	—	—	—	—
1	1	00	—	—	—	—	—	00	—
		04	03	02	01	—	—	02	01
		08	07	06	05	—	—	04	03
		—	11	10	09	—	—	—	05

6.3.2 DATA ASSEMBLY/DISASSEMBLY

Before being placed on either the TDAT or IDAT data buses during an EISA read or EISA write, the data will be assembled or disassembled as required. The IDAT data is assembled and disassembled by the FIFO/data aligner portion of the BMIC and the TDAT data is assembled and disassembled by the Transfer Buffer interface portion of the BMIC. The following paragraphs illustrate the BMIC's assembly and disassembly approach during 32- and 16-bit transfers. The illustration assumes that byte alignment is not required.

During 32-bit EISA read transfers, the 32-bit doublewords are removed from the EISA bus and placed into the FIFO. After flowing through the FIFO, the 32-bit doublewords are copied-down to 16-bit words and then placed on the TDAT bus.

During 32-bit EISA write transfers, the 16-bit words are removed from the TDAT lines, assembled into 32-bit doublewords, and then placed into the FIFO. After flowing through the FIFO, the 32-bit data is placed on the EISA bus. No further assembly or disassembly is required after the FIFO as the data is already in 32-bit doubleword form.

During 16-bit EISA read burst transfers, the 16-bit words are removed from the EISA bus, assembled into 32-bit doublewords, and then placed into the FIFO. After flowing through the FIFO, the 32-bit data is copied-down to 16-bit words and then placed on the TDAT bus.

During 16-bit EISA write burst transfers, the 16-bit words are removed from the TDAT bus, assembled into 32-bit doublewords, and then placed into the FIFO. After flowing through the FIFO, the 32-bit data is copied-down to 16-bit words and then placed on the EISA bus.

7.0 LOCAL PROCESSOR INTERFACE

The BMIC's Local Processor interface is based on an asynchronous, 8-bit interface. All of the slave signals required for a local processor to program the BMIC are provided through this interface. These signals include (LCS#, LRD#, LWR#); two address lines (LADS0 and LADS1) for addressing internal registers; an 8-bit data path (LDAT); an interrupt signal (LINT); and a ready signal (LRDY). LINT allows the BMIC to interrupt the local processor and the ready signal (LRDY) indicates when valid data is available on the LDAT lines (shared register accesses only, see below). If a local processor is not used, the Local Processor interface can be connected to the 8-bit ISA bus (refer to Section 7.3). The choice of the local microprocessor or microcontroller used depends upon the specific application and the degree of performance and data processing needed (refer to Section 7.2).

The Local Processor interface portion of the BMIC contains two 8-bit registers which are used by the local processor to access all of the BMIC's internal registers. These registers are mapped into the Local Processor interface and include a Local Data register and a Local Index register. These registers are selected using the Local Processor interface's two address lines. The Local Status/Control register is also directly mapped into the Local Processor interface and is used to provide the local processor with the interrupt, peek/poke, and Base register status.

The BMIC allows the local processor and the EISA bus to communicate with each other through a set of Command/Status registers. The Command/Status registers are referred to as shared registers and include a set of Mailbox registers, Semaphore ports, and doorbell registers. The mailbox registers are used to pass messages to and from the local processor and the EISA bus and are controlled by the Semaphore ports. The Doorbell register set is used to inform the respective processor of new messages. Also part of the shared register set are the ID registers, which are used to support the EISA expansion board ID function.

The BMIC allows the local processor access to individual locations in system memory or I/O space using the Peek/Poke feature. The local processor can also initiate BMIC burst and non-burst (two BCLK) data transfers to and from system memory.

7.1 Shared Registers—Status/ Command Support

As data transfer rates increase, it is critical that an efficient command and status passing mechanism be implemented so that command and status exchange does not become a new bottleneck to system performance. The BMIC utilizes a high-performance command/status interface between the main system and the local processor to minimize command/status overhead.

The Shared registers are a group of registers accessable by the system CPU or EISA bus master and the local processor for general-purpose command and status interactions and EISA expansion board ID function support. The features of the BMIC command/status support include a pair of semaphore ports, a set of interrupt ports ("doorbell registers"), and a set of mailbox registers. With these functions, many different types of high-performance communication protocols can be defined between the system and the expansion board. The Global

Configuration register, the System Interrupt Enable/Control register, and the ID registers are also part of the shared register set.

7.1.1 SEMAPHORE PORTS

The two semaphore ports are specifically designed to allow set-and-test functions in I/O space. Specifically, the ports are used to lock access to the mailbox registers and to lock access to links in main memory. Each of the semaphore ports consists of two parts: the semaphore flag bit and the semaphore test bit.

When a write occurs to the semaphore flag bit through either the EISA interface or the Local Processor interface, the old value of the semaphore flag bit is copied to the appropriate semaphore test bit. The old value of the semaphore flag bit is then available in the test bit to be read back by the processor. If the value read back from the semaphore test bit is a "1", the requested resource is unavailable for use. If the value read back is a "0", the requested resource is available for use and is now locked by the requesting processor or bus master. In this manner, set-and-test algorithms can be implemented without using the EISA bus lock function. The processor or EISA bus master unlocks the semaphore by simply writing a "0" to the associated semaphore flag bit.

NOTE:
The Semaphore ports and resources are locking only in a software sense, as in any semaphore in main memory. The Semaphore ports are identical and are not associated with either interface (EISA or Local). The protocol for the semaphores and the effect they have on other shared registers, like the Mailbox registers, is strictly a matter of how the system software chooses to use them.

Implementing the semaphore in the BMIC instead of main memory eliminates the need for the BMIC to arbitrate for the EISA bus every time it wishes to update or test the semaphore. Note that the semaphore scheme described here is functional only when a single device on the EISA is communicating with the BMIC; the semaphore coordinates "locks" between the single device and the local processor. In the case that multiple masters attempt to lock access to the BMIC, the masters must first agree amongst themselves which one has the privilege to use the BMIC semaphore port(s).

7.1.2 MAILBOX REGISTERS

A set of 16 8-bit general-purpose mailbox registers are used to pass information between the bus master expansion board and the EISA system. The 16 registers are mapped contiguously in EISA slot-specific I/O space, so they can be accessed as bytes, words, or doublewords. These registers can be used to directly pass command and status information, or they can be used as pointers to larger command blocks in memory.

The mailbox registers can be read or written at any time from either the EISA bus or the Local Processor interface. An internal arbitration is implemented in such a way that if there is a simultaneous read and write from both sides of a mailbox register, then the read operation will not contain indeterminate bits. In other words, when a read operation is done on a mailbox register at the same time as a write operation to that register, the bit pattern that is read will be either the old bit pattern in the mailbox, or the new bit pattern being written, but never some transitory, invalid bit pattern.

7.1.3 DOORBELL REGISTERS

There are two 8-bit doorbell Interrupt/Status registers in the BMIC, one assigned to the EISA side and one assigned to the expansion board side. The EISA System Doorbell register is used by the local processor to request service from the EISA side and the Local Doorbell register is used by the device on the EISA side to send an interrupt request to the local processor on the bus master expansion board. The doorbell Interrupt/Status registers are implemented with "sticky" bits, so that individual bits in the register can be set by the interrupting device or reset by the servicing device without knowledge of the states of the other bits in the register. The eight bits in each doorbell register allow up to eight separate devices or events in each direction to have interrupt requests pending simultaneously. The interrupt requests pending in either of the two Doorbell registers are ORed with the other interrupt sources from within the BMIC, and the result is sent out over one of the two interrupt pins: LINT or EINT.

Each doorbell register has an associated 8-bit Interrupt Enable register used to enable or disable the interrupts on an individual basis. The BMIC also includes a System Interrupt Enable/Control register and a Local Status/Control register used to disable the system (EINT) and local (LINT) interrupts and to verify the status of the system and local interrupts on a global basis (refer to Sections 8.1.1.3.3 and 8.2.2).

The following paragraphs describe the operation of the Local Doorbell Interrupt/Status register. The EISA System Doorbell Interrupt/Status register is similar, but operates in the opposite direction.

Each device or event that can interrupt the bus master expansion board can be assigned a bit position within the BMIC's Local Interrupt/Status Doorbell

register. When the device on the EISA bus wants to send an interrupt request to the bus master expansion board, it writes to the Local Interrupt/Status Doorbell register (from the EISA side) with that device's assigned bit position set active. This will set that bit in the Local Interrupt/Status Doorbell register, but leave the other bits in the register unaffected. If that bit position is not disabled, then the interrupt signal to the local processor will be asserted.

When the local processor services the interrupt, it checks the Local Status/Control Register to determine the source of the interrupt. If the control register indicates that the Local Doorbell register is one of the active interrupt sources, then the local processor can read the Local Doorbell register to determine which bits are active and requesting interrupts. If the local processor decides to service one of the requests from the Local Doorbell register, it can write to the Local Doorbell register with that bit's position set. This action will cause that bit in the Local Doorbell register to reset, but the other bits will remain unaffected. Thus, each bit in the Local Doorbell register is like a set-reset flip-flop, with the EISA bus controlling the "set" input, and the Local Processor interface controlling the "reset" input.

7.2 Local Processor Recommendations

The Local Processor interface to the BMIC will support numerous processors, from the 8088 microprocessor to the 376 embedded processor.

The 80186, 80C186, 80188, and 80C188 family of processors provides a clean interface to the BMIC's Local Processor interface and eliminates the need for additional logic. An on-board programmable wait-state generator eliminates the need for external wait-state generation logic between the processor and the BMIC during non-shared register accesses.

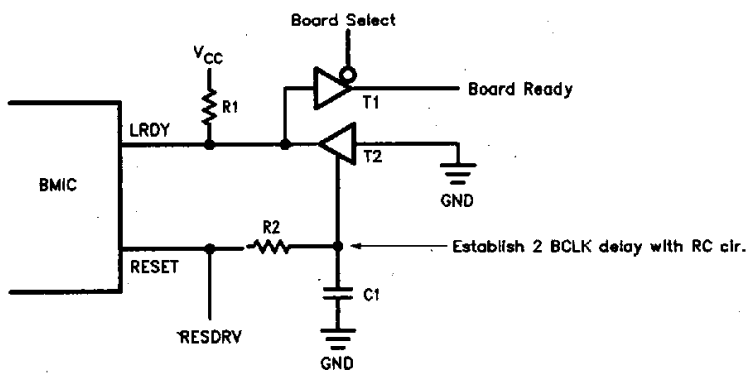
7.3 Requirements for No Local Processor

The BMIC allows for expansion board designs that do not require a local processor. To support the programming of the BMIC in a no local processor board design, the Local Processor interface must be connected to the ISA bus. However, when the ISA bus is used, the BMIC must be informed that there is no local processor and that it must change its function slightly (refer to next section). To inform the BMIC that no local processor is present, LRDY must be driven low during RESET and remain low a minimum of two BCLKs after RESET is negated.

The following circuit can be used to establish the proper LRDY/RESET timing as required for a no local processor design (see Figure 7-1).

7.4 EISA ID Function Support/ Registers

The BMIC provides support for the EISA expansion board ID function. The primary ID implementation takes advantage of the local processor. Upon reset, the local processor executes a routine from its ROM that writes the product identifier for the expansion board to the four 8-bit ID registers in the BMIC. The registers are accessed through the Local Processor interface and are located at local index addresses 00h–03h. On the EISA side, these registers are mapped into the EISA slot specific ID address range XC80h–XC83h.



R1 = 8.2K
R2 = Depends on frequency of BCLK and application implemented.
C1 = Depends on frequency of BCLK and application implemented.
T1 = Active low tri-state buffer
T2 = Active high tri-state buffer

Figure 7-1. LRDY/RESET Circuit with No Local Processor

If the host CPU accesses the ID registers in the BMIC before the local processor has programmed them, the BMIC will return the setup delay ID code 0111XXXXh in the byte 0 ID register located at EISA slot specific I/O address XC80h. The byte 0 ID register should be programmed last by the local processor.

If a local processor is not used, external registers will have to be implemented on the expansion board to hold the expansion board ID value. The BMIC will automatically set its I/O Decode Range 0 Control register to decode 8-bit EISA ID addresses. The IOSEL0# output signal can then be used to trigger external logic on the expansion board to enable ID data onto the IDAT<7:0> data lines. The ID register must be connected as shown in Figure 7-2. The external logic should monitor SA1 and SA0 on the ISA bus to determine which data byte to drive.

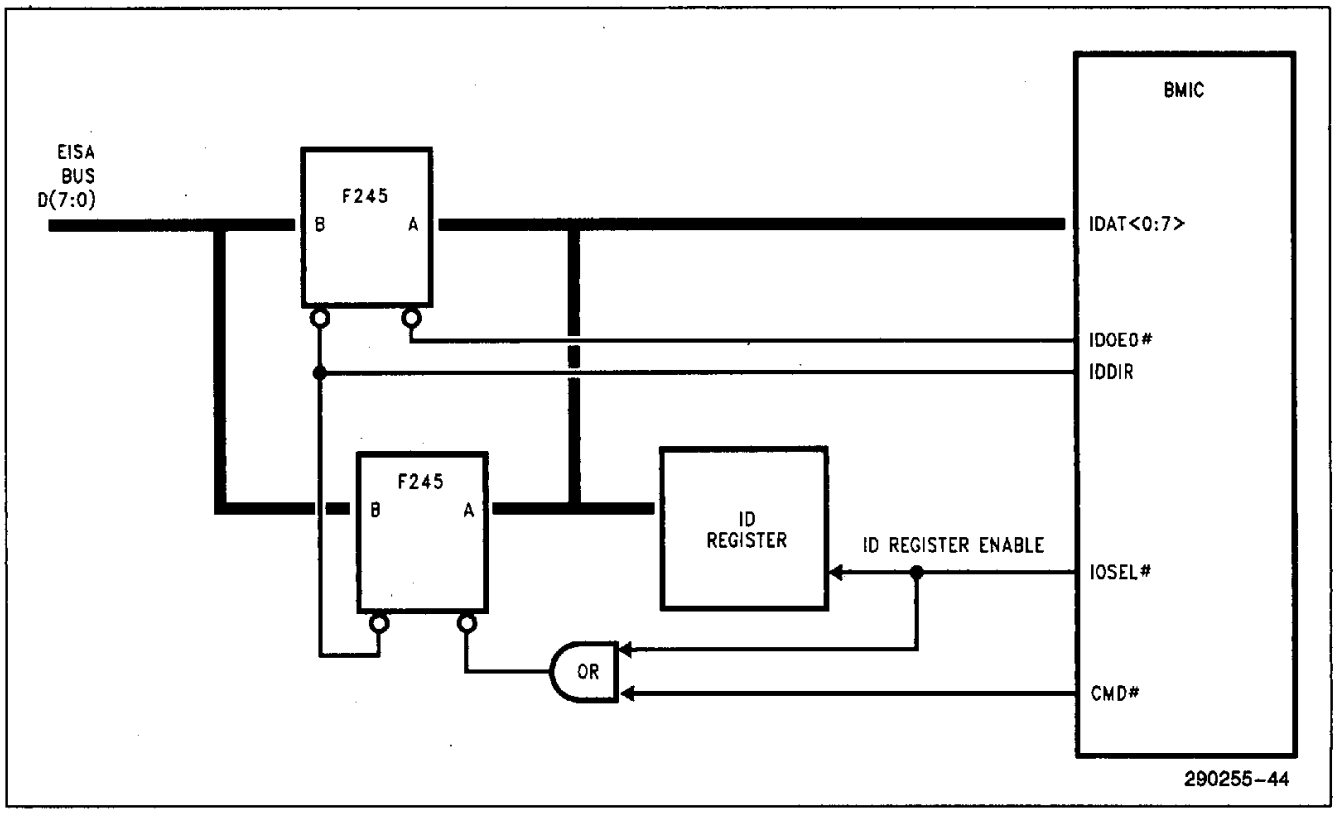


Figure 7-2. IDOE0# Connection during ID Register Access

8.0 REGISTER DESCRIPTION

8.1 Shared Register Description

The following is a table of the Shared register group listing the number of registers, register type (read/write) as related to the local and EISA side, register name, and register size:

Number	EISA	Local Type	Register Name Type	Active Bits per Register
2	R/W	R/W	Semaphore Register	2 Bits
16	R/W	R/W	Mailbox Register	8 Bits
1	R/W	R/W	Local Doorbell Interrupt/Status Register	8 Bits
1	R	R/W	Local Doorbell Enable Register	8 Bits
1	R/W	R/W	EISA System Doorbell Interrupt/Status Register	8 Bits
1	R/W	R	EISA System Doorbell Enable Register	8 Bits
1	R/W	R	System Interrupt Enable/Control Register	8 Bits
1	R	R/W	Global Configuration Register	8 Bits
4	R	R/W	ID Register	8 Bits

8.1.1 COMMAND/STATUS SUPPORT REGISTERS

8.1.1.1 Semaphore Ports (Read/Write)

The BMIC contains two Semaphore ports which can be used to software lock resources between the EISA bus and the local processor. Each semaphore port controls a 1-bit semaphore flag. Upon reset, the Semaphore ports are reset to 0.

Semaphore Port 0 EISA Address—XC8Ah
Semaphore Port 0 Local Index Address—0Ah

Semaphore Port 1 EISA Address—XC8Bh
Semaphore Port 1 Local Index Address—0Bh

						Bit 1	Bit 0
—	—	—	—	—	—		

Bit 7–2 —Reserved, set to 0
Bit 1 —Semaphore Test bit (Read Only)
Bit 0 —Semaphore Flag bit (Read/Write)

Bit (0) reflects the actual value of the semaphore at any given instant. Whenever a write is done to the Semaphore Flag bit (0), its previous value is simultaneously copied to the Semaphore test bit (1). Internal to the BMIC, there are two test bits for each semaphore port: one for the EISA interface and one for the Local Processor interface. To do a test-and-set function, write to the semaphore port with the desired semaphore value in the flag bit. After a write has been completed, read the semaphore port and check the test bit to verify that a collision did not occur.

8.1.1.2 Mailbox Registers (Read/Write)

The mailbox registers are sixteen 8-bit, general purpose registers. The format of the contents of the mailbox registers is user-defined. The Mailbox register set is not initialized to a fixed value upon reset.

EISA Address—XC90h through XC9Fh
Local Index Address—10h through 1Fh

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

8.1.1.3 Doorbell Registers

8.1.1.3.1 Local Doorbell Interrupt/Status Register (Read/Write)

This register is implemented with "sticky" bits (refer to Section 7.1.3). The Local Doorbell Interrupt/Status register is used by the EISA bus to send an interrupt request to the expansion board. When read from, this register indicates the status of pending interrupt events. Upon reset, the Doorbell Interrupt/Status register is reset to 0.

EISA Address—XC8Dh
Local Index Address—0Dh

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

Bit 7–0 1 = Doorbell interrupt pending (local CPU read)
Set Doorbell bit (EISA write)
Reset Doorbell bit (Local CPU write)
0 = No doorbell interrupt pending (Local CPU read)
No action (EISA or local CPU write)

Bits 0–7 allow up to eight events or devices on the EISA side to interrupt the local side of the BMIC. The above bits can only be reset by the servicing processor on the local side.

8.1.1.3.2 Local Doorbell Enable Register (Read/Write)

The Local Doorbell Enable register is used by the local processor to enable or disable interrupt requests to the local expansion board. This register is read only from the EISA side. Upon reset, the Doorbell Enable register is set to 0.

EISA Address—XC8Ch
Local Index Address—0Ch

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

Bit 7–0 1 = Enable doorbell interrupt for corresponding bit position
0 = Disable doorbell interrupt for corresponding bit position
No action (local CPU write)

Bits 0 through 7 act as interrupt enables for bits 0 through 7 in the Local Doorbell Interrupt/Status register respectively.

8.1.1.3.3 EISA System Doorbell Interrupt/Status Register (Read/Write)

This register is implemented with "sticky" bits (refer to Section 7.1.3). The EISA System Doorbell Interrupt/Status register is used by the expansion board to send an interrupt request to the EISA bus. When read from, this register indicates the status of pending interrupt events. Upon reset, the EISA System Doorbell Interrupt/Status register is reset to 0.

EISA Address—XC8Fh
Local Index Address—0Fh

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

Bit 7–0 1 = Doorbell interrupt pending (EISA read)
Set Doorbell bit (Local CPU write)
Reset Doorbell bit (EISA write)
0 = No doorbell interrupt pending

Bits 7–0 allow up to eight events or devices on the expansion board to send interrupts to the EISA bus. The above bits can only be reset by the servicing processor on the EISA side.

8.1.1.3.4 EISA System Doorbell Enable Register (Read/Write)

The EISA System Doorbell Enable register is used by the EISA processor to enable or disable interrupt requests to the EISA side. This register is read only from the local side. Upon reset, the EISA System Doorbell Enable register is reset to 0.

EISA Address—XC8Eh
Local Index Address—0Eh

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

Bit 7–0 1 = Enable doorbell interrupt for corresponding bit position
0 = Disable doorbell interrupt for corresponding bit position

Bits 0 through 7 act as interrupt enables for bits 0 through 7 in the EISA System Doorbell Interrupt/Status register respectively.

8.1.1.3.5 System Interrupt Enable/Control Register (Read/Write)

This register is used by the processor on the EISA side to disable the EINT signal. The EISA processor also can read this register to determine whether there are any pending interrupt requests in the EISA System Doorbell Interrupt/Status register. This register is read only from the local side. Upon reset, this register is reset to 0.

EISA Address—XC89h
Local Index Address—09h

—	—	—	—	—	—	Bit 1	Bit 0

Bit 7–2 — Reserved, set to 0

Bit 1 — (read-only bit)
1 = Enabled interrupts are pending in EISA System Doorbell Interrupt/Status register
0 = No enabled interrupts are pending in EISA System Doorbell Interrupt/Status register

Bit 0 —
1 = Enable interrupts from System Doorbell register (EISA write)
0 = Disable interrupts from System Doorbell register (EISA write)

8.1.2 GLOBAL CONFIGURATION REGISTER (READ/WRITE)

This register is used to program the type of protocol, edge or level-triggered, that will be used with the EINT and LINT interrupt signals. The Global Configuration register is also used to program the preempt timer and provide four bits for a BMIC hardware revision number. This register is read only from the EISA side. Upon reset, bits 0–3 are reset to 0.

EISA Address—XC88h
Local Index Address—08h

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

Bits 7–4 (read-only)
Hardware revision number of the BMIC

Bit 3
1 = System interrupt pin (EINT) uses edge-triggered protocol (Active high)
0 = System interrupt pin (EINT) uses level-triggered protocol (Active low open collector)

Bit 2
1 = Local interrupt pin (LINT) is set for active high operation
0 = Local interrupt pin (LINT) is set for active low operation

Bits 1, 0 Delay to give up bus after preempt
00 = 3 BCLKs
01 = 32 BCLKs
10 = 64 BCLKs
11 = reserved

EISA Address—XC80h through XC83h (bytes 0–3)
Local Index Address—0h through 3h (bytes 0–3)

8.1.3 ID REGISTERS

The ID register set consists of four 8-bit registers. These registers are programmed at initialization time with the product identifier for the expansion board which contains the BMIC. The registers are mapped as read-only into the EISA ID I/O address range. Upon reset, the ID byte 0 register will contain the value 0111XXXX, which is the EISA ID delay value. The local processor should program byte 0 last. If the external ID support scheme is selected, then these registers are disabled. The bit definitions defined below have significance for the EISA ID protocol but not for any BMIC hardware functionality. Upon reset, ID bytes 1–3 are not initialized to a fixed value.

ID Register Bytes 0–3:

	7	6	5	4	3	2	1	0
Byte 0	—	MCC14	MCC13	MCC12	MCC11	MCC10	MCC24	MCC23
Byte 1	MCC22	MCC21	MCC20	MCC34	MCC33	MCC32	MCC31	MCC30
Byte 2	MCC43	MCC42	MCC41	MCC40	MCC53	MCC52	MCC51	MCC50
Byte 3	MCC63	MCC62	MCC61	MCC60	MCC73	MCC72	MCC71	MCC70

ID Register Byte 0:

Bit 7	—	Reserved
Bits 6–2	MCC1<4:0>	First character of manufacturer's code
Bits 1, 0	MCC2<4:3>	First portion of second character of manufacturer's code

ID Register Byte 1:

Bits 7–5	MCC2<2:0>	Second portion of second character of manufacturer's code
Bits 4–0	MCC3<4:0>	Third character of manufacturer's code

ID Register Byte 2:

Bits 7–4	MCC4<3:0>	First hex digit of product number
Bits 3–0	MCC5<3:0>	Second hex digit of product number

ID Register Byte 3:

Bits 7–4	MCC6<3:0>	Third hex digit of product number
Bits 4–0	MCC7<3:0>	Hexadecimal digit of product revision

8.2 Local Processor Only Registers

The following is a table of the Local Processor Only register group listing the number of registers, register type (read/write) as related to the local side, register name, and register size:

Number	Local Type	Register Name	Active Bits per Register
INDEX REGISTERS			
1	R/W	Local Index Register	8 Bits
1	R/W	Local Data Register	8 Bits
1	R/W	Local Status/Control Register	8 Bits
DATA CHANNEL TRANSFER REGISTERS			
4	R/W	Data Transfer Channel 0 Base Address Register	8 Bits
4	R/W	Data Transfer Channel 1 Base Address Register	8 Bits
4	R	Data Transfer Channel 0 Current Address Register	8 Bits
4	R	Data Transfer Channel 1 Current Address Register	8 Bits
3	R/W	Data Transfer Channel 0 Base Count Register	8 Bits
3	R/W	Data Transfer Channel 1 Base Count Register	8 Bits
3	R	Data Transfer Channel 0 Current Count Register	8 Bits
3	R	Data Transfer Channel 1 Current Count Register	8 Bits
DATA TRANSFER CONTROL/STATUS REGISTERS			
1	W	Channel 0 Transfer Strobe Register	0
1	W	Channel 1 Transfer Strobe Register	0
1	R/W	Channel 0 Configuration Register	8 Bits
1	R/W	Channel 1 Configuration Register	8 Bits
1	R/W	Channel 0 Status Register	6 Bits
1	R/W	Channel 1 Status Register	6 Bits
PEEK/POKE REGISTER			
4	R/W	Peek/Poke Address Register	8 Bits
4	R/W	Peek/Poke Data Register	8 Bits
1	R/W	Peek/Poke Control Register	8 Bits
I/O DECODE REGISTERS			
1	R/W	I/O Decode Range 0 Base Address Register	8 Bits
1	R/W	I/O Decode Range 1 Base Address Register	8 Bits
1	R/W	I/O Decode Range 0 Control Register	8 Bits
1	R/W	I/O Decode Range 1 Control Register	8 Bits
TRANSFER BUFFER INTERFACE (TBI) REGISTERS			
2	R/W	TBI Channel 0 Base Address Register	8 Bits
2	R/W	TBI Channel 1 Base Address Register	8 Bits
2	R	TBI Channel 0 Current Address Register	8 Bits
2	R	TBI Channel 1 Current Address Register	8 Bits

8.2.1 INDEX REGISTERS

The BMIC's register set is accessed using the local Index and Local Data register set (refer to Section 3.2.6.1). The Local Index and Local Data registers are mapped directly into the Local Processor interface of the BMIC.

8.2.1.1 Local Index Register (Read/Write)

The Local Index register contains the address of the BMIC register that is currently being accessed. An optional auto-increment mode is supported through this register, which automatically increments the index register after each Local Data register read or write. Upon reset, the Local Index register is set to 0.

Local Address—1h

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

Bit 7 — 1 = Autoincrement local index register after access to local data register
0 = Do not autoincrement

Bits 6–0 — Local index address

8.2.1.2 Local Data Register (Read/Write)

During a BMIC local register access, the value of the register being accessed is passed through this register.

Local Address—0h

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

8.2.2 LOCAL STATUS/CONTROL REGISTER (READ/WRITE)

The Local Status/Control register is directly mapped into the Local Processor interface and is accessible using the two address lines (LADS<1:0>). This register provides current local doorbell interrupt status, current Channel 0 and Channel 1 interrupt and Base register status, and current peek/poke cycle status. This register is also used by the local processor on the expansion board to disable and provide the current status of the LINT signal (active or inactive). Bit 4 in this register is read/write and the remaining bits are read only. Upon reset, the Local Status/Control register is reset to 0.

Local Address—2h

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

Bit 7 — R 1 = Enabled interrupts are pending in Local Doorbell register
0 = No enabled interrupts are pending in Local Doorbell register

Bit 6 — R 1 = Enabled interrupts are pending from channel 1 events
0 = No enabled interrupts are pending from channel 1 events

Bit 5 — R 1 = Enabled interrupts are pending from channel 0 events
0 = No enabled interrupts are pending from channel 0 events

Bit 4 — R/W 1 = Local interrupts enabled
0 = All local interrupts disabled

Bit 3 — R 1 = Local interrupt signal (LINT) is currently active
0 = LINT signal is currently inactive

Bit 2 — R 1 = Most recent peek/poke command is still pending
0 = Most recent peek/poke command is complete

Bit 1 — R 1 = Base register set for channel 1 is busy
0 = Base register set channel 1 is available

Bit 0 — R 1 = Base register set for channel 0 is busy
0 = Base register set for channel 0 is available

8.2.3 DATA CHANNEL TRANSFER REGISTERS

The Data Channel Transfer register set is used to control burst and standard EISA data transfers. Each transfer channel has a set of Base and Current registers, and also a Transfer Strobe, Configuration, and Status register.

NOTE:
The Base register set and the Transfer Strobe register must be initialized before a transfer can take place. They are not initialized to a fixed value upon reset.

8.2.3.1 Channel 0 and 1 Transfer Base Address Registers (Read/Write)

Each Channel has an associated Base Address register set. The Transfer Base Address registers are programmed with the 32-bit starting address to be used during the data transfer. After the Base registers have been programmed, they should not be programmed again until the contents of the Base registers have been transferred to the Current registers. The Base Address registers are not initialized to a fixed value upon reset.

Channel 0 Local Index Address—43h through 46h (bytes 0 through 3)
Channel 1 Local Index Address—63h through 66h (bytes 0 through 3)

BYTE 3	BYTE 2	BYTE 1	BYTE 0

8.2.3.2 Channel 0 and 1 Transfer Current Address Registers (Read Only)

Each Channel has an associated Current Address register set. The Transfer Current Address registers contain the real-time status of the 32-bit transfer address. The Current Address registers are not initialized to a fixed value upon reset.

NOTE:
The current register set is readable by the local processor. However, there are possible coherency problems involved with reading multiple bytes while the current registers are being updated during a transfer. To avoid these problems, a channel's transfer should be temporarily suspended (using the channel's Configuration Register) before trying to read the channel's current register set.

Channel 0 Local Index Address—53h through 56h (bytes 0 through 3)
Channel 1 Local Index Address—73h through 76h (bytes 0 through 3)

BYTE 3	BYTE 2	BYTE 1	BYTE 0

8.2.3.3 Channel 0 and 1 Transfer Base Count Registers (Read/Write)

Each Channel has an associated Base Count register set. The Transfer Base Count registers are programmed with the number of bytes to be transferred. Each Channel has 22 bits of counter space for a maximum transfer block size of 4 Mbytes. Bits 22 and 23 are used for channel control. The Base Count registers are not initialized to a fixed value upon reset.

Channel 0 Local Address—40h through 42h (bytes 0 through 2)
Channel 1 Local Address—60h through 62h (bytes 0 through 2)

BYTE 2			BYTE 1	BYTE 0
Bit 23	Bit 22	Bit 16–21	Bits 8–15	Bits 0–7

Bit 23 — R/W
1 = Start transfer as soon as base register set is copied to current register set
0 = Hold transfer after current register set is loaded. Wait for transfer suspend bit 0 to be reset

Bit 22 — W
1 = Transfer from bus master expansion board to EISA bus (EISA write)
0 = Transfer from EISA bus to bus master expansion board (EISA read). This is applicable only to channel 1 and not for channel 0, as channel 0 can perform EISA WRITE transfers only.

Bits 0–21 — R/W
Transfer byte count

If bit 23 in the Base Count register is not set to a 1, the channel suspend bit (CFGSU) in that channel's corresponding configuration register is automatically set to a 1. The bit will be set during the Base register to Current register transfer. This ensures that a channel request for that channel is not generated. When the local processor resets the channel suspend bit to 0 in the corresponding Configuration register, a transfer request will be generated.

NOTE:
If the initial byte count is programmed to be "0", no transfer request will be generated and no transfer will occur.

8.2.3.4 Channel 0 and 1 Transfer Current Count Registers (Read Only)

Each Channel has an associated Current Count register set. The Transfer Current Count registers contain the 22-bit value representing the number of bytes remaining to be transferred on the channel. This value can be from one byte to four Mbytes. Bit 23 is reserved. Bit 22 is used to indicate the direction of the transfer. Upon reset, the Current Count registers are not initialized. At the end of a transfer, this register contains the value of the number of bytes transferred during the last cycle.

Channel 0 Local Index Address—50h through 52h (bytes 0 through 2)
Channel 1 Local Index Address—70h through 72h (bytes 0 through 2)

BYTE 2			BYTE 1	BYTE 0
Bit 23	Bit 22	Bit 16–21	Bits 8–15	Bits 0–7

Bit 23 — Reserved

Bit 22 — 1 = Current transfer is from bus master expansion board to EISA bus
0 = Current transfer is from EISA bus to bus master expansion board

Bits 0–21 — Current transfer byte count

8.2.4 DATA TRANSFER STATUS/CONTROL REGISTERS

8.2.4.1 Channel 0 and 1 Transfer Strobe Registers (Write Only)

Each channel has an associated Transfer Channel Strobe register. The Strobe register is used to initiate the transfer of information from the Base register set to the Current register set. The act of writing to this register will initiate the Base to Current transfer. There are no bits to this register, the data written to this register is ignored and the register cannot be read.

If bit 23 in the Transfer Base Count Register is set to a 1, the data transfer will be requested immediately. Otherwise, the transfer will wait until the transfer suspend bit CFGSU for the corresponding channel is reset. The transfer suspend bit is located in the Configuration register.

Channel 0 Local Index Address—49h
Channel 1 Local Index Address—69h

8.2.4.2 Channel 0 and 1 Transfer Channel Configuration Registers (Read/Write)

Each channel has an associated Transfer Configuration register. Upon reset, the Configuration registers are reset to 0. The Configuration register set is used to configure the channels as follows:

Channel 0 Local Index Address—48h
Channel 1 Local Index Address—68h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CFGEA	CFGIE	CFGIT	CFGFF	CFGBR	CFGCL	CFGEI	CFGSU

Bit 7 — CFGEA 1 = Enable real-time address transfer to transfer buffer logic
0 = Disable real-time address transfer to transfer buffer logic

Bit 6 — CFGIE Reserved. This bit must always be written with 0.

Bit 5 — CFGIT 1 = Enable interrupt on transfer complete
0 = Disable interrupt on transfer complete

Bit 4 — CFGFF 1 = Give up ownership of EISA bus if a transfer interruption occurs on this channel
0 = Retain ownership of EISA bus if a transfer interruption occurs on this channel

Bit 3 — CFGBR 1 = Enable EISA burst transfer
0 = Disable burst transfers (channel uses non-burst (2 BCLK) cycle transfers)

Bit 2 — CFGCL 1 = Clear channel
Stop any transfers and flush the data FIFO
0 = No operation
Always returns a 0 when read

Bit 1 — CFGEI Reserved. This bit must always be written with 0.

Bit 0 — CFGSU 1 = Temporarily suspend transfer
0 = Allow transfer to proceed

The CFGEA Bit enables the real-time address transfer to the transfer buffer logic. If the CFGEA bit is set to a 1, the transfer buffer real-time address for the active channel is transferred to the transfer buffer logic each time that channel regains the bus and the start address is transferred each time the Base register contents are loaded into the corresponding Current registers. If the CFGEA bit is set to 0, the address load signal (TLD#) is activated only when the Base is loaded into the Current register (refer to Section 5.3).

The CFGIE Bit is a reserved bit. Zero (0) must always be written at this bit location. This bit can be ignored during register reads.

The CFGIT Bit enables an interrupt on transfer complete (EOP).

The CFGFF Bit controls whether EISA bus ownership is relinquished or maintained after a transfer interruption. When a channel is interrupted for any reason, (1K page break, FIFO stall, channel clear, transfer suspend, or transfer complete), the BMIC may relinquish the EISA bus depending on the state of the CFGFF bit in the above register. The function of the CFGFF bit, as related to the above channel interruptions, is as follows:

If the CFGFF bit = 1, the BMIC will relinquish control of the EISA bus upon the detection of any of the above interruptions. This will occur regardless if there are additional data transfer requests pending. If there are additional data transfer requests pending, the BMIC will reassert MREQ# a minimum of two BCLK's later to reacquire the EISA bus.

If the CFGFF bit = 0, the BMIC retains ownership of the EISA bus upon detection of a FIFO stall or 1K page break as long as a preempt timer timeout has not occurred. If there are additional data requests pending, the BMIC will immediately perform the pending transfer and then rearbitrate for the EISA bus to complete the interrupted transfer. If there are no additional data requests pending, the BMIC will relinquish ownership of the EISA bus only after the current transfer interruption has been serviced and completed.

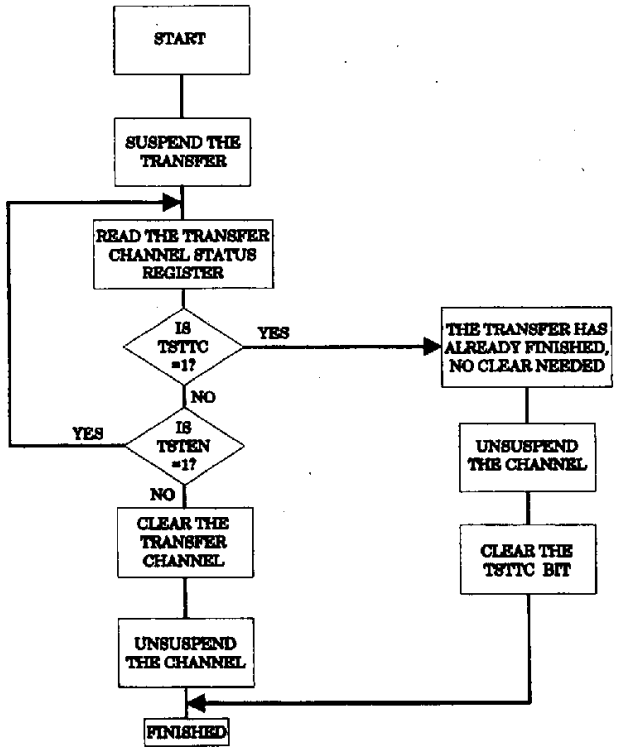
NOTE:
During a FIFO pause, CFGFF is ignored.

The CFGBR Bit defines the type of transfer cycles (burst or non-burst) that can be requested on the transfer channel. If burst cycles have been selected and system memory is unable to run burst cycles, the BMIC will default to non-burst (two BCLK) or mismatched cycles.

The CFGCL Bit is used to generate a channel clear. A channel clear terminates the current transfer and flushes the associated FIFO. The FIFO is reset during the next Base to Current register copy.

Before a channel is issued a clear command, the channel must first be suspended by writing a "1" into Bit 0 (CFGSU) of the Transfer Channel Configuration Register. Next the Transfer Channel Status Register must be read. If the TSTTC (Bit 0) is set to a "1", then the channel has already completed the transfer. The channel is then unsuspended (write a "0" into Bit 0 [CFGSU] of the Transfer Channel Configuration Register), and the TSTTC bit is then cleared.

If the TSTTC bit is a "0", then the TSTEN bit is checked. If this bit is a "1", then the channel has not returned to idle yet, and the Transfer Channel Status Register is re-polled. If the TSTEN bit is a "0", then the channel has successfully returned to idle and can now be cleared with no errors. This is done by setting Bit 2 (CFGCL bit) to a "1" in the Transfer Channel Configuration Register. A flowchart for this operation is shown in the following figure.



NOTE:
TSTTC must ALWAYS be cleared when a Transfer is completed.

Figure 8-1. Channel Clear Flowchart

If a channel is enabled for a transfer during a Channel clear, the BMIC will generate an end of process by asserting TEOP#. If the channel is not enabled for a transfer or the channel clear is preceded by a channel suspend, a TEOP# will not be generated. The channel clear will be active for at least two complete BCLK cycles.

The CFGEI Bit is a reserved bit. Zero (0) must always be written at this bit location. This bit can be ignored during register reads.

The CFGSU Bit is used to temporarily suspend the data transfer.

8.2.4.3 Channel 0 and 1 Transfer Channel Status Registers (Read/Write)

Each channel has an associated Transfer Channel Status register. Bits 2 through 4 are read only and bits 5 through 7 are reserved. Upon reset, the Channel Status register set is reset to "0".

Channel 0 Local Index Address—4Ah
Channel 1 Local Index Address—6Ah

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	TST1K	RSVD	TSTEN	TSTIP	TSTET	TSTTC

Bit 7, 6, 5 —W — Reserved. 0 should be written into these bits during writes. Ignore any data on these bits during register reads

Bit 4 — R — Reserved

Bit 3 — R — 1 = The transfer channel is enabled for transfer (transfer in progress)
0 = The transfer channel is not enabled for transfer (transfer not in progress)

Bit 2 — R — 1 = A transfer request is active on this channel.
0 = No transfer request is active on this channel.

Bit 1 — Reserved. Ignore data at this bit location.

Bit 0 — R/W — 1 = Transfer completed on this channel (read)
Reset this bit (write)
0 = No transfer completion on this channel (read)
No action (write)

Bits (7), (6), TST1K, and (4) are reserved. Any data read from these bits should be ignored.

The TSTIP and TSTEN Bits are read only and indicate whether the corresponding channel is requesting a transfer or whether the channel's transfer is currently in progress.

The TSTET Bit is a reserved bit and should be ignored during all register reads. Zero (0) should always be written at this bit location.

The TSTTC Bit is read/write and is used to indicate the current end-of-process status of the transfer. If an EOP occurs, the BMIC will set bit (0) to a "1" and generate an interrupt to the local processor. The BMIC will not generate the interrupt if the CFGIT bit in the channel's corresponding Transfer Configuration register is set to a "0".

NOTE:
The TSTTC bit is implemented as a sticky bit. This bit can be reset by the local processor without affecting the status of the other bits in the register.

8.2.5 PEEK/POKE REGISTERS

The Peek/Poke register set consists of four 8-bit Address registers, four 8-bit Data registers and one Peek/Poke control register. The Address and Data registers are used to define the 32-bit address and data that will be used during the peek/poke cycles, and the Control register is used to request and define the type of cycle that will be generated (peek, poke, or locked exchange). The peek/poke or locked exchange cycle is initiated by writing to the Peek/Poke control register. During Reset, the Peek/Poke Address registers and the Control register are reset to "0".

8.2.5.1 Peek/Poke Address Registers (Read/Write)

The four 8-bit Peek/Poke Address registers contain the 32-bit peek/poke address. Only the lower 16 bits are used for I/O cycles. Address bits 0 and 1 are ignored. Upon reset, this register is reset to "0".

Local Index Address—34h through 37h (bytes 0 through 3)

BYTE 3	BYTE 2	BYTE 1	BYTE 0
Bits 24–31	Bits 16–23	Bits 8–15	Bits 2–7

8.2.5.2 Peek/Poke Data Registers (Read/Write)

The four 8-bit peek/poke data registers hold the data for the peek/poke cycle. Each peek/poke data register is associated with one byte lane. During peek transfers, only those peek/poke data registers whose corresponding byte enable bit is set in the peek/poke control register contain valid data. During poke transfers, the data must be placed in the appropriate register as determined by the corresponding byte enable bit.

Local Index Address—30h through 33h (bytes 0 through 3)

BYTE 3	BYTE 2	BYTE 1	BYTE 0
Bits 24–31	Bits 16–23	Bits 8–15	Bits 0–7

The Shared register timings (t85, t93, t96–t98) are used when accessing the Peek/Poke Data registers.

8.2.5.3 Peek/Poke Control Register (Read/Write)

The Peek/Poke Control register is written to by the local processor when a peek/poke transfer is desired over the EISA bus. Upon reset, this register is reset to "0".

Local Index Address—38h

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

Bit 7 — Reserved. Set to 0

Bits 6, 5 — 10 = Do read cycle (peek)
01 = Do write cycle (poke)
11 = Do locked exchange cycle (peek/poke)
00 = Do Nothing (Nop)

Bit 4 — 1 = Do memory cycle
0 = Do I/O cycle

Bit 3
Bit 2
Bit 1
Bit 0
1 = Byte enable for given byte lane
0 = Byte disable for given byte lane

Bits (6) and (5) are used to define the type of cycle requested (peek, poke or locked exchange).

Bit (4) defines whether the cycle is memory or I/O.

Bits (3–0) are used to define the byte enables for the doubleword data written to or read from the Peek/Poke data register. Peek/Poke cycles will not be generated for illegal combinations of byte enables.

ILLEGAL COMBINATIONS OF BYTE ENABLES:

Bits 3–0	IBE# <3:0>
0000	1111
0101	1010
1001	0110
1010	0101
1011	0100
1101	0010

NOTE:
Bits 3–0 in the above register are active high whereas the EISA byte enables (IBE# <3:0>) are active low.

8.2.6 I/O RANGE DECODE REGISTERS

The I/O Decode Range register set consists of two I/O Decode Range Base Address registers and two I/O Decode Range Control Registers. The Address registers are used to define the address range of interest to the expansion board and the Control registers are used to define the decode range size, type of decode (slot specific or general), and the response of the local I/O (32-bit EISA or 8-bit EISA). The I/O decode register set controls the two IOSEL# pins on the BMIC. Each pin has an associated Address and Control register.

Upon reset, the I/O Decode Range registers are initialized according to the following table:

Local Processor	*Local Processor Not Present		*Local Processor Present	
Range	**Rng 0**	**Rng 1**	**Rng 0**	**Rng 1**
Control Registers	60h	60h	20h	20h
Address Registers	E0h	00h	00h	00h

*Refer to Section 7.3 for information regarding "local processor present" or "local processor not present".

8.2.6.1 Range 0 and 1 I/O Decode Base Address Registers (Read/Write)

Each Decode range and IOSEL# pin has an associated I/O Decode Range Base Address register.

Range 0 Local Index Address—39h
Range 1 Local Index Address—3Bh

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

During general I/O decode, bits 7–0 are used to compare against EISA address lines LA<9:2>. During slot specific decode, bits 7 and 6 are compared against EISA address lines LA<11:10> and bits 5–0 are compared against EISA address lines LA<7:2> (refer to Section 4.8).

8.2.6.2 Range 0 and 1 I/O Decode Control Registers (Read/Write)

Each Decode range and IOSEL# pin has an associated I/O Decode Range Control register.

Range 0 Local Index Address—3Ah
Range 1 Local Index Address—3Ch

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

Bit 7 — 1 = Respond as a 32-bit EISA I/O device
0 = Respond as an 8-bit EISA I/O device

Bit 6 — 1 = Slot Specific I/O Decode
0 = General I/O Decode

Bit 5 — 1 = IOSEL# held during CMD# active
0 = IOSEL# follows I/O address changes

Bit 4, Bit 3 — 1 = Do not compare I/O Range Base Address Register and corresponding EISA address bit (Mask)

Bit 2, Bit 1, Bit 0 — 0 = Compare I/O Range Base Address Register with corresponding EISA address bit

Refer to Section 4.8 for a complete description of the I/O Decode Range Control registers and the BMIC decode function in general.

8.2.7 TRANSFER BUFFER INTERFACE (TBI) REGISTERS (READ/WRITE)

The TBI registers are programmed to provide the 16-bit start address of the data transfer for use by the transfer buffer logic (refer to Section 5.3). Each transfer channel has a corresponding TBI Base and Current Address register set. The contents of the TBI Base Address registers are transferred to the TBI Current Address registers during a write to the channel's corresponding Transfer Channel Strobe Register.

8.2.7.1 Channel 0 and 1 TBI Base Address Registers (Read/Write)

The BMIC provides two 8-bit TBI Base Address registers per channel. The registers are programmed with the 16-bit start address of the data in the Transfer Buffer memory space. The TBI Base Address register set is not initialized to a fixed value upon reset.

Channel 0 Local Index Address—4Bh and 4Ch (byte 0 and 1)
Channel 1 Local Index Address—6Bh and 6Ch (byte 0 and 1)

Byte 1	Byte 0

8.2.7.2 Channel 0 and 1 TBI Current Address Registers (Read Only)

The BMIC provides two 8-bit TBI Current Address registers per channel. The TBI Current Address registers contain the 16-bit real-time address of the data transfer. The contents of the Current register set are transferred to the external buffer logic at the beginning of every new data block transfer. The BMIC may also be programmed to transfer the contents of the Current Address register each time the corresponding channel regains control of the bus (refer to Section 5.3). The TBI Current Address register set is reset to "0" upon device reset.

Channel 0 Local Index Address—58h and 59h (byte 0 and 1)
Channel 1 Local Index Address—78h and 79h (byte 0 and 1)

Byte 1	Byte 0

NOTE:
The TBI current registers contain real-time status and may change at anytime. If a stable value is needed while reading a set of these registers, the channel should be temporarily suspended by setting the CFGSU bit in the Channel Configuration register to a "1" before these registers are read.

9.0 DETAILED PIN DESCRIPTION

9.1 EISA Interface Signals

Pin Name	Description
START#	I/O TRI-STATED (EISA CYCLE START STROBE) The START# signal provides timing control at the start of a cycle. During EISA master mode, the BMIC drives this signal low after LA<31:2> and M/IO become valid and negates START# on the rising edge of BCLK after one BCLK cycle time. During EISA slave mode, the BMIC uses this signal to indicate the start of a slave bus cycle. It is sampled on the rising edge of BCLK. Upon reset, this pin is tri-stated and placed in input mode.
CMD#	INPUT (EISA COMMAND STROBE) The CMD# provides timing control within the cycle. The 82358 Bus Controller asserts CMD# on the rising edge of BCLK, simultaneously with the negation of START#. CMD# is held low until the end of the cycle. The BMIC uses CMD# in EISA slave mode for timing control during internal Shared register read/write accesses.
M/IO	I/O TRI-STATED (EISA MEMORY/IO CYCLE STATUS PIN) M/IO is used to indicate that the type of cycle in progress is a memory cycle (high) or I/O cycle (low). M/IO is pipelined from one cycle to the next and must be latched by the addressed memory slave if needed for the whole cycle. During EISA master mode, the BMIC drives this signal. The BMIC will drive this pin high during burst and non-burst (two BCLK) cycles. The value of M/IO in a Peek/Poke or locked exchange cycle depends on the programmed value of bit 4 in the Peek/Poke Control register. During EISA slave mode, the M/IO pin is an input. As a slave, the BMIC will respond only as an I/O device. Upon reset, this pin is tri-stated and placed in input mode.
W/R	I/O TRI-STATED (EISA WRITE/READ CYCLE STATUS PIN) The W/R status signal identifies the cycle as a write (high) or read (low). W/R is pipelined from one cycle to the next and must be latched by the addressed memory slave if needed for the whole cycle. During EISA master mode, the BMIC drives this signal. During EISA slave mode, this pin is an input. Upon reset, W/R is tri-stated and placed in input mode.
EXRDY	I/O OPEN COLLECTOR (EISA READY SIGNAL) EXRDY is used by EISA I/O and memory slaves to request wait states during a cycle. Each wait state is one BCLK period. During EISA master mode, the BMIC first samples this signal on the falling edge of BCLK after CMD# is asserted. If it is low, the BMIC will insert a wait state, and continue inserting wait states as long as EXRDY is low at each successive falling edge of BCLK. During EISA slave mode, the BMIC drives EXRDY inactive until it is ready to complete cycles addressed to it. The EXRDY pin is an open collector output.
EX32#	I/O OPEN COLLECTOR (EISA 32-BIT SLAVE RESPONSE PIN) EX32# is an open collector and is used by memory or I/O slaves to indicate their support of 32-bit transfers. During EISA master mode, the BMIC samples EX32# on the same rising edge of BCLK that START# is deasserted. The BMIC uses this pin to determine if the addressed slave is capable of 32-bit transfers. During peek/poke and non-burst EISA data transfers, the BMIC is a 32-bit master only and will allow the 82358 Bus Controller to do all necessary bus conversions. During EISA slave mode, the BMIC drives EX32# low if it has 32-bit data to send to the EISA bus, otherwise this signal is inactive.

9.1 EISA Interface Signals (Continued)

Pin Name	Description
MASTER16#	OUTPUT OPEN COLLECTOR (EISA 16-BIT MASTER CONTROL) In master mode, the BMIC will assert MASTER16# (at the same time as START#) for one BCLK period when it is capable of downshifting from a 32-bit master to a 16-bit master. The BMIC will downshift if necessary during memory burst transfers only. The BMIC will automatically downshift from a 32- to 16-bit master if the EX32# signal is sampled inactive and the SLBURST# signal is sampled active. MASTER16# has no function in slave mode.
AEN	INPUT (EISA ADDRESS ENABLE SIGNAL) The BMIC uses AEN when in EISA slave mode to qualify I/O addresses. When negated (low), the BMIC uses AEN to decode possible accesses to its general and slot specific I/O space. When asserted (high), the address on the EISA bus will be ignored by the BMIC. AEN is sampled on the falling edge of CMD#. This signal is not used in master mode.
MSBURST#	OUTPUT TRI-STATED (EISA MASTER BURST SIGNAL) The BMIC asserts MSBURST# to indicate to the addressed memory slave that the BMIC will provide burst cycles. If the BMIC samples SLBURST# active on the rising edge of BCLK after START# is asserted, the BMIC will activate MSBURST# on the next BCLK falling edge and will proceed with burst cycles. If the BMIC samples SLBURST negated, MSBURST# will not be activated and the BMIC will proceed with either non-burst (two BCLK) or mismatched cycles, depending on the size of the slave device addressed. This signal is not used in slave mode. Upon reset, this pin is tri-stated.
SLBURST#	INPUT (EISA SLAVE BURST SIGNAL) The BMIC uses this signal in master mode to determine if the addresed slave memory is capable of supporting burst transfers. If the BMIC samples SLBURST# active on the rising edge of BCLK after START# is asserted, the BMIC will proceed with burst cycles. If the BMIC samples SLBURST# negated, either non-burst (two BCLK) or mismatched cycles will be generated.
LOCK#	OUTPUT TRI-STATED (EISA RESOURCE LOCK SIGNAL) The BMIC asserts this signal to guarantee exclusive memory and I/O access during locked peek/poke exchange. Upon reset, this pin is tri-stated.
MREQ#	OUTPUT (EISA MASTER BUS REQUEST SIGNAL) MREQ# is asserted by the BMIC to request EISA bus access. The BMIC will begin driving the bus with the address and control signals on the falling edge of BCLK, two BCLKs after MAK# is sampled active. During an EISA write transfer, MREQ# will not be asserted until the FIFO on the selected channel is full. During an EISA read transfer, MREQ# will be asserted immediately after receiving a transfer request, assuming that a slave cycle is not currently in progress. Upon reset, this pin is driven inactive high.
MAK#	INPUT (EISA MASTER BUS ACKNOWLEDGE SIGNAL) The MAK# signal is asserted by the 82357 (ISP) to grant EISA bus access to the BMIC. The BMIC samples MAK# on the falling edge of BCLK and will begin driving the bus with the address and control signals on the falling edge of BCLK, two BCLKs after MAK# is sampled active. The MAK# signal may be negated by the ISP to indicate to the BMIC that another device requires EISA bus access. The BMIC will negate MREQ# to release the bus within 64 BCLKs (8 μs) of sampling MAK# negated.
EINT	OUTPUT OPEN COLLECTOR (EISA INTERRUPT REQUEST SIGNAL) The EINT line is used by the BMIC to interrupt the system CPU or EISA bus master to request service. EINT can be programmed for either edge or level-triggered operations and is an open collector output in level-triggered mode. Upon reset, EINT is placed in level-triggered mode and floating.
BCLK	INPUT (EISA BUS CLOCK) This clock signal is used by the BMIC to synchronize the EISA control signals and data transfers to the system clock. BCLK typically runs at a frequency of 8.33 MHz with a normal duty cycle of 50%. The BCLK period is sometimes extended by the 82358 (EBC) by up to one BCLK period for synchronization purposes.

9.1 EISA Interface Signals (Continued)

Pin Name	Description
RESET	INPUT (EISA RESET SIGNAL) This signal is used by the BMIC to initialize all of its internal registers and state machines to a known state. This signal is asynchronous with respect to BCLK. To reset the BMIC properly, the RESET signal must be active for eight BCLK periods.
IDAT<31:0>	I/O TRI-STATED (EISA DATA LINES/UPPER 22 ADDRESS LINES) These data signals interface to the EISA bus through external, 74F245 bi-directional TTL buffers. The upper 22 data lines are also multiplexed to function as the upper 22 EISA address lines. The 22 upper address signals are latched into external 74F573 TTL latches during transfers as necessary by the BMIC. Both the external data buffers and the address latches are controlled by the BMIC during all slave and master mode data transfers. Upon reset, these pins are tri-stated.
IADS<11:10>	(INPUT) (EISA ADDRESS INPUT LINES) These two address lines are input only and are only used during slave mode. They are used along with IADS<9:2> and EISA byte enables IBE<3:0># for I/O address decoding. The corresponding EISA output address lines LA<11:10> are part of the upper 22 address lines that are multiplexed and sent out through the upper 22 data lines.
IADS<9:2>	I/O TRI-STATED (EISA LOWER ADDRESS LINES) These eight address lines are part of the lower EISA address lines and are connected directly to the EISA bus. When the BMIC is a master, it drives these lines directly to the EISA bus. The upper 22 addresses are latched from the data bus. IADS<9:2> are pipelined from one cycle to the next and should be latched by the addressed slave if required for the whole cycle. When the BMIC is a slave, it monitors these lines along with EISA address lines IADS<11:10> and EISA byte enables IBE<3:0># for I/O address decoding. Upon reset, these pins are tri-stated and placed in input mode. The following address lines are used during I/O decoding as shown: Slot specific I/O address decoding (expansion board)—IADS<11:2> Slot specific I/O address decoding (shared registers)—IADS<11:2>/IBE<3:0># General I/O address decoding (expansion board)—IADS<9:2>
IBE<3:0>#	I/O TRI-STATED (EISA BYTE ENABLES) IBE#<3:0> are the byte enables of the EISA bus and identify the specific bytes that are active during the current EISA bus cycle. During EISA master mode, the BMIC drives these signals. IBE#<3:0> are pipelined from one cycle to the next and should be latched by the addressed slave if required for the whole cycle. During EISA slave mode, the byte enables are inputs and are used along with EISA address lines IADS<11:2> for internal shared register decoding. Upon reset, these pins are tri-stated and placed in input mode.

9.2 EISA Buffer Control Signals

Pin Name	Description
UALOE#	OUTPUT (EISA UPPER ADDRESS LATCH STROBE AND OUTPUT ENABLE) The UALOE# signal is used by the BMIC to control the external latching of the upper 22 address lines LA<31:10>. UALOE# is designed to be connected to the latch enables and output enables of the 74F573 external address latches. The BMIC updates the external address latches at the beginning of all master mode transfers. The desired address value is placed on the IDAT<31:10> lines and latched by the external latches on the falling edge of UALOE# at the beginning of the transfer. During EISA master mode to enable the EISA address lines <31:10>, the BMIC drives UALOE# low on the rising edge of BCLK, one BCLK prior to the falling edge of START#. UALOE# will remain active until the end of the cycle. During slave mode, the BMIC holds UALOE# high to disable the latches. For additional information with regards to the timing for this signal, refer to the A.C. timing and Basic Function timing sections. Upon reset, this pin is driven inactive high.
IDDIR	OUTPUT (EISA DATA DIRECTION SIGNAL) The IDDIR signal is used by the BMIC to control the direction of the external 74F245 data buffers. During data transfers from the BMIC to the EISA bus, this signal will be driven low. During data transfers from the EISA bus to the BMIC, this signal will be driven high. For additional information regarding the timing for this signal, refer to the A.C. timing and Basic Function timing sections (master and slave). Upon reset, this pin is driven high.
IDOE23# IDOE1# IDOE0#	OUTPUT (EISA DATA BYTE LANE BUFFER ENABLES) The IDOE# signals are used by the BMIC to control the output enables on the external 74F245 data buffers. The IDOE# signals will be driven so that the data buffers are enabled at the appropriate times during master and slave transfers. For additional information with regards to the timing for these signals, refer to the A.C. timing and Basic Function timing sections. Upon reset, these signals are driven inactive high.

9.3 Address Decode Signals

Pin Name	Description
IOSEL# <1:0>	OUTPUT (ADDRESS RANGE DECODE OUTPUTS) The IOSEL# signals are used by the BMIC to enable external logic on the expansion board during slot specific and general purpose I/O decode. These pins become active when the LA<11:2> address lines on the EISA bus contain a value mapped into one of the two possible I/O address decode ranges provided by the BMIC (refer to Section 4.8). Upon reset, these pins are driven inactive high.

9.4 Transfer Buffer Interface Signals

Pin Name	Description
TRQ#	OUTPUT (LOCAL DATA TRANSFER REQUEST SIGNAL) When a data transfer is desired over the Transfer Buffer interface, TRQ# is driven low, indicating to the transfer buffer logic that a transfer is following. TRQ# will remain active until the data transfer is completed or a transfer interruption occurs. Upon reset, this pin is driven inactive high.
TACK#	INPUT (LOCAL DATA TRANSFER ACKNOWLEDGE SIGNAL) External logic uses this signal to acknowledge the transfer of a data item (16-bit word) over the Transfer Buffer interface.
TLD#	OUTPUT (LOCAL ADDRESS COUNTER LOAD SIGNAL) This signal when asserted (low) is used to load the transfer start address and the transfer real-time address into an external address counter as required for data transfers (refer to Section 5.3). TLD# is asserted at the beginning of all new channel accesses to the transfer buffer logic and will remain asserted until acknowledged by TACK#. Upon reset, this pin is driven inactive high.
TDIR	OUTPUT (DATA TRANSFER DIRECTION SIGNAL) This signal is used to inform the transfer buffer logic as to the direction of the current data transfer. When driven (high) data will be transferred from the EISA bus to the expansion board. When driven (low) data will be transferred from the expansion board to the EISA bus. TDIR will be held valid whenever TLD# and TRQ# are active. TDIR will not change states when TRQ# is active. Upon reset, this pin is driven high.
TCHAN	OUTPUT (TRANSFER CHANNEL SELECT SIGNAL) This signal is used by the BMIC to inform the transfer buffer logic as to which channel will be active during the transfer. When driven (low) transfer channel 0 is active and when driven (high) transfer channel 1 is active. TCHAN has the same timings as TDIR and will not change states when TLD# or TRQ# are active. Upon reset, this pin is driven low.
TDAT<15:0>	I/O TRI-STATED (TRANSFER DATA LINES) This bidirectional bus is the BMIC's Transfer Buffer interface data bus. It is used during data transfers between the external transfer buffer logic and the BMIC. The data transferred across the TDAT bus is word aligned. The data lines are also used to transport the transfer address to the transfer buffer logic on the expansion board (refer to Section 5.3). The TDAT bus can be unconditionally disabled by driving the TDOE# signal high. **NOTE:** During EISA write data transfers, the TDAT lines are inputs and operate independent of the value of TDOE#. Upon reset, the TDAT bus is tri-stated.
TDOE#	INPUT (TRANSFER INTERFACE DATA OUTPUT ENABLE) When driven high, this pin can be used by external logic to unconditionally disable the BMIC from driving the TDAT<15:0> lines. This feature eliminates the need for the BMIC to gain prior permission to drive the TDAT bus and also allows external logic the ability to time-share the TDAT bus.
TEOP#	OUTPUT OPEN COLLECTOR (TRANSFER END-OF-PROCESS SIGNAL) This signal is an open collector signal that indicates the end of a transfer to the external transfer buffer logic. TEOP# is driven low by the BMIC to indicate the end of transfer. The TEOP# pin requires an external 2.5K to 3.2K pullup resistor for proper operation.
TCLK	INPUT (TRANSFER CLOCK) All transfer control signals are synchronous to this clock. The frequency should be in the range of 16 MHz to 20 MHz to maintain a 33 Mbyte/sec burst transfer rate over the EISA bus. This clock may be completely asynchronous to the EISA BCLK signal.

9.5 Local Processor Interface Signals

Pin Name	Description
LDAT<7:0>	I/O TRI-STATED (LOCAL PROCESSOR INTERFACE DATA BUS) This bidirectional bus is used to transfer commands and status between the BMIC and the local processor on the expansion board. If a local Processor is not present, this bus will need to be connected to the ISA bus (refer to Section 7.3). Upon reset these pins are tri-stated.
LRD#	INPUT (LOCAL PROCESSOR INTERFACE READ STROBE) The local processor asserts LRD# to indicate to the BMIC that it should drive its data onto the LDAT bus. LRD# is asserted for register access to the BMIC's Local Processor interface. The LADS lines and the LCS# signal must be valid 10 ns before the falling edge of LRD# and remain valid until LRD# is deasserted.
LWR#	INPUT (LOCAL PROCESSOR INTERFACE STROBE) The local processor asserts LWR# to indicate to the BMIC that it may latch data from the LDAT bus. LWR# is asserted for write accesses to the BMIC's Local Processor interface. The LADS lines and the LCS signal must be valid 10 ns before the falling edge of LWR# and remain valid until LWR# is deasserted.
LCS#	INPUT (LOCAL PROCESSOR INTERFACE CHIP) A (low) on this pin enables LWR# and LRD# communication between the BMIC and the local processor on the expansion board. The LRD# and LWR# signals are ignored unless the LCS# signal is active. LCS# must be asserted 10 ns before LRD# and LWR# and remain active until the inactive edge of LRD# and LWR#.
LADS<1:0>	INPUT (LOCAL PROCESSOR ADDRESS SELECT) These address lines are used by the local processor to select the Local Data, Local Index, and Local Status/Control registers. The BMIC uses these registers as part of an indexing scheme to access all of its internal registers (refer to Sections 3.2.6.1 and 8.2.1). **LADS1 LADS0** 0 0 = Local Data register 0 1 = Local Index register 1 0 = Local Status/Control register 1 1 = Reserved
LINT	OUTPUT (LOCAL PROCESSOR INTERRUPT SIGNAL) This signal informs the local processor that an event has occurred which requires the local processor's attention. This pin can be programmed for either active high or active low level operations. After being asserted, LINT will not return to an inactive state until the interrupt has been serviced. The LINT signal is not an open collector output during active low operations and will require external logic if interrupts need to be tied together on the local side. Upon reset, this pin is driven high and placed in active low level mode.
LRDY	I/O (LOCAL PROCESSOR READY) This signal is the acknowledgement from the BMIC to the local processor that it is finished with the current Shared register access cycle. The LRDY pin is also used by external logic to indicate to the BMIC that a local processor is not present. If a local processor is not present, the LRDY signal must be driven low during reset (refer to Section 7.3). If a local processor is present, a weak pullup resistor must be connected to the LRDY output to insure that LRDY is high during the time reset is active.

9.6 Power Supplies

V_{CC} — 11 Power pins

V_{SS} — 13 Ground pins

Total number of power supply pins: 24

10.0 BASIC FUNCTION TIMING DIAGRAMS

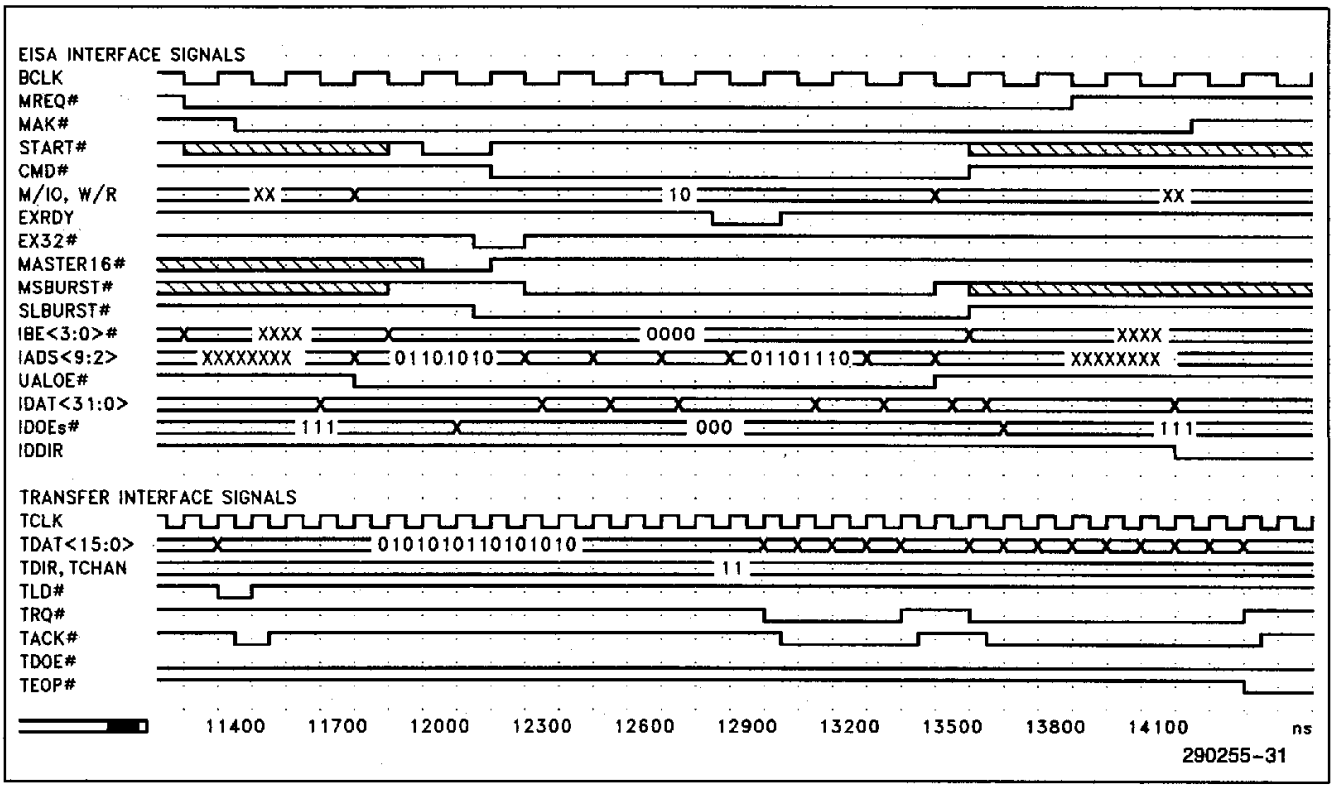


Figure 10-1. 32-Bit Burst Cycle (EISA Read)

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

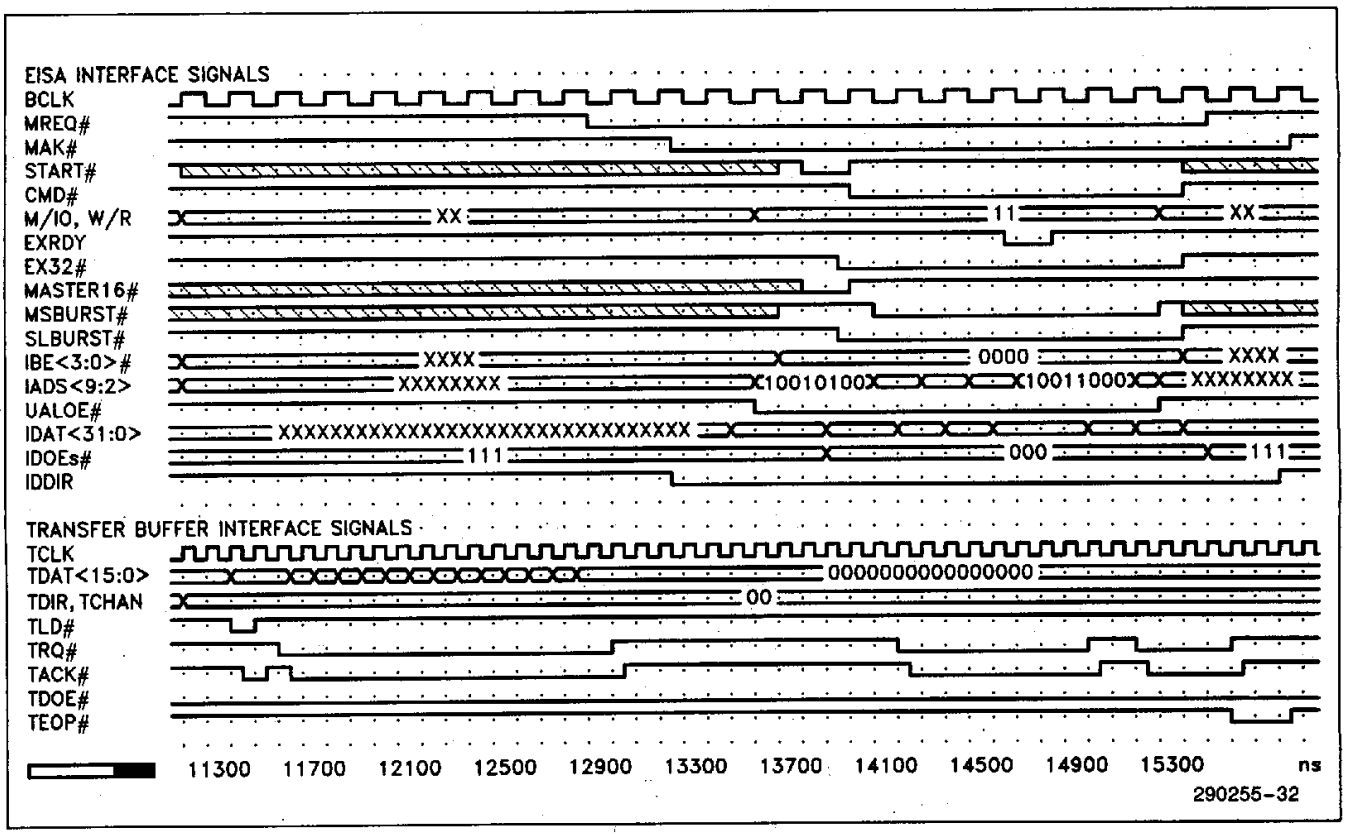


Figure 10-2. 32-Bit Burst Cycle (EISA Write)

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

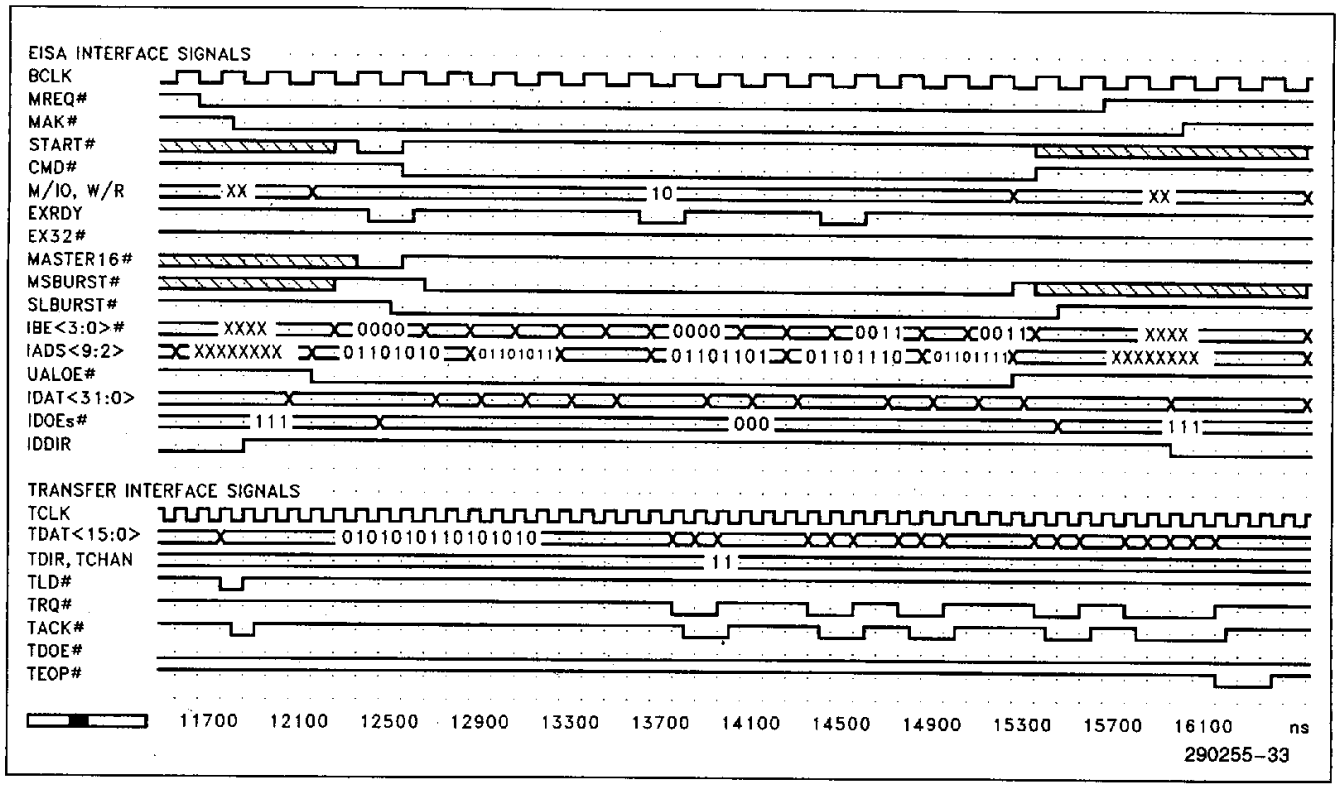


Figure 10-3. 16-Bit Burst Cycle (EISA Read)

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

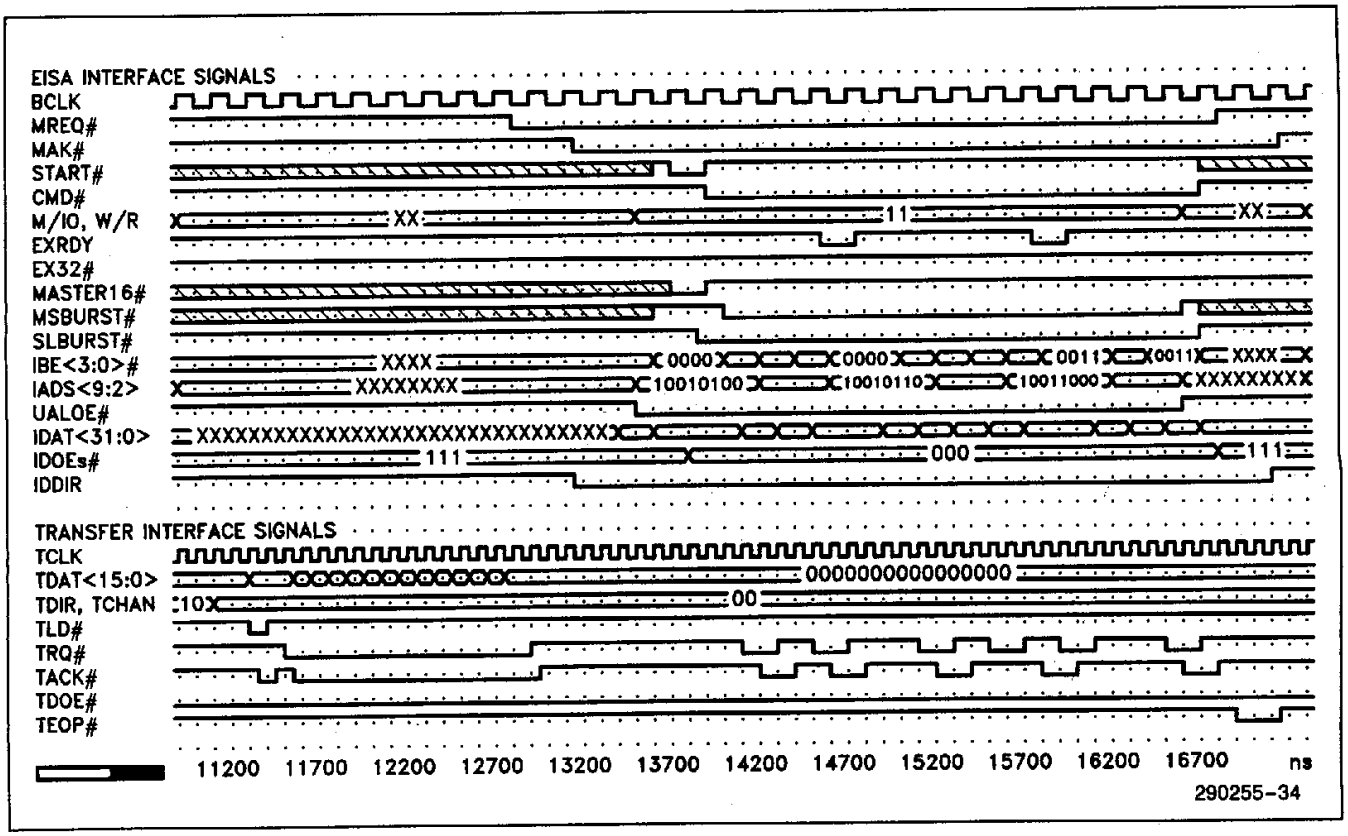


Figure 10-4. 16-Bit Burst Cycle (EISA Write)

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

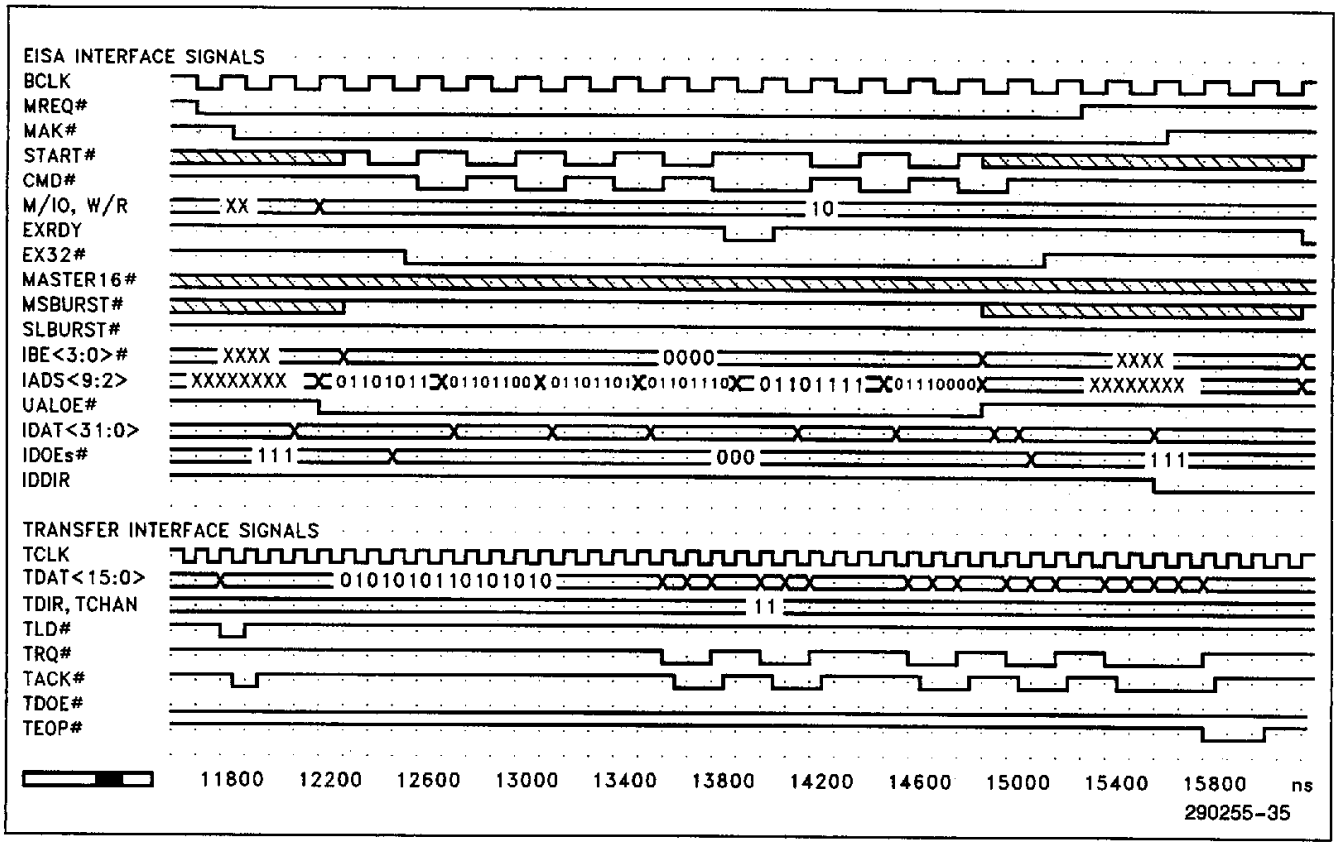


Figure 10-5. 32-Bit Non-Burst Cycle (EISA Read)

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

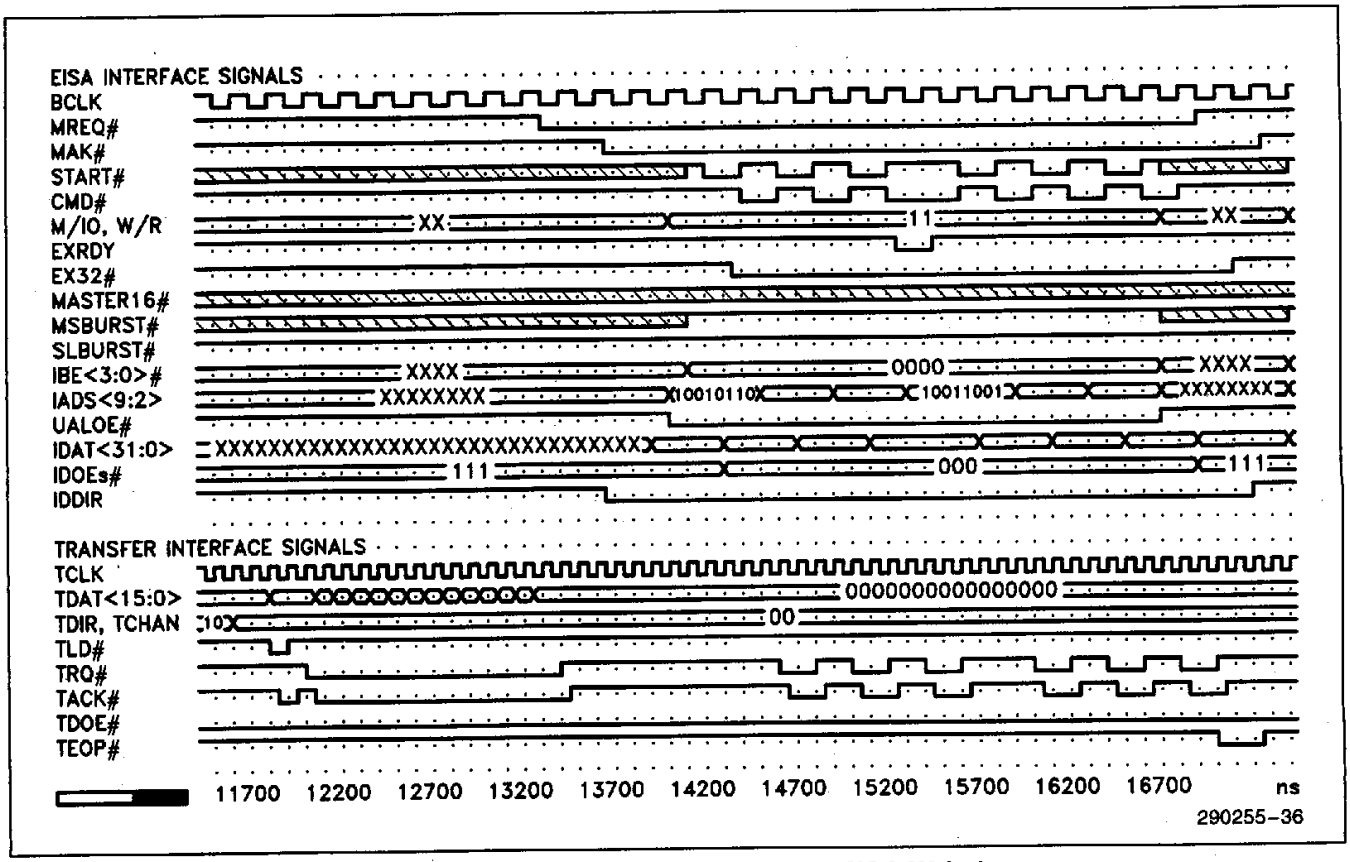


Figure 10-6. 32-Bit Non-Burst Cycle (EISA Write)

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

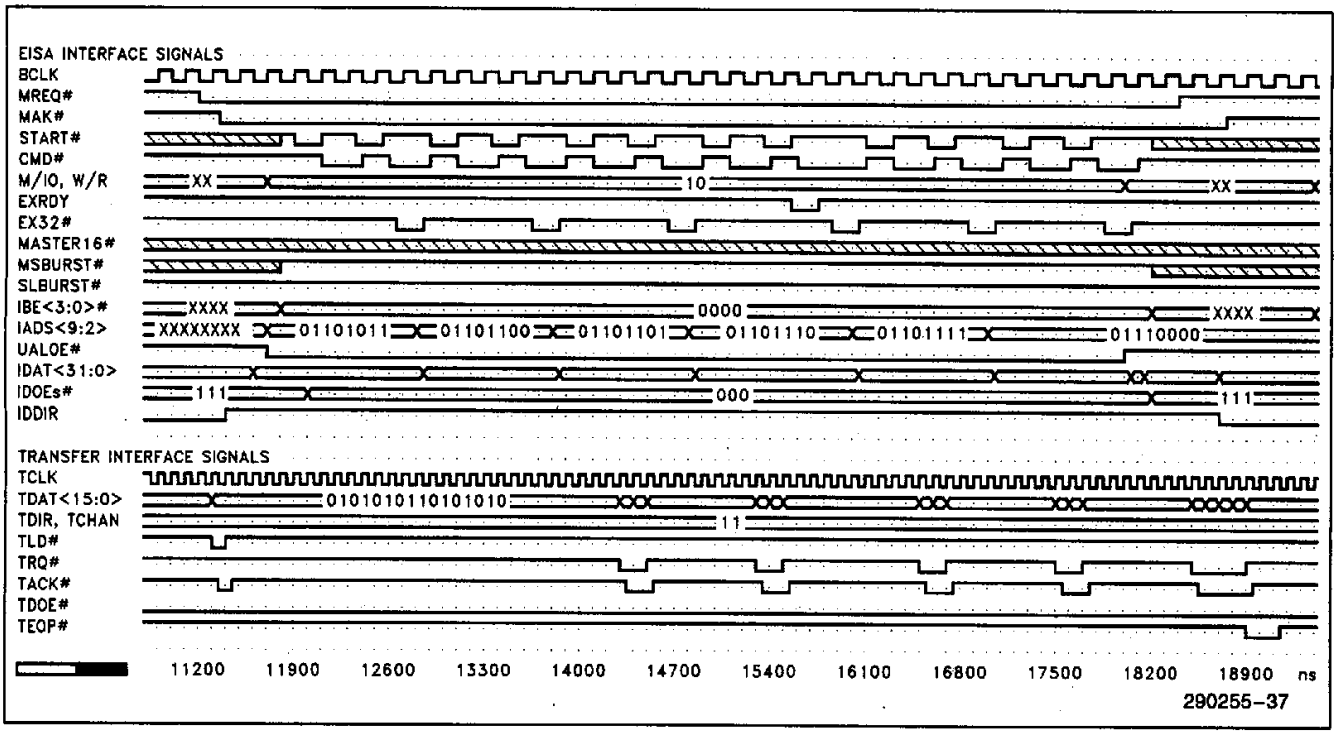


Figure 10-7. Mismatched Cycle (EISA Read)

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

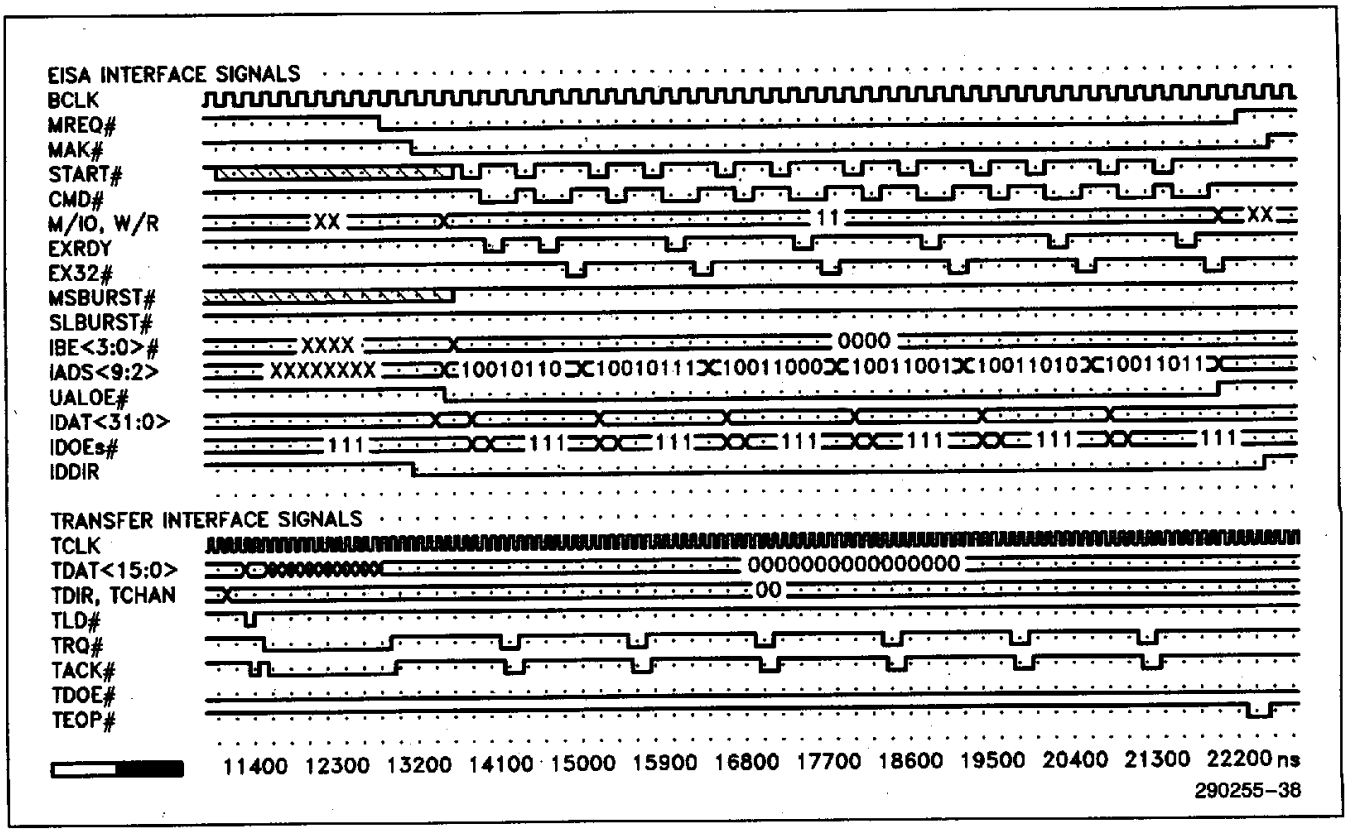


Figure 10-8. Mismatched Cycle (EISA Write)

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

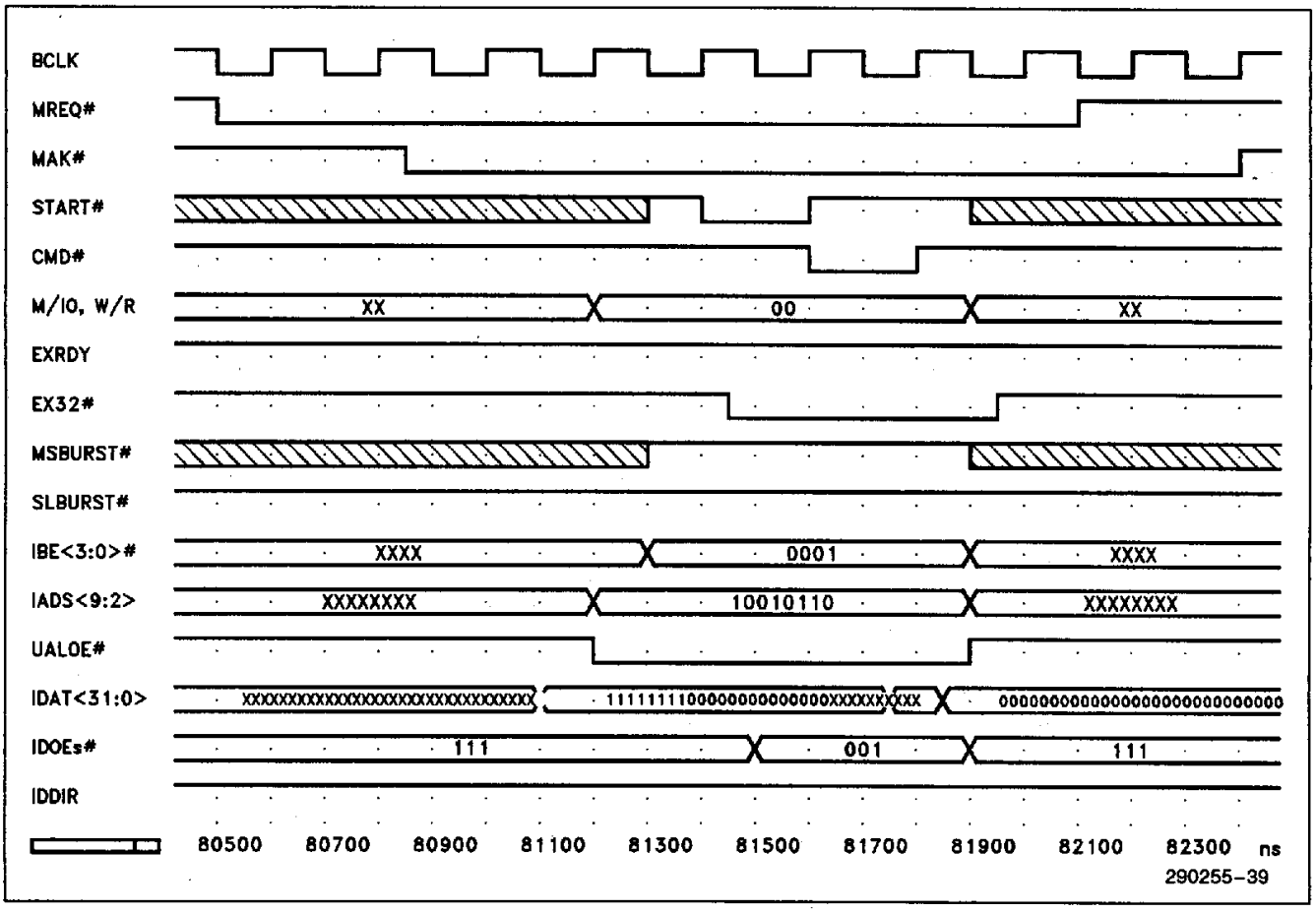


Figure 10-9. I/O Peek Cycle

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

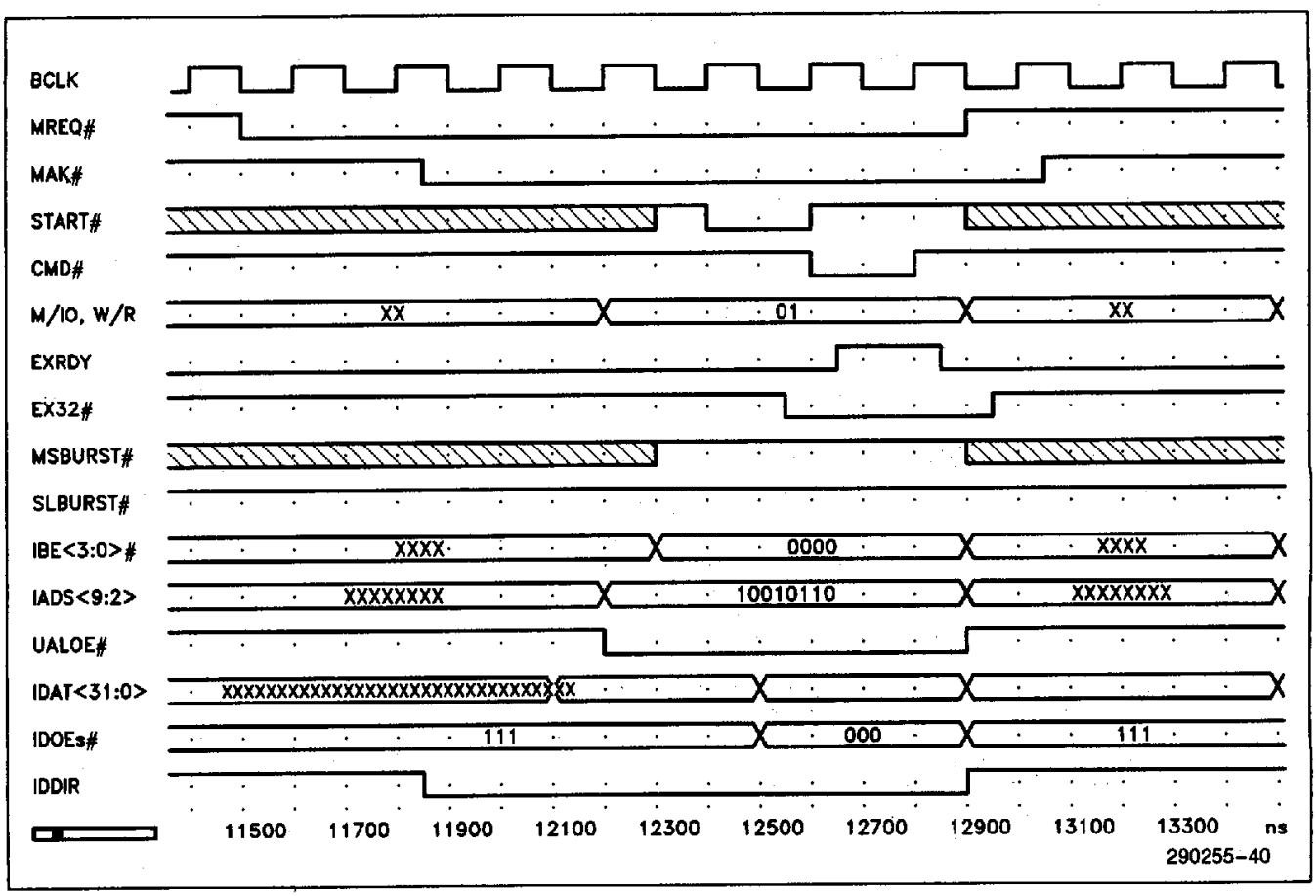


Figure 10-10. I/O Poke Cycle

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

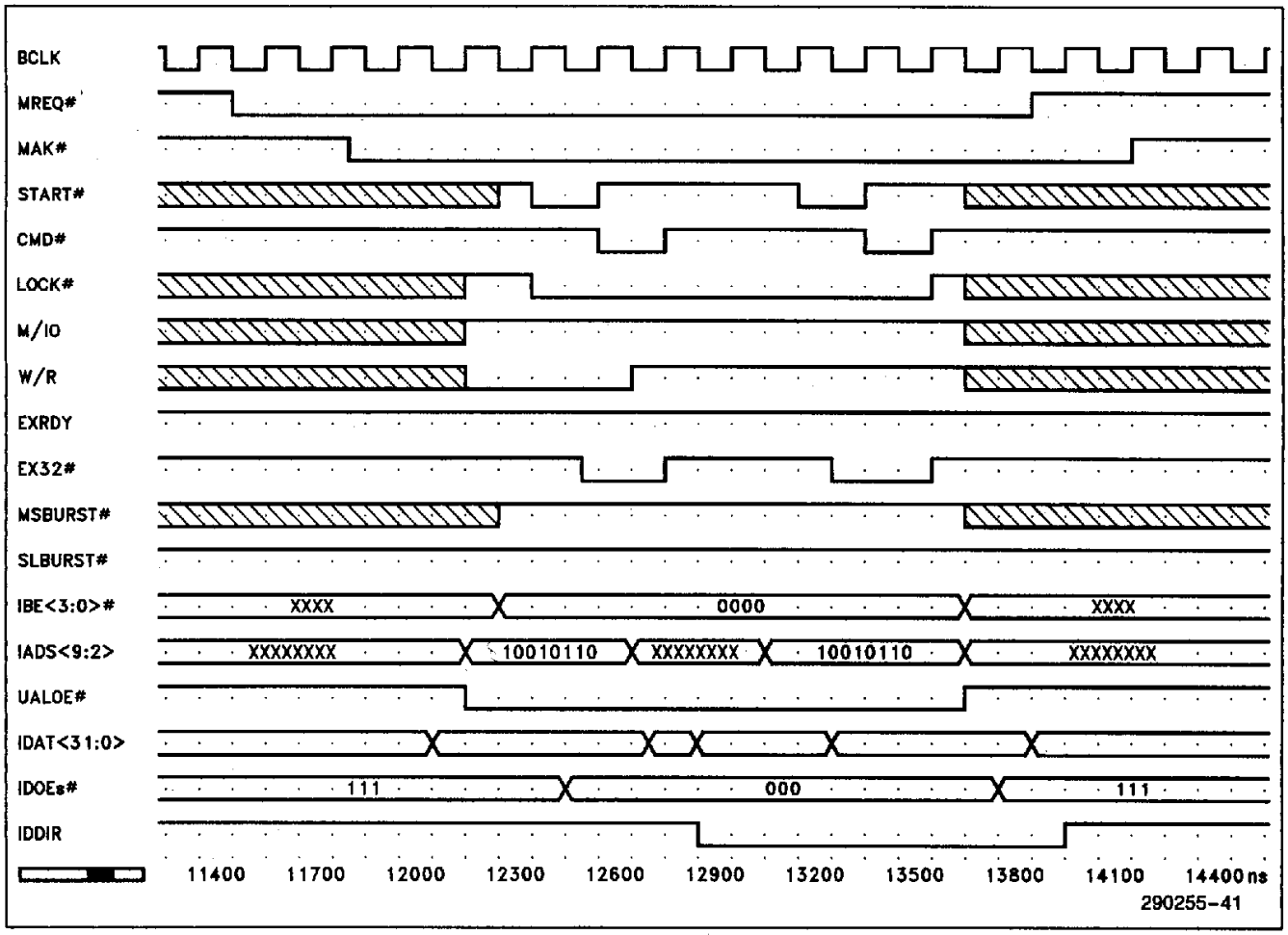


Figure 10-11. Locked Exchange Cycle

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

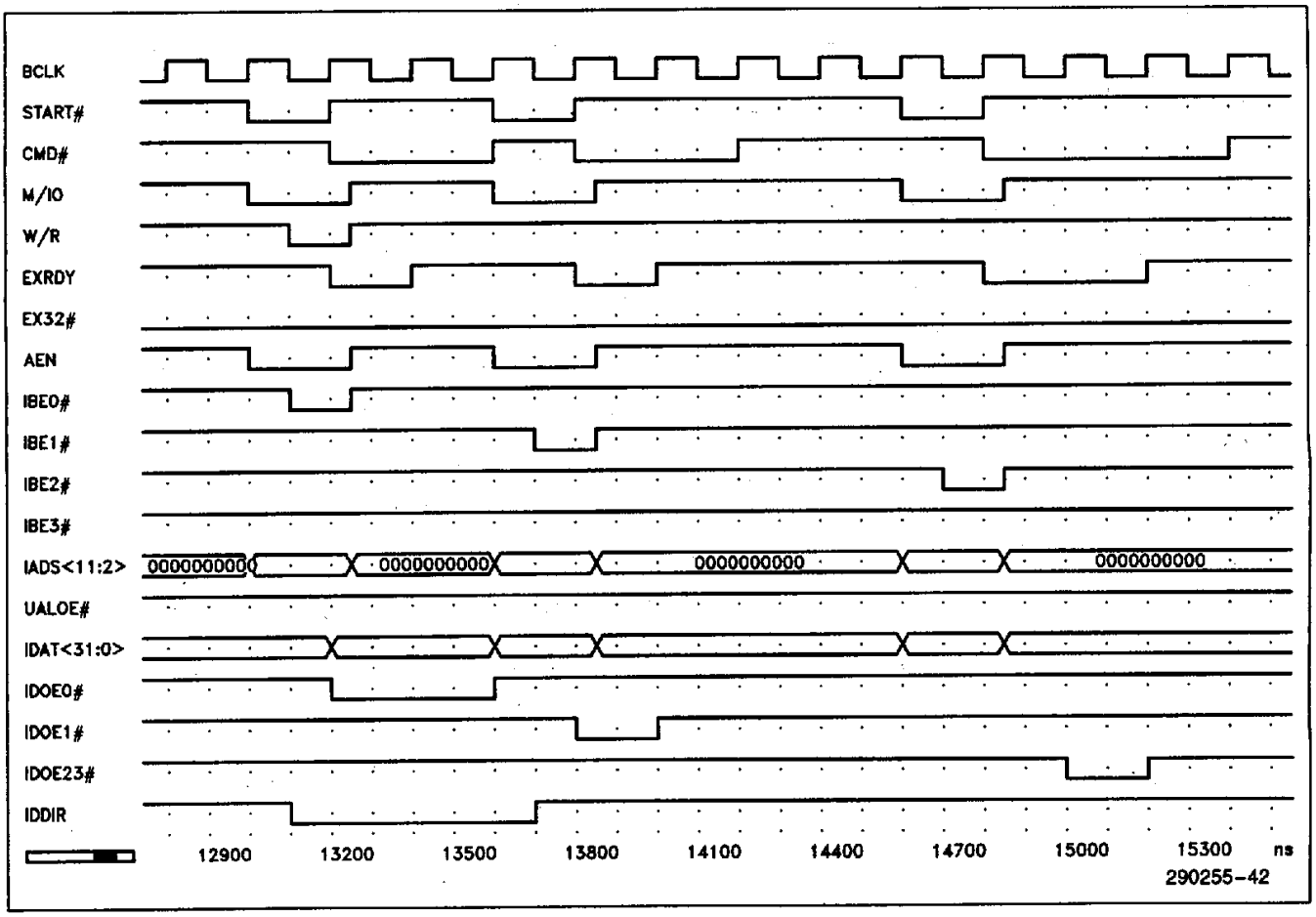


Figure 10-12. Slave Access to BMIC

11.0 D.C. SPECIFICATIONS

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

11.1 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 +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.

1

11.2 D.C. Characteristics Table

T_{CASE} = 0°C to 70°C, V_{CC} = 5V ±5%, $T_{AMBIENT}$ = 0°C to 55°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}	CLOCK Input Low	−0.5	0.8	V	
V_{IHC}	CLOCK Input High	2.0	V_{CC}+0.5	V	
V_{OL1}	Output Low Voltage		0.45	V	I_{OL} = 2.5 mA
V_{OH1}	Output High Voltage	2.4		V	I_{OH} = −2.5 mA
V_{OL2}	Output Low Voltage		0.45	V	I_{OL} = 6 mA
V_{OH2}	Output High Voltage	2.4		V	I_{OH} = −4 mA
V_{OL3}	Output Low Voltage		0.45	V	I_{OL} = 24 mA
V_{OH3}	Output High Voltage	V_{CC}−0.4		V	I_{OH} = −100 µA
V_{OL4}	Output Low Voltage		0.45	V	I_{OL} = 4.0 mA
V_{OH4}	Output High Voltage	2.4		V	I_{OH} = −2.5 mA
ILI	Input Leakage		±10	µA	
ILO	Output Leakage		±10	µA	
C_{IN}	Capacitance Input		10	pF	@ 1 MHz(2)
C_{OUT}	Capacitance Output or I/O		12	pF	@ 1 MHz(2)
C_{CLK}	BCLK or TCLK		15	pF	@ 1 MHz(2)
I_{CC}	V_{CC} Supply Current		190	mA	(3)

NOTES:

1. V_{OL1} = IDOE23#, IDOE1#, IDOE0#, LRDY, LDAT<7:0>, IDAT<31:0>, TEOP#, TDIR, TCHAN, IOSEL0#, IOSEL1#, TRQ#, TLD#, and TDAT<15:0>
   V_{OL2} = MREQ#, EINT, and LINT
   V_{OL3} = IADS<9:2>, START#, M/IO, W/R, EXRDY, MASTER16#, EX32#, IBE#<3:0>, MSBURST#, and LOCK#
   V_{OL4} = UALOE#, IDDIR
   V_{OH1} = IDOE23#, IDOE1#, IDOE0#, LRDY, LDAT<7:0>, IDAT<31:0>, TDIR, TCHAN, TRQ#, TLD#, IOSEL0#, IOSEL1#, TDAT<15:0>, MREQ#, EINT, LINT, and TEOP#
   V_{OH2} = IADS<9:2>, START#, M/IO, W/R, IBE#<3:0>, MSBURST#, LOCK#, EXRDY, EX32#, and MASTER16#
   V_{OH3} = UALOE#, IDDIR, IDOE23#, IDOE1#, IDOE0#, IADS<9:2>, LRDY, LDAT<7:0>, IDAT<31:0>, TDIR, TCHAN, EINT, IOSEL0#, IOSEL1#, TRQ#, TLD#, TDAT<15:0>, MREQ#, LINT, IADS<9:2>, START#, M/IO, W/R, IBE#<3:0>, MSBURST#, LOCK#, and TEOP#
   V_{OH4} = UALOE#, IDDIR
   The following outputs are open collector: EXRDY, EX32#, MASTER16#, and TEOP#; EINT is an open collector output when programmed for active low operation.
2. Sampled only
3. Tested at V_{CC} = 5.30V and Frequency = BCLK (8.33 MHz) and TCLK (20 MHz)

12.0 A.C. SPECIFICATIONS

12.1 A.C. Characteristics Tables

The A.C. specifications given in the following tables consist of output delays/float times and input setup and hold times.

T_{CASE} = 0°C to 70°C, V_{CC} = 5V ±5%, $T_{AMBIENT}$ = 0°C to 55°C

BCLK Timing

Symbol	Parameter	Min	Max	Units	Notes
t1	Period	120	2500	ns	Typical = 125 ns
t2	High Time	50		ns	Measured @ 2.0V
t3	Low Time	50		ns	Measured @ 0.8V
t4	Rise Time		10	ns	(13)
t5	Fall Time		10	ns	(13)

Reset Timing

Symbol	Parameter	Min	Max	Units	Notes
t6	Pulse Width	8 (t1)		ns	

Master Timing

Symbol	Parameter	Min	Max	Units	Notes
	MREQ#				
t7	Delay ACT/Inact		33	ns	From BCLK Falling
	MAK#				
t8	Setup Time	10		ns	To BCLK Falling
t9	Hold Time	25		ns	From BCLK Falling
	IADS<9:2>, M/IO, W/R				
t10	Delay Valid	2	45	ns	From BCLK Falling(17)
	IADS<9:2>, M/IO, W/R				
t10a	Delay Valid		75	ns	From BCLK Rising(18)
t11	Delay Float		40	ns	From BCLK Falling(7)
	IBE#<3:0>				
t12	Delay Valid	2	45	ns	From BCLK Falling
t13	Delay Float		40	ns	From BCLK Falling(7, 8)
	START#				
t14	Delay Act/Inact		25	ns	From BCLK Rising
t15	Delay Float		40	ns	From BCLK Falling(7, 8)
	EX32#				
t16	Setup Time	15		ns	To BCLK Rising(9)
t17	Hold Time	50		ns	From BCLK Rising(9)
	EXRDY				
t18	Setup Time	15		ns	To BCLK Falling
t19	Hold Time	2		ns	From BCLK Falling
	IDAT<31:0>				
t20	Delay Valid	3	27	ns	From BCLK Falling(1)
t21	Delay Float		25	ns	From BCLK Falling(7, 8)
t22	Setup Time	4		ns	To BCLK Rising(2)
t23	Hold Time	6		ns	From BCLK Rising(2)
	IDAT<31:10>				
t24	Delay Valid		45	ns	From BCLK Falling(10)
	LOCK#				
t25	Delay Act/Inact	2	60	ns	From BCLK Rising
t26	Delay Float		40	ns	From BCLK Falling(7)
	IDOE#				
t27	Delay Act/Inact		25	ns	From BCLK Falling
	UALOE#				
t28a	Delay Active		60	ns	From BCLK Rising
t28b	Delay Inactive		35	ns	From BCLK Falling
	IDDIR				
t29	Delay Act/Inact		40	ns	From BCLK Falling

1

Master Timing (Burst)

Symbol	Parameter	Min	Max	Units	Notes
	MSBURST#				
t30	Delay ACT/INACT		35	ns	From BCLK Falling
t31	Delay Float		40	ns	From BCLK Rising(8)
	START#, IBE# <3:0>				
t31a	Delay Float		40	ns	From BCLK Rising(19)
	SLBURST#				
t32	Setup Time	15		ns	To BCLK Rising
t33	Hold Time	50		ns	From BCLK Rising
	IDOE#				
t34	Delay Act/Inact		25	ns	From BCLK Rising
	IDAT <31:0>				
t35	Setup Time (Read)	5		ns	To BCLK Rising(2)
t36	Hold Time (Read)	6		ns	From BCLK Rising(2)
t37	Delay Valid	3	27	ns	From BCLK Rising(1)
t38	Delay Invalid	0		ns	From BCLK Rising(1)
	MASTER16#				
t39	Delay Act		50	ns	From BCLK Rising
t40	Delay Float	1	40	ns	From BCLK Rising(7,8)

Slave Timing

Symbol	Parameter	Min	Max	Units	Notes
	IADS<11:12>, M/IO				
t41	Setup Time	120		ns	To CMD# Falling
t42	Hold Time	25		ns	From CMD# Falling
	EX32#				
t43	Delay Act/Float		54	ns	From IADS<11:2>, M/IO
t44	Delay Act/Float		34	ns	From AEN
	AEN				
t45	Setup Time	95		ns	To CMD# Falling
t46	Hold Time	25		ns	From CMD# Falling
	START#				
t47	Pulse Width	110		ns	
	IBE# <3:0>, W/R				
t48	Setup Time	80		ns	To CMD# Falling
t49	Hold Time	25		ns	From CMD# Falling
	EXRDY				
t50	Delay Negated		124	ns	From START# Falling(3)
t51	Delay Float	1	40	ns	From BCLK Falling
	CMD#				
t52	Pulse Width	110		ns	
	IDAT<31:0>				
t53	Setup Time	−35		ns	To CMD# Falling(2)
t54	Hold Time	0		ns	From CMD# Rising(2)
t55	Delay Valid		100	ns	From BCLK Rising(1)
t56	Delay Invalid	0		ns	From CMD# Rising(1)
t57	Delay Float		50	ns	From CMD# Rising
	IDDIR				
t58	Delay Valid		50	ns	From W/R Valid
t59	Delay Invalid	2		ns	From CMD# Rising
	IDOE#				
t60	Delay Act (Read)		25	ns	From CMD# Falling
t61	Delay Inact (Read)		20	ns	From CMD# Rising
t62	Delay Act/Inact (Write)		45	ns	From BCLK Rising
	IOSEL#				
t63	Delay Active		60	ns	From IADS<11:2>
t64	Delay Inactive	5		ns	From CMD# Rising If Latched

Transfer Buffer Interface Timing

Symbol	Parameter	Min	Max	Units	Notes
	TCLK				
t65	Period	50	250	ns	
t66	High Time	18		ns	Measured @ 2.0V
t67	Low Time	20		ns	Measured @ 0.8V
	TRQ#				
t68	Delay Act/Inact		15	ns	From TCLK Rising
	TLD#				
t69	Delay Act/Inact		25	ns	From TCLK Rising
	TEOP#				
t70	Delay Act/Float		25	ns	From TCLK Rising
	TCHAN, TDIR				
t73	Setup Time	25		ns	To TLD# or TRQ# Active(11)
	TACK#				
t74	Setup Time	15		ns	To TCLK Rising
t75	Hold Time	1		ns	To TCLK Rising
	TDAT<15:0>				
t76	Delay Valid	4	25	ns	From TCLK Rising/TDOE# Falling
t77	Delay Float		25	ns	From TCLK/TDOE# Rising
t78	Setup Time	10		ns	To TCLK Rising
t79	Hold Time	1		ns	From TCLK Rising
t80	Ratio of TCLK to BCLK	1.1			

Local Processor Interface Timing (Read Cycle)

Symbol	Parameter	Min	Max	Units	Notes
	LADS<1:0>, LCS#				
t81	Setup Time	10		ns	To LRD# Falling
t82	Hold Time	0		ns	From LRD# Rising
	LRD#				
t83	Pulse Width	150		ns	
	LDAT<7:0>				
t84	Delay Valid		130	ns	From LRD# Falling(4)
t85	Max Delay Valid		2.5 (t1) + 120	ns	From LRD# Falling(5)
t86	Delay Float		40	ns	From LRD# Rising
t87	LRD# (Inact) to LRD# (Act) or LWR# (Act) Recovery Time	60		ns	

Local Processor Interface (Write Cycle)

Symbol	Parameter	Min	Max	Units	Notes
	LADS<1:0>, LCS#				
t88	Setup Time	10		ns	To LWR# Falling
t89	Hold Time	0		ns	From LWR# Rising
	LWR#				
t90	Pulse Width	100		ns	(4)
	LDAT<7:0>				
t91	Setup Time	60		ns	To LWR# Rising(4)
t92	Hold Time	10		ns	From LWR# Rising
t93	Data Valid		70	ns	From LWR# Falling(5)
	LWR# (Inact) to LWR# (Act) or LRD# (Act)				
t94	Recovery Time	60		ns	

Local Processor Ready Timing

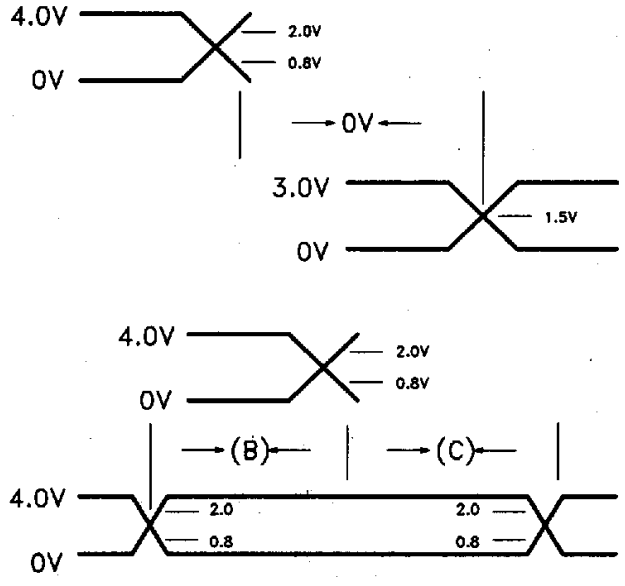
Symbol	Parameter	Min	Max	Units	Notes
	LRDY				
t95	Delay Inactive		50	ns	From LADS and LCS# Valid(5)
	Delay Active				From LRD# or LWR# Active
t96	Max Delay		3.5 (t1)+60	ns	(5, 6)
t97	Max Delay		2.5 (t1)+60	ns	(5, 6)
t98	Min Delay	1.5 (t1)		ns	(5, 6)
	LDAT<7:0>				
t99	Delay Valid		0	ns	From LRDY Rising(5, 12)

NOTES FOR A.C TIMINGS:

1. Specification does not include allowance for 13 ns max. and 2 ns min. into 240 pF for external buffer delay to EISA bus.
2. Specification does not include allowance for 8 ns max. and 1 ns min. into 25 pF for external input delay from EISA bus.
3. Delay includes 40 ns for pull-up rise time (300Ω into 240 pF, 2V rise).
4. Applies to all non-shared registers excluding the Peek/Poke Data registers. LRDY will remain active.
5. Applies to the Peek/Poke Data and Shared Registers. LRDY will be taken inactive as soon as LA<1:0> and LCS# are valid, and remain inactive until valid data is available, or has been written. The deassertion of the local read strobe (LRD#) or local write strobe (LWR#) indicates the end of the current shared register or peek/poke data register access. If the local chip select (LCS#) input remains asserted and the local address selects remain low (LADS<1:0>) after LRD# or LWR# deasserts, a new shared register or peek/poke data register cycle begins. Under these conditions, the LRDY output will become inactive again (driven low) within the time specified by t95.
6. The maximum LRDY delay, 3.5 (t1) + 60 ns from LRD# or LWR#, only occurs if the local processor access loses the internal register access arbitration to an EISA access and if the following BCLK cycle is stretched. Without BCLK stretching, the maximum delay is 2.5 (t1) +60 ns. The minimum LRDY delay is 1.5 (t1). **NOTE:** The maximum BCLK stretch that will be seen by the BMIC is one BCLK period; this is assuming that the bus controller is the 82358 (EBC). If the 82358 is not used as the bus controller, the LRDY and data delay max. specs (t96/t85) will not necessarily be valid.
7. Exiting master mode, the address lines <31:2>, M/IO, LOCK# START#, IBE#<3:0>, MSBURST#, IDAT<31:0>, and W/R will float no later than the falling edge of BCLK after CMD# is deasserted.
8. During a mismatched cycle START#, IBE#<3:0>, and IDAT<31:0> will float from the first falling edge of BCLK after START# is negated.
9. Includes mismatched cycles.
10. Refers to the upper 22 EISA address lines which are multiplexed into the upper 22 data lines IDAT<31:10>. The address will be available for latching into the external address latches 45 ns from the falling edge of BCLK.
11. The TDIR and TCHAN signals are referenced to the falling edge of TRQ# during the cycles that TLD# is not requested.
12. LRDY going active will always be delayed from data valid. The maximum delay seen will be no greater than one (t1) period.
13. Characterized, not tested.
14. Under non-preempt, MREQ# will deassert a minimum of 0.5 BCLKs after the negating edge of the last CMD# of the transfer, depending on the cycle type (refer to the Basic Function Timings, Section 10.0).
15. During an EISA read transfer, the BMIC will assert TEOP# typically eight TCLKs after CMD# is deasserted from the last EISA cycle, indicating end of transfer (refer to the Basic Function Timings, Section 10.0).
16. During an EISA write transfer, the BMIC will assert TEOP# two TCLKs after CMD# is deasserted, indicating end of transfer (refer to the Basic Function Timings, Section 10.0).
17. For address changes while CMD# is active.
18. During an upper address load cycle, at the beginning of a transfer sequence, CMD# is inactive.
19. For "Downshifting Cases" where the transfer is misaligned.

12.2 A.C. Characteristics Waveforms

A.C. Testing Input, Output Waveforms



290255-12

NOTE:
The input waveforms have tr < 2.0 ns from 0.8V to 2.0V

A. Output delay specification referenced from one of the following signals: BCLK, TCLK, CMD#, START#, AEN, IADS<11:2>, W/R, TDOE#, LRD#, LWR#, LADS<1:0>, LCS#, LRD#, or LWR#.

B. Minimum input setup specification referenced to one of the following signals: BCLK, TCLK, CMD#, LWR#, LRD#, TLD#, or TRQ#.

C. Minimum input hold specification referenced to one of the following signals: BCLK, TCLK, CMD#, LWR#, LRD#, TLD#, or TRQ#.

A.C. Testing: All inputs are driven at 4V for a logic "1" and 0V for a logic "0". A.C. Timings are measured from the 0.8V and 2.0V levels on the source signal to either the 0.8V and 2V or 1.5V level on the signal under test; except as noted by the following:

1. BCLK and TCLK high time measurements are made at 2.0V
2. BCLK and TCLK low time measurements are made at 0.8V
3. START#, CMD#, LRD#, and LWR# pulse width measurements are made at 0.8V

A.C. TEST LOADS

CL = 25 pF on IDAT<31:0>, IDOE#, IOSEL#<1:0>, TRQ#, TLD#, TEOP#, TDAT<15:0>, TCHAN, TDIR, LRDY, and LDAT<7:0>

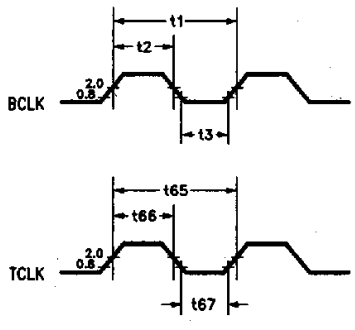
CL = 35 pF on IDDIR

CL = 50 pF on UALOE# and LINT

CL = 120 pF on MREQ# and EINT

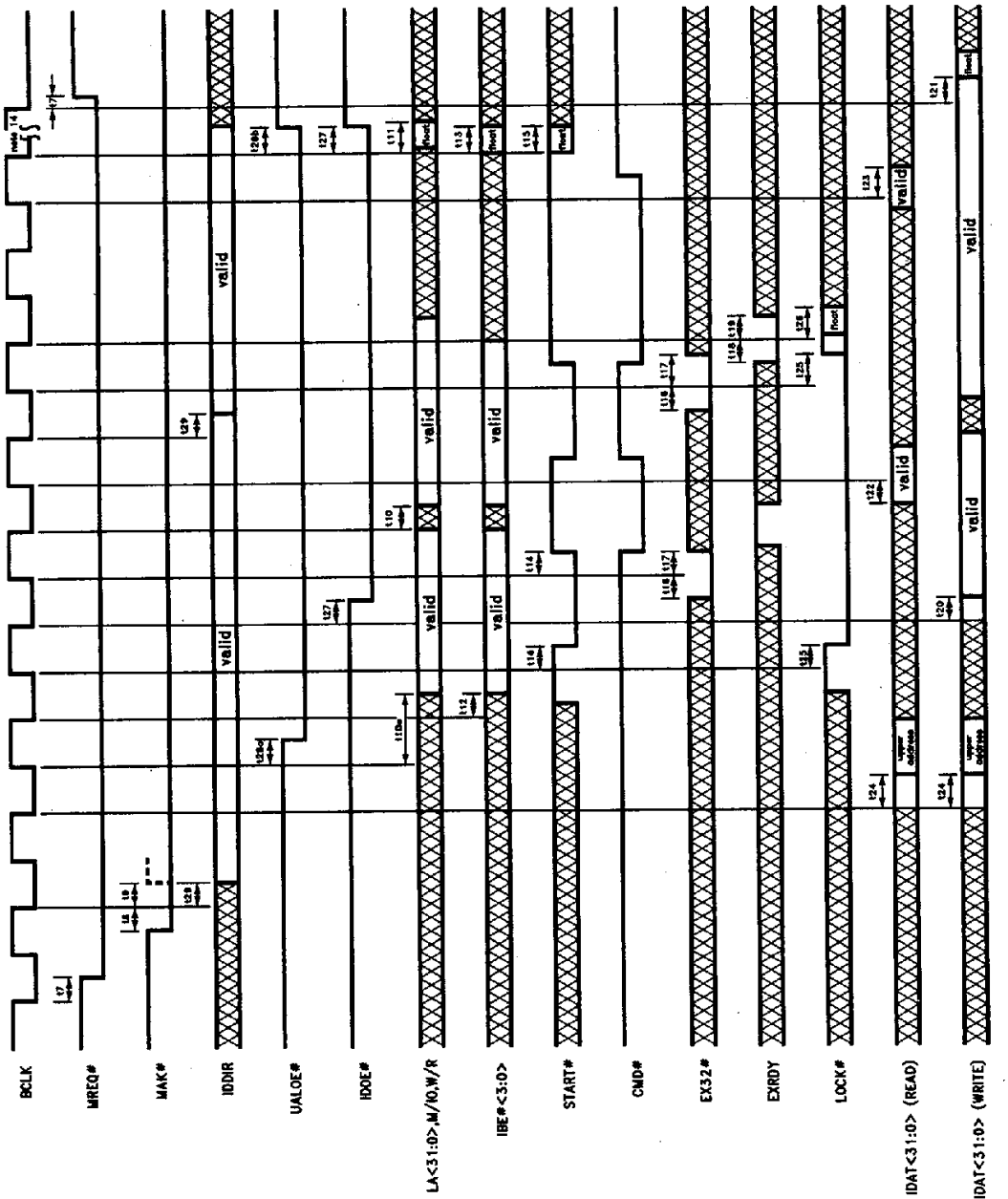
CL = 240 pF on IADS<9:2>, BE#<3:0>, W/R, START#, EX32#, LOCK, MSBURST#, MASTER16#, EXRDY, and M/IO

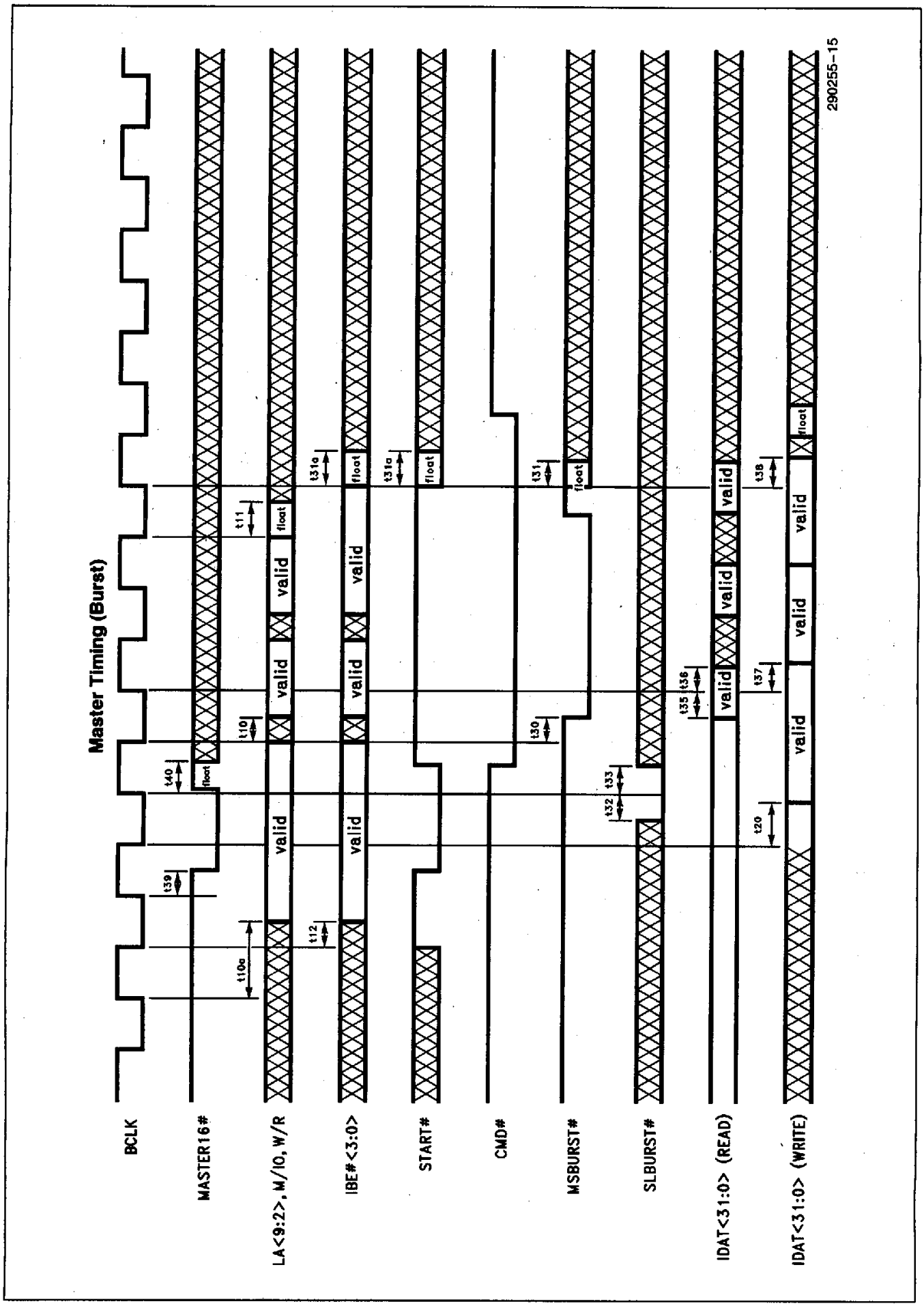
BCLK, TCLK Timing

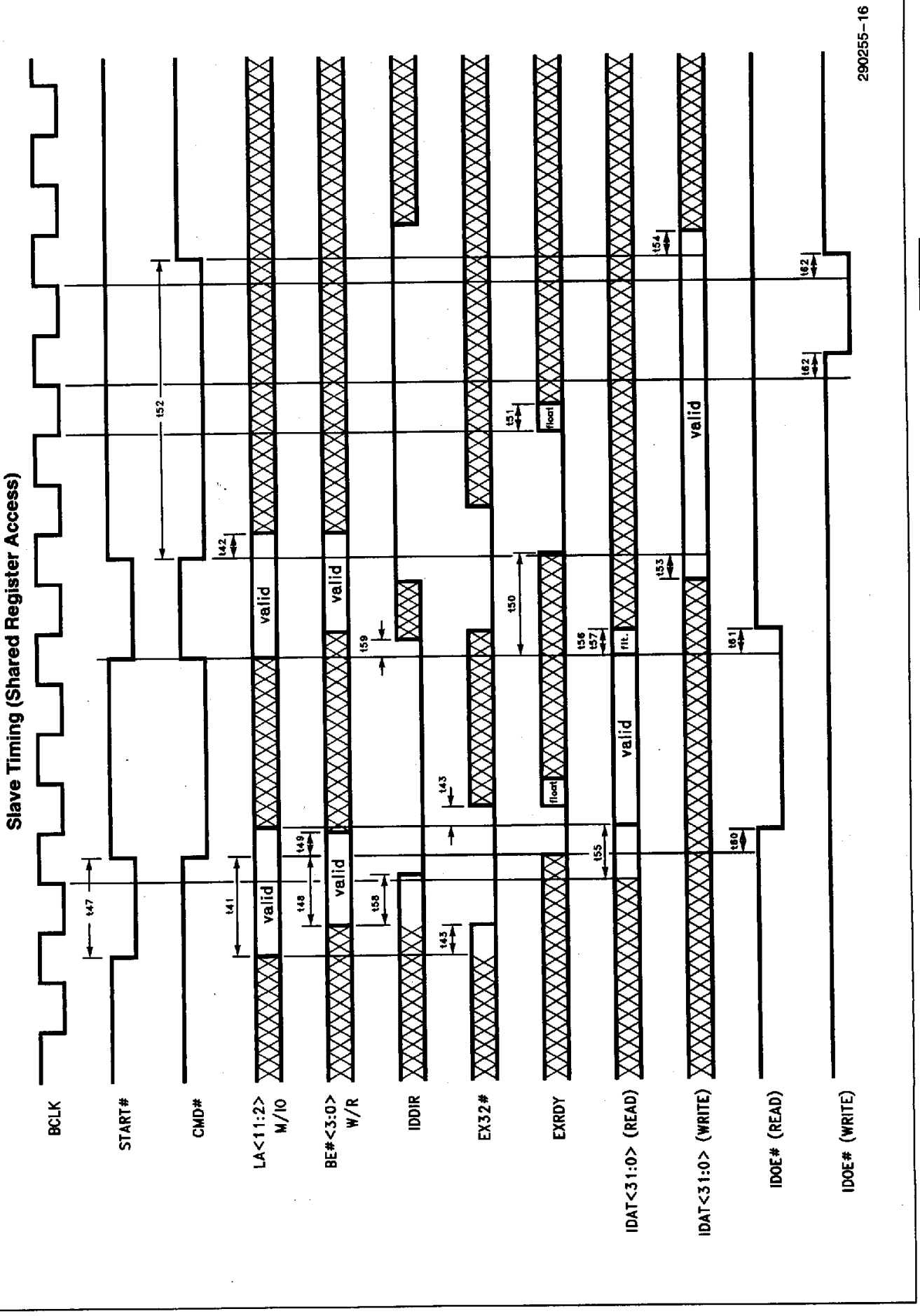


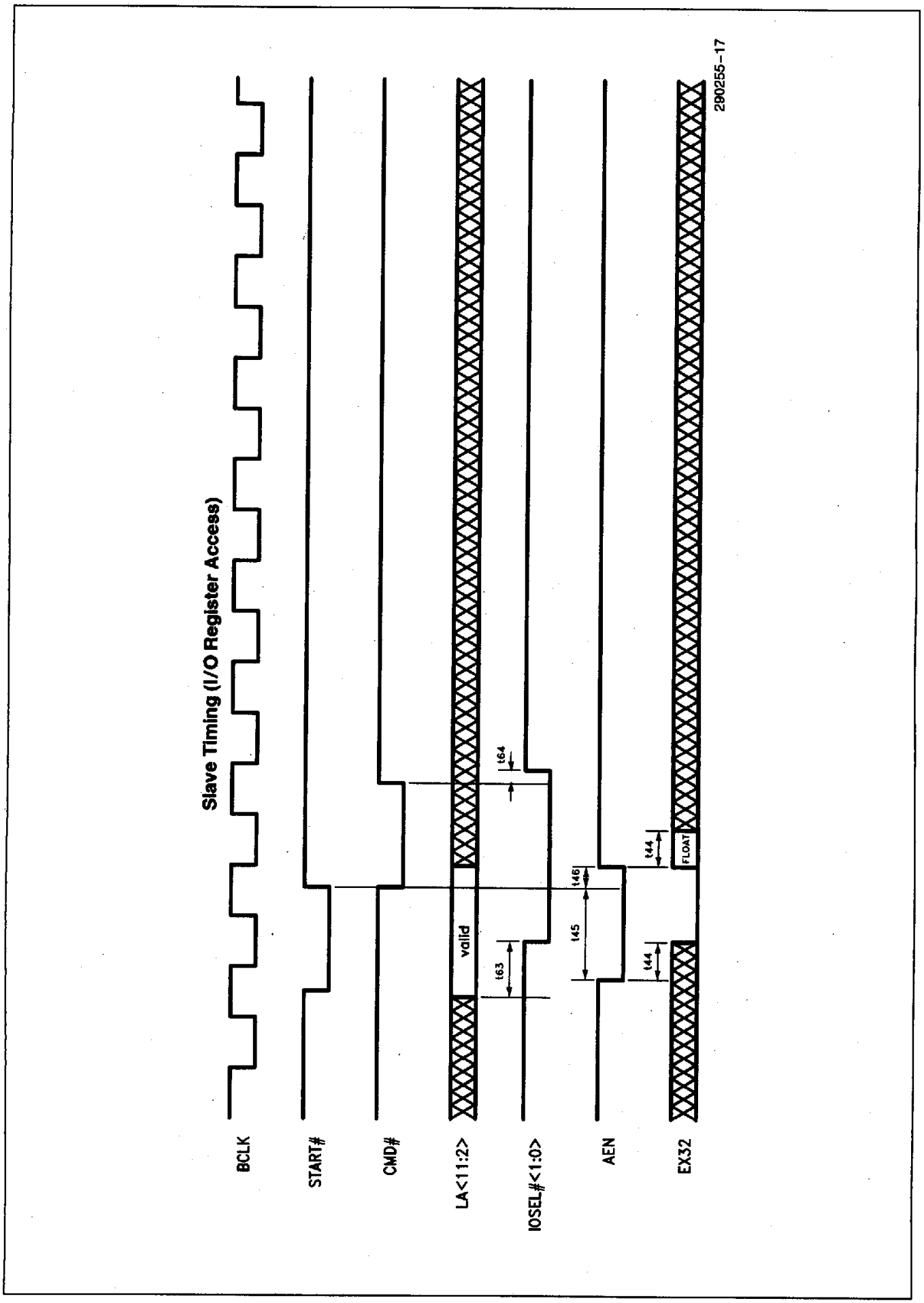
290255-13

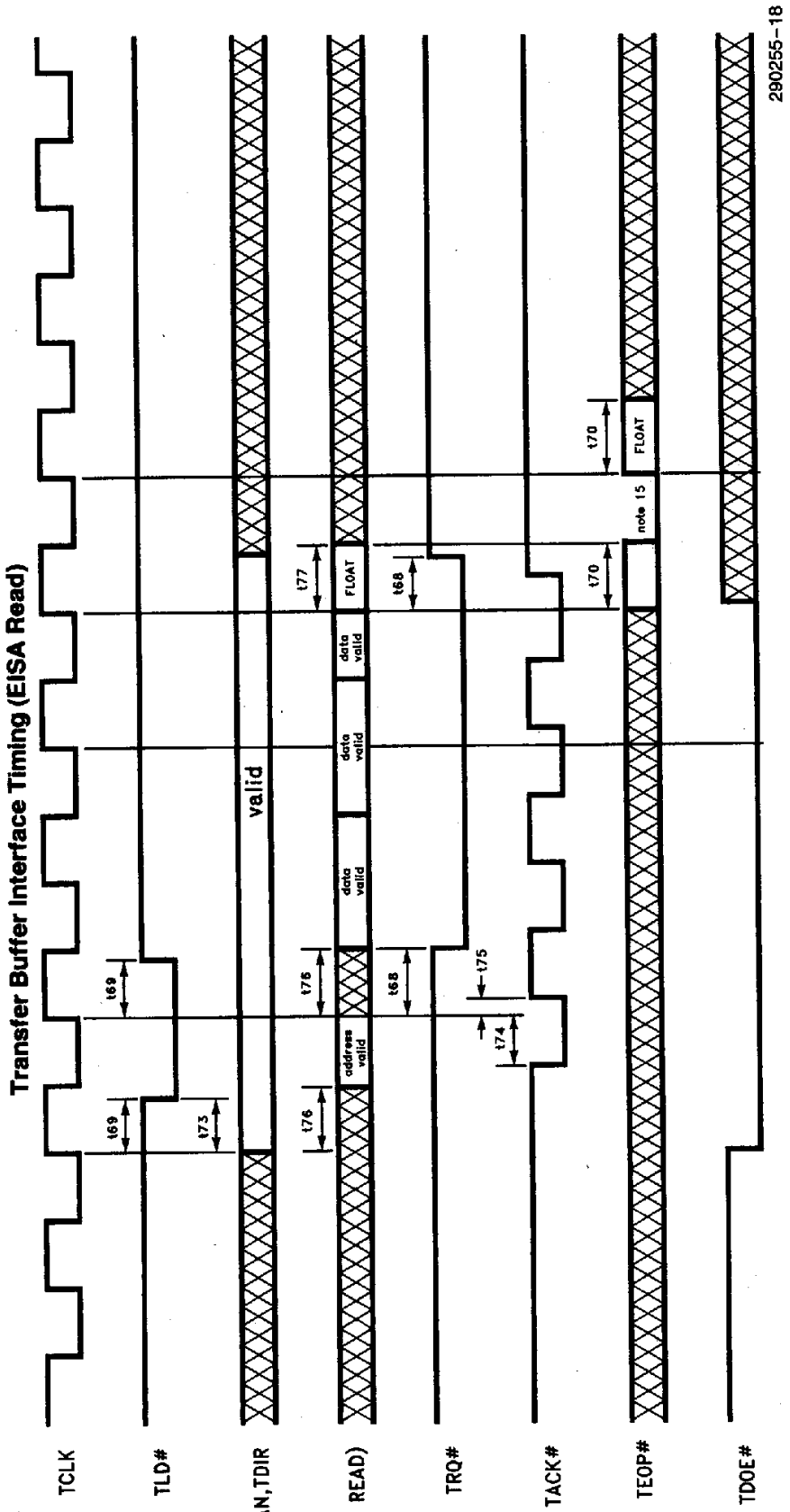
Master Timing
(Includes: all Cycle Types—Initial Burst, Non-Burst, Peek/Poke, and Mismatched)

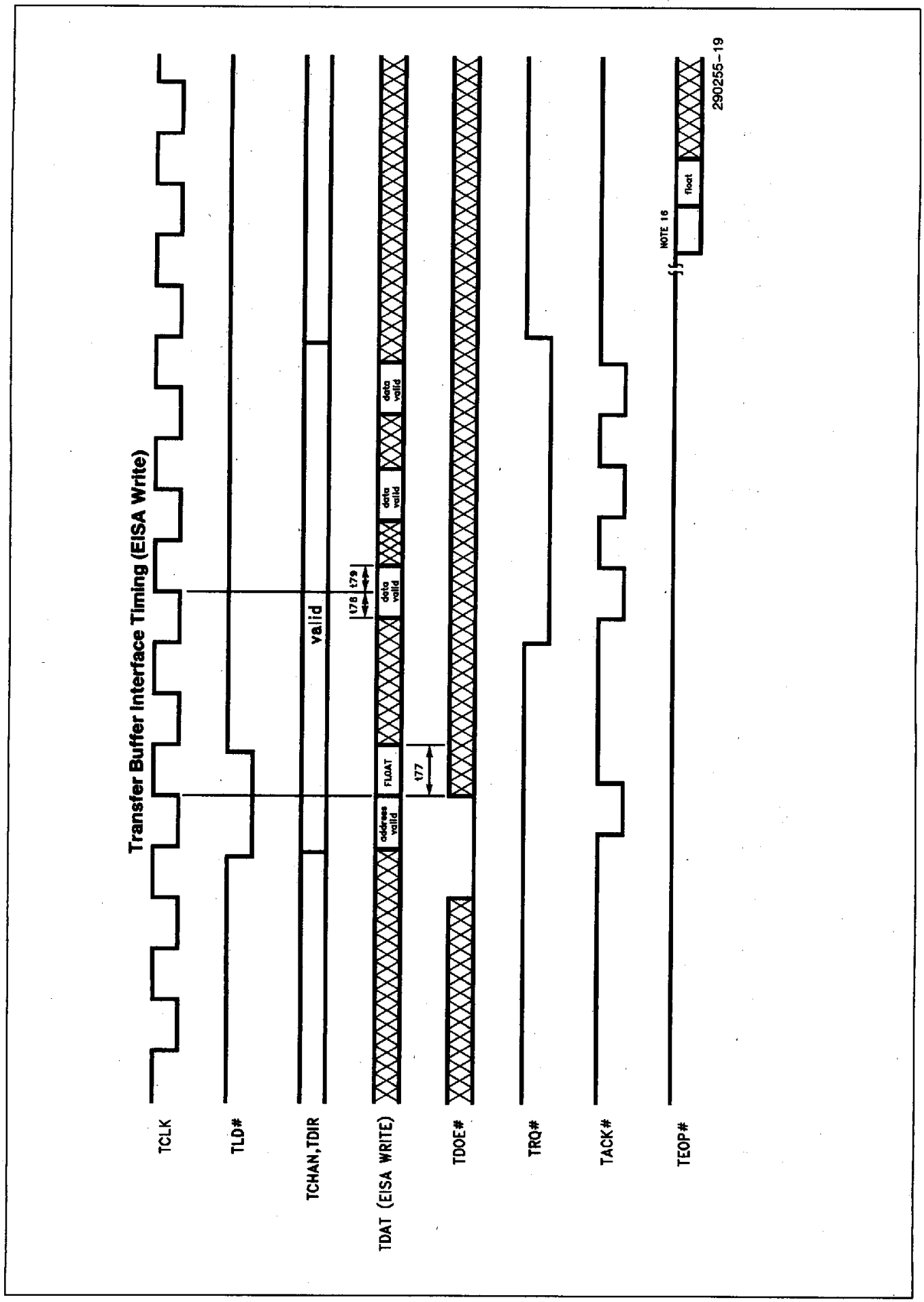




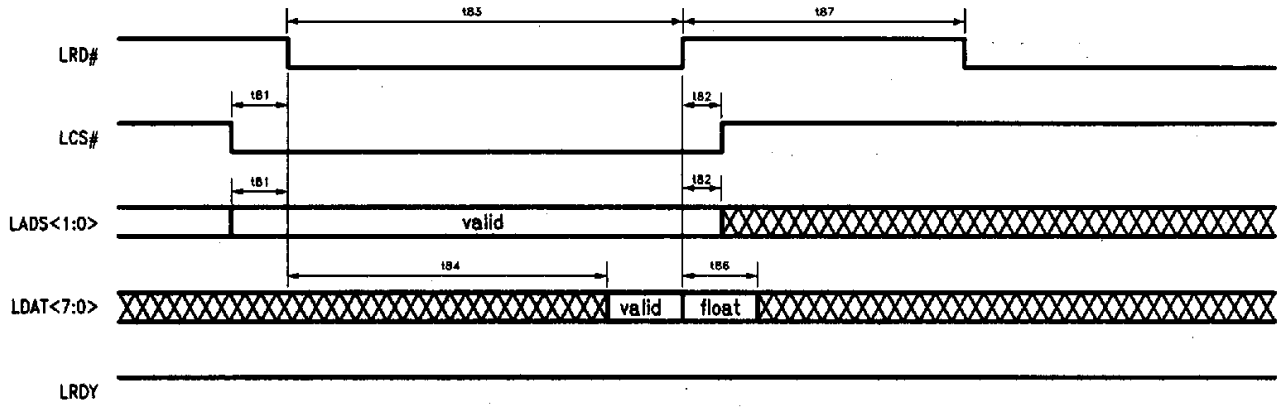






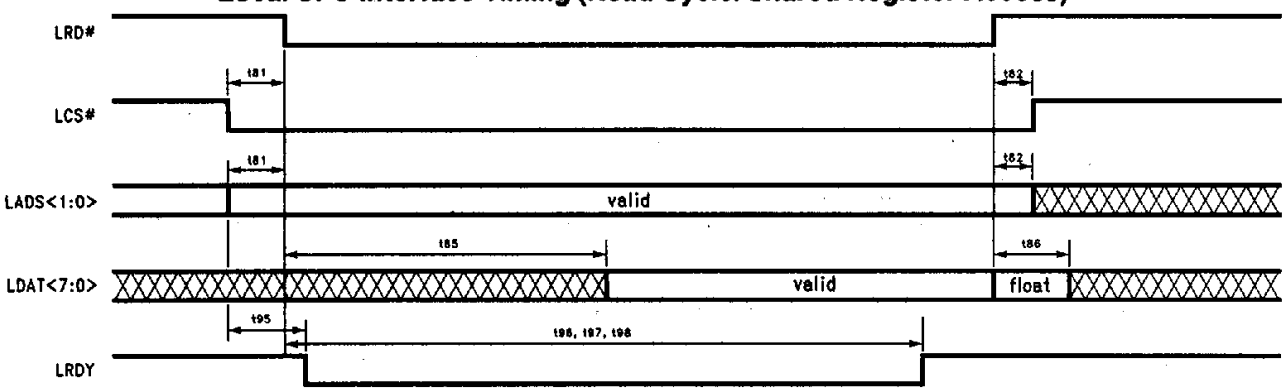


Local CPU Interface Timing (Read Cycle/Non-Shared Register Access)



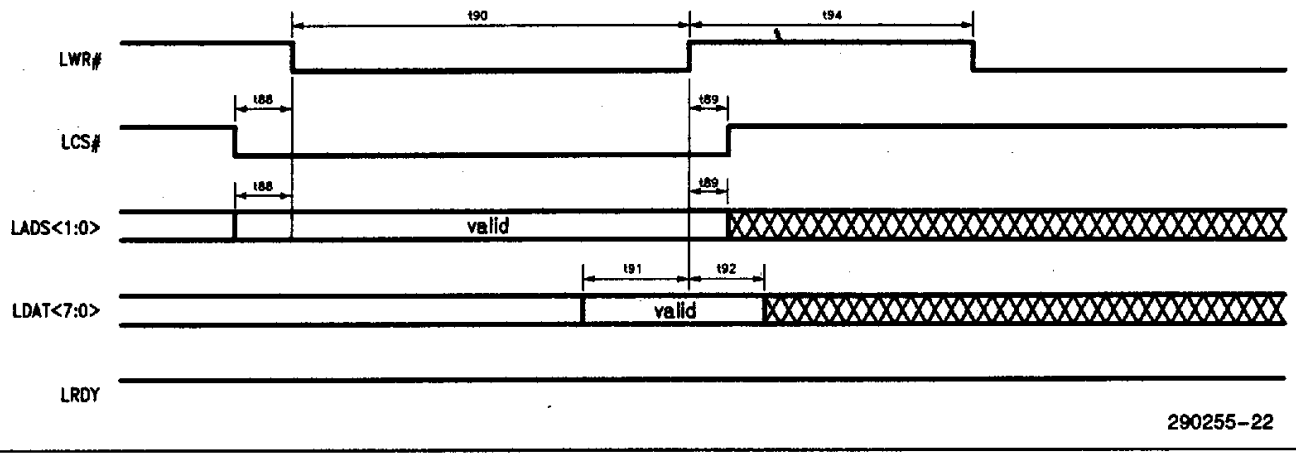
290255-20

Local CPU Interface Timing (Read Cycle/Shared Register Access)

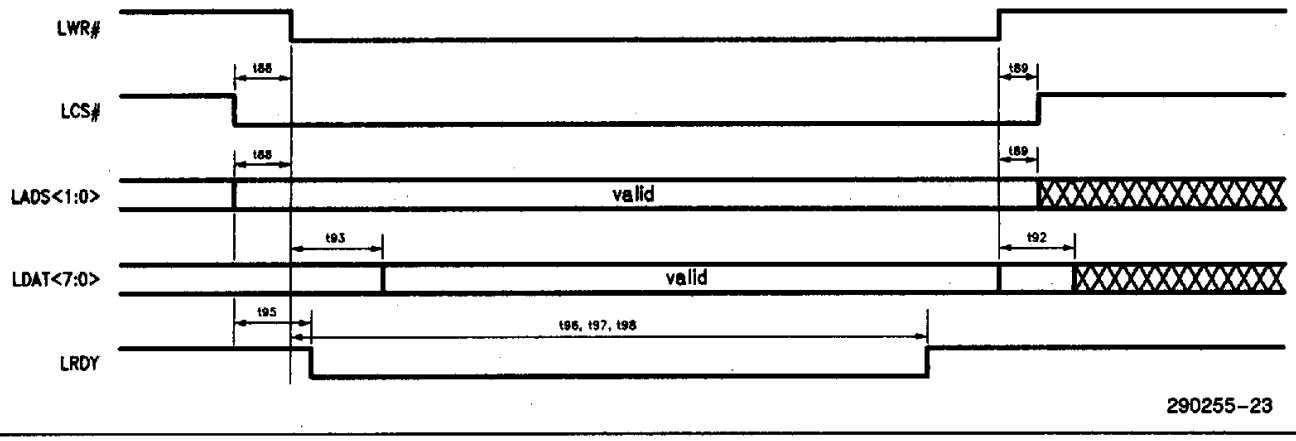


290255-21

Local CPU Interface Timing (Write Cycle/Non-Shared Register Access)



Local CPU Interface Timing (Write Cycle/Shared Register Access)



13.0 BMIC PN AND PACKAGE INFORMATION

13.1 Signal Overview

Name = Pin Name, Type = I—Input, O—Output, OC—Open Collector, B—Both Input and Output, BC—Both and Open Collector, Pin = Pin Location

Name	Type	Pin	Description
EISA BUS INTERFACE SIGNALS			
START#	B	84	EISA Start of Cycle
CMD#	I	102	EISA Command Strobe
M/IO	B	81	EISA Memory/IO Cycle Status Signal
W/R	B	80	EISA Write/Read Status Signal
EXRDY	I, OC	103	EISA Ready Signal
EX32#	I, OC	104	EISA 32-Bit Slave Response Signal
MASTER16#	OC	82	EISA 16-Bit Master Control Signal
IBE# <3:0>	B	64, 61, 60 59	EISA Byte Enable Lines
AEN	I	107	EISA Address Enable Signal
MSBURST#	O	96	EISA Master Burst Signal
SLBURST#	I	97	EISA Slave Burst Signal
LOCK#	O	98	EISA Resource Lock Signal
MREQ#	O	99	EISA Bus Master Request Signal
MAK#	I	100	EISA Master Bus Acknowledge Signal
EINT	OC	109	EISA Interrupt Request Signal
BCLK	I	101	EISA Bus Clock
RESET	I	125	EISA Reset Signal
IDAT<31:0>	B	Section	EISA Data Lines
IADS<11:10>	I	105, 106	EISA Address Input Lines
IADS<9:2>	B	57–55, 53, 44, 40–38	EISA Lower Address Lines
EISA BUFFER CONTROL SIGNALS			
UALOE#	O	78	EISA Upper Address Latch and Output Enable
IDDIR	O	79	EISA Data Buffer Direction Signal
IDOE23#	O	75	EISA Data Byte Line Buffer Enable (Bytes 3, 2)
IDOE# <1:0>	O	76, 77	EISA Data Byte Line Buffer Enables (Bytes 1, 0)
TRANSFER BUFFER INTERFACE SIGNALS			
TCLK	I	32	Transfer Clock
TRQ#	O	7	Transfer Data Request Signal
TACK#	I	6	Transfer Data Acknowledge Signal
TDIR	O	3	Transfer Data Direction Signal
TCHAN	O	4	Transfer Data Channel Select Signal
TLD#	O	5	Transfer Address Counter Load Signal
TDOE#	I	2	Transfer Data Bus Output Enable
TEOP#	I, OC	1	Transfer End-of-Process
TDAT<15:0>	B	Section	Transfer Data Bus Lines

13.1 Signal Overview (Continued)

Name = Pin Name, Type = I—Input, O—Output, OC—Open Collector, B—Both Input and Output, BC—Both and Open Collector, Pin = Pin Location

Name	Type	Pin	Description
LOCAL PROCESSOR INTERFACE SIGNALS			
LRD#	I	130	Local Read Signal
LWR#	I	129	Local Write Signal
LCS#	I	128	Local Chip Select Signal
LDAT<7:0>	B	121–118 115–112	Local Data Bus Lines
LADS<1:0>	I	127, 126	Local Address Register Select Signals
LRDY#	B	122	Local Ready Signal
LINT#	O	123	Local Processor Interrupt Signal
MISCELLANEOUS SIGNALS			
IOSEL#<1:0>	O	111, 110	Expansion Board Address Range Decode Signals
POWER PINS			
V_{CC}		108, 124	Power Pins for the Internal Logic
V_{SS}		42, 58	Ground Pins for the Internal Logic
V_{CCB}		12, 23, 41, 63, 74, 83, 94, 117, 132	Power Pins for the Output Buffers
V_{SSB}		13, 22, 33, 43, 54, 62, 73, 85, 95, 116, 131	Ground Pins for the Output Buffers

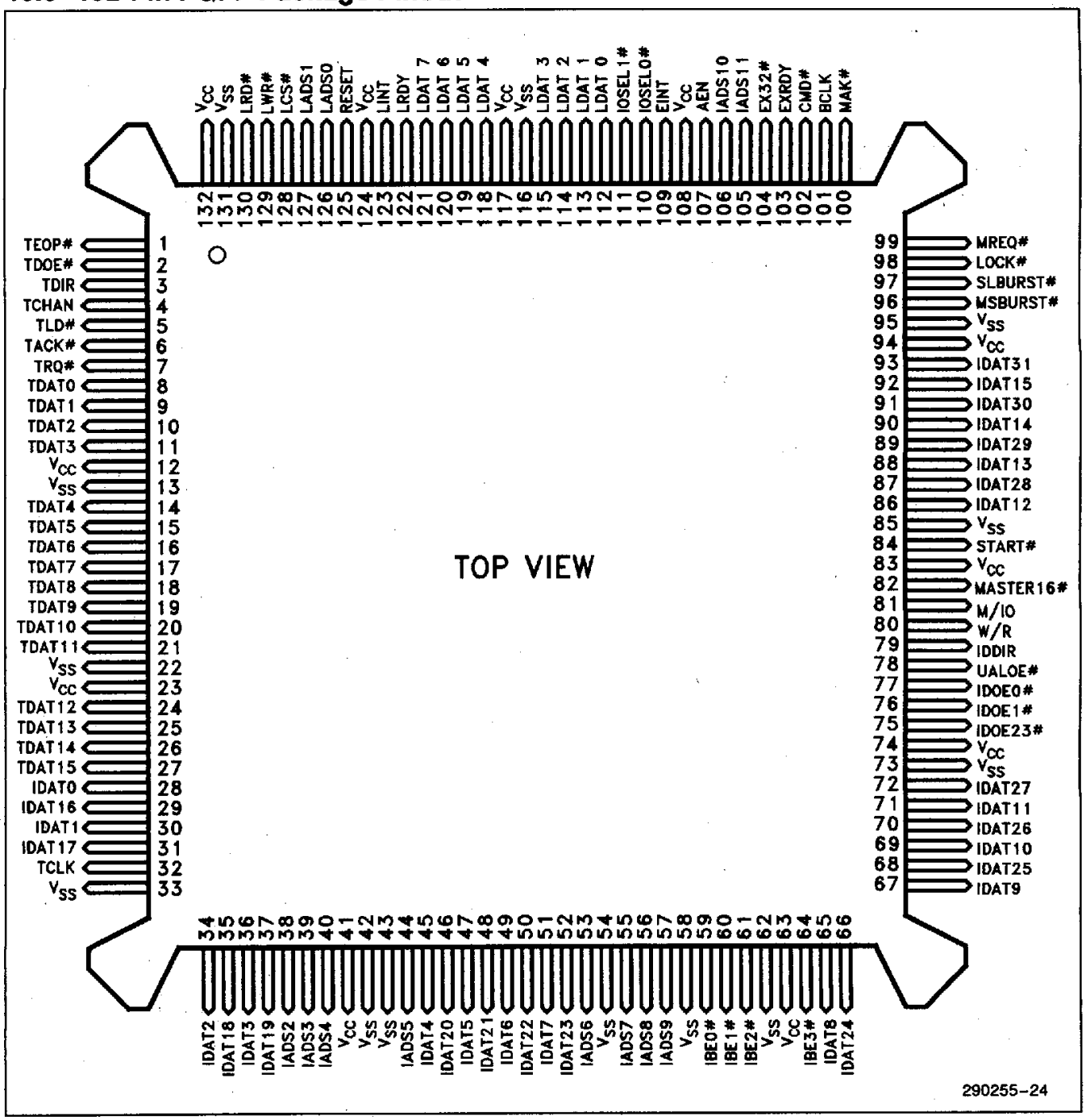
13.2 Device Pinout

I = Input, O = Output, OC = Open Collector, B = Both Input and Output, BC = Both and Open Collector

Device Pinout—132 Lead PQFP

A Row			B Row			C Row			D Row		
Pin	Label	Type	Pin	Label	Type	Pin	Label	Type	Pin	Label	Type
1	TEOP#	I, OC	34	IDAT2	B	67	IDAT9	B	100	MAK#	I
2	TDOE#	I	35	IDAT18	B	68	IDAT25	B	101	BCLK	I
3	TDIR	O	36	IDAT3	B	69	IDAT10	B	102	CMD#	I
4	TCHAN	O	37	IDAT19	B	70	IDAT26	B	103	EXRDY	I, OC
5	TLD#	O	38	IADS2	B	71	IDAT11	B	104	EX32#	I, OC
6	TACK#	I	39	IADS3	B	72	IDAT27	B	105	IADS11	I
7	TRQ#	O	40	IADS4	B	73	V_{SSB}		106	IADS10	I
8	TDAT0	B	41	V_{CCB}		74	V_{CCB}		107	AEN	I
9	TDAT1	B	42	V_{SS}		75	IDOE23#	O	108	V_{CC}	
10	TDAT2	B	43	V_{SSB}		76	IDOE1#	O	109	EINT	OC
11	TDAT3	B	44	IADS5	B	77	IDOE0#	O	110	IOSEL0#	O
12	V_{CCB}		45	IDAT4	B	78	UALOE#	O	111	IOSEL1#	O
13	V_{SSB}		46	IDAT20	B	79	IDDIR	O	112	LDAT0	B
14	TDAT4	B	47	IDAT5	B	80	W/R	B	113	LDAT1	B
15	TDAT5	B	48	IDAT21	B	81	M/IO	B	114	LDAT2	B
16	TDAT6	B	49	IDAT6	B	82	MASTER16#	OC	115	LDAT3	B
17	TDAT7	B	50	IDAT22	B	83	V_{CCB}		116	V_{SSB}	
18	TDAT8	B	51	IDAT7	B	84	START#	B	117	V_{CCB}	
19	TDAT9	B	52	IDAT23	B	85	V_{SSB}		118	LDAT4	B
20	TDAT10	B	53	IADS6	B	86	IDAT12	B	119	LDAT5	B
21	TDAT11	B	54	V_{SSB}		87	IDAT28	B	120	LDAT6	B
22	V_{SSB}		55	IADS7	B	88	IDAT13	B	121	LDAT7	B
23	V_{CCB}		56	IADS8	B	89	IDAT29	B	122	LRDY	B
24	TDAT12	B	57	IADS9	B	90	IDAT14	B	123	LINT	O
25	TDAT13	B	58	V_{SS}		91	IDAT30	B	124	V_{CC}	
26	TDAT14	B	59	IBE0#	B	92	IDAT15	B	125	RESET	I
27	TDAT15	B	60	IBE1#	B	93	IDAT31	B	126	LADS0	I
28	IDAT0	B	61	IBE2#	B	94	V_{CCB}		127	LADS1	I
29	IDAT16	B	62	V_{SSB}		95	V_{SSB}		128	LCS#	I
30	IDAT1	B	63	V_{CCB}		96	MSBURST#	O	129	LWR#	I
31	IDAT17	B	64	IBE3#	B	97	SLBURST#	I	130	LRD#	I
32	TCLK	I	65	IDAT8	B	98	LOCK#	O	131	V_{SSB}	
33	V_{SSB}		66	IDAT24	B	99	MREQ#	O	132	V_{CCB}	

13.3 132-Pin PQFP Package Pinout



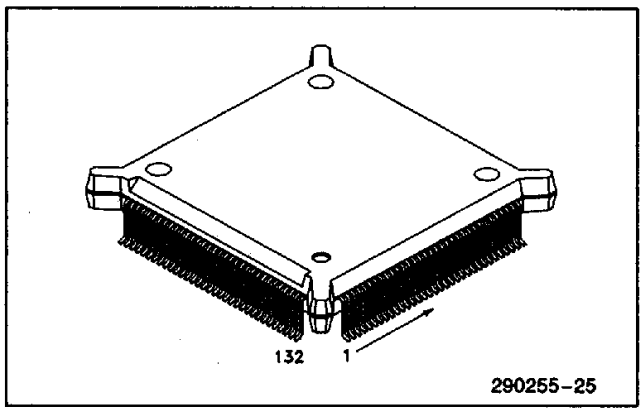
PACKAGING INFORMATION

(See Packaging Specification Order # 240800, Package Type NG)

Introduction

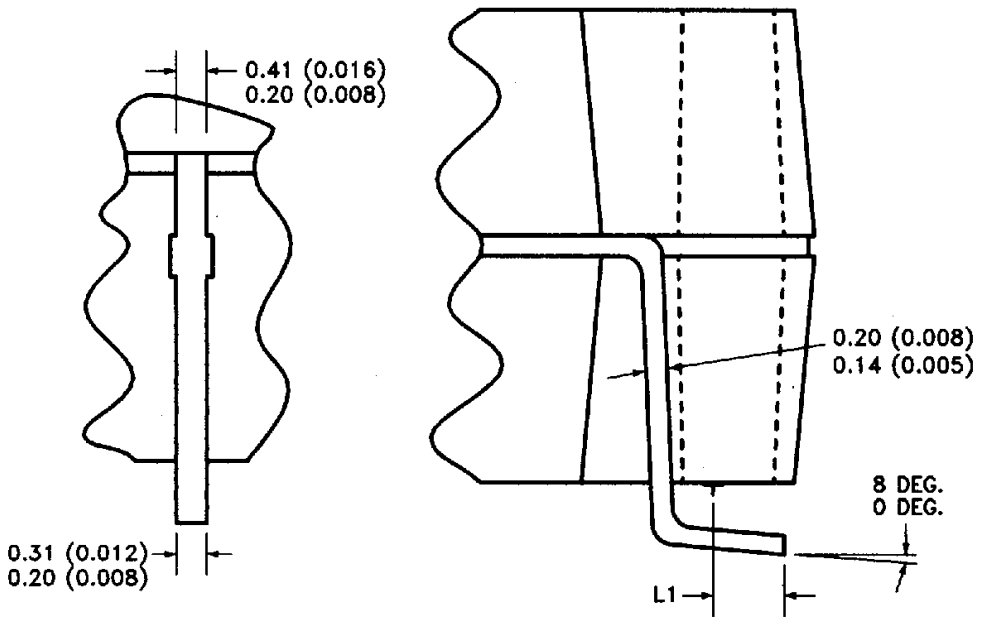
The individual components of Intel's EISA Chip Set come in JEDEC standard Gull Wing packages (25 MIL pitch), with "bumpers" on the corners for ease of handling. Please refer to the accompanying table for the package associated with each device, and to the individual component specifications for pinouts. (Note that the individual pinouts are numbered consistently with the numbering scheme depicted in the accompanying figures.)

PLASTIC QUAD FLAT PACK (PQFP)



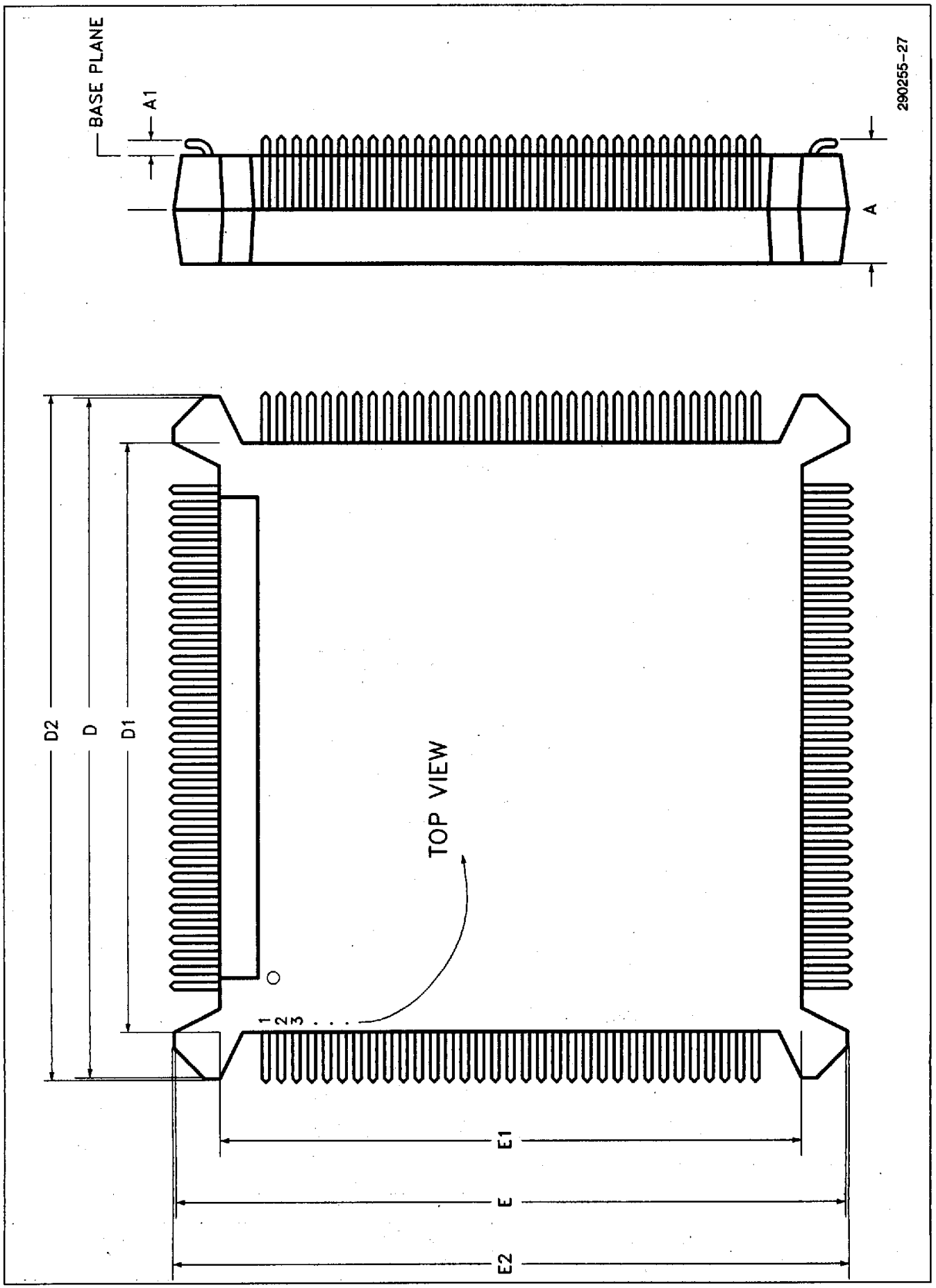
TYPICAL LEAD

Case Outline Drawings
Plastic Fine Pitch Chip Carrier
0.84 mm Pitch

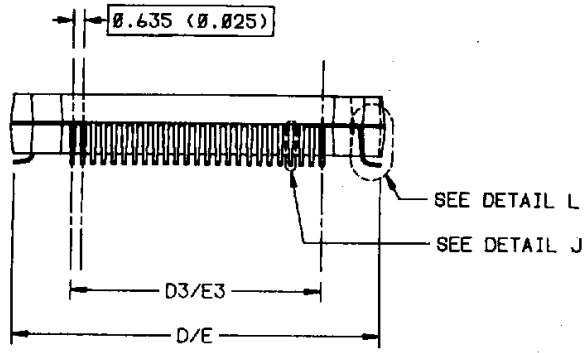


Symbol	Description	Inch		mm	
		Min	Max	Min	Max
N	Lead Count	132		132	
A	Package Height	0.160	0.170	4.06	4.32
A1	Standoff	0.020	0.030	0.51	0.76
D, E	Terminal Dimension	1.075	1.085	27.31	27.56
D1, E1	Package Body	0.947	0.953	24.05	24.21
D2, E2	Bumper Distance	1.097	1.103	27.86	28.02
D3, E3	Lead Dimension	0.800 Ref		20.32 Ref	
L1	Foot Length	0.020	0.030	0.51	0.76

13.4 PRINCIPAL DIMENSIONS & DATUMS



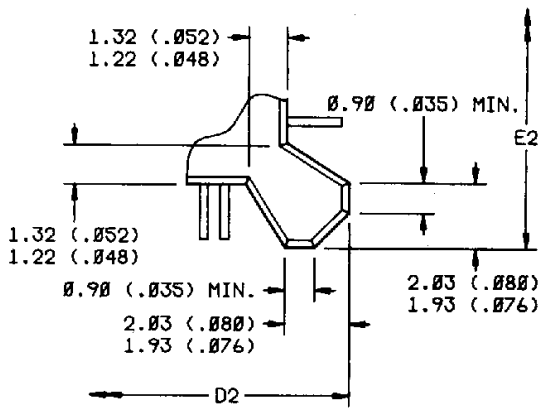
TERMINAL DETAILS



mm (inch)

290255-28

BUMPER DETAIL



mm (inch)

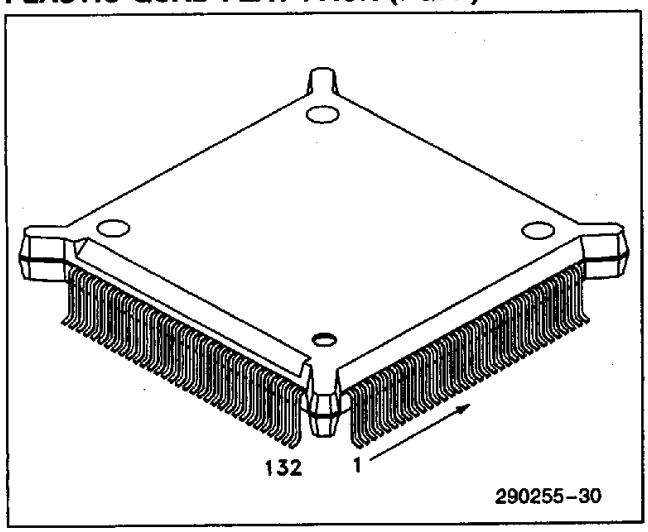
290255-29

13.5 Package Thermal Specification

The 82355 (BMIC) is specified for operation when the case temperature is within the range of 0°C–70°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 the figure below.

PLASTIC QUAD FLAT PACK (PQFP)



82355 PQFP Package Thermal Characteristics

Thermal Resistance—°C/W							
Parameter	Air Flow Rate (ft/min)						
	0	50	100	200	400	600	800
θ Junction—Case	7	7	7	7	7	7	7
θ Case to Ambient	22	21	19.5	17.5	14.5	12	10

NOTES:
1. Table applies to 82355 PQFP plugged into socket or soldered directly into board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$.

Process Name:

1.2μ CHMOS III P-well

I_{CC} at Hot with no Resistive Loads:

150 mA max at 70°C
Measure PQFP case temperature
at center of top surface

14.0 BMIC REGISTER ADDRESS MAP

14.1 Index Register Set

The following registers are mapped directly into the local processor interface:

Local Address	Type	Register Description
0	R/W	Local Data Register
1	R/W	Local Index Register
2	R/W	Local Status/Control Register
3	—	Reserved

14.2 Shared Register Set

EISA Address	Type	Index Address	Type	Register Description
XC80	R	00	R/W	ID Byte 0
XC81	R	01	R/W	ID Byte 1
XC82	R	02	R/W	ID Byte 2
XC83	R	03	R/W	ID Byte 3
XC84	—	04	—	Non BMIC Register (For Expansion Board Use)
XC85	—	05	—	Non BMIC Register (For Expansion Board Use)
XC86	—	06	—	Non BMIC Register (For Expansion Board Use)
XC87	—	07	—	Non BMIC Register (For Expansion Board Use)
XC88	R	08	R/W	Global Configuration Register
XC89	R/W	09	R	System Interrupt Enable/Control Register
XC8A	R/W	0A	R/W	Semaphore Port 0
XC8B	R/W	0B	R/W	Semaphore Port 1
XC8C	R	0C	R/W	Local Doorbell Enable Register
XC8D	R/W	0D	R/W	Local Doorbell Interrupt/Status Register
XC8E	R/W	0E	R	EISA System Doorbell Enable Register
XC8F	R/W	0F	R/W	EISA System Doorbell Interrupt/Status Register
XC90	R/W	10	R/W	Mailbox Register (1)
XC91	R/W	11	R/W	Mailbox Register (2)
XC92	R/W	12	R/W	Mailbox Register (3)
XC93	R/W	13	R/W	Mailbox Register (4)
XC94	R/W	14	R/W	Mailbox Register (5)
XC95	R/W	15	R/W	Mailbox Register (6)
XC96	R/W	16	R/W	Mailbox Register (7)
XC97	R/W	17	R/W	Mailbox Register (8)
XC98	R/W	18	R/W	Mailbox Register (9)
XC99	R/W	19	R/W	Mailbox Register (10)
XC9A	R/W	1A	R/W	Mailbox Register (11)
XC9B	R/W	1B	R/W	Mailbox Register (12)
XC9C	R/W	1C	R/W	Mailbox Register (13)
XC9D	R/W	1D	R/W	Mailbox Register (14)
XC9E	R/W	1E	R/W	Mailbox Register (15)
XC9F	R/W	1F	R/W	Mailbox Register (16)
XCA0–XCAF	R/W	20–2F	—	Reserved

14.3 Processor Only Register Set

Index Address	Type	Register Description
30	R/W	Peek/Poke Data Register Byte 0
31	R/W	Peek/Poke Data Register Byte 1
32	R/W	Peek/Poke Data Register Byte 2
33	R/W	Peek/Poke Data Register Byte 3
34	R/W	Peek/Poke Address Register Byte 0
35	R/W	Peek/Poke Address Register Byte 1
36	R/W	Peek/Poke Address Register Byte 2
37	R/W	Peek/Poke Address Register Byte 3
38	R/W	Peek/Poke Control Register
39	R/W	I/O Decode Range 0 Base Address Register
3A	R/W	I/O Decode Range 0 Control Register
3B	R/W	I/O Decode Range 1 Base Address Register
3C	R/W	I/O Decode Range 1 Control Register
3D	—	Reserved
3E	—	Reserved
3F	—	Reserved
40	R/W	Channel 0 Base Count Register Byte 0
41	R/W	Channel 0 Base Count Register Byte 1
42	R/W	Channel 0 Base Count Register Byte 2
43	R/W	Channel 0 Base Address Register Byte 0
44	R/W	Channel 0 Base Address Register Byte 1
45	R/W	Channel 0 Base Address Register Byte 2
46	R/W	Channel 0 Base Address Register Byte 3
47	—	Reserved
48	R/W	Channel 0 Configuration Register
49	W	Channel 0 Transfer Strobe Register
4A	R/W	Channel 0 Status Register
4B	R/W	Channel 0 TBI Base Address Register Byte 0
4C	R/W	Channel 0 TBI Base Address Register Byte 1
4D	—	Reserved
4E	—	Reserved
4F	—	Reserved
50	R	Channel 0 Current Count Register Byte 0
51	R	Channel 0 Current Count Register Byte 1
52	R	Channel 0 Current Count Register Byte 2
53	R	Channel 0 Current Address Register Byte 0
54	R	Channel 0 Current Address Register Byte 1
55	R	Channel 0 Current Address Register Byte 2
56	R	Channel 0 Current Address Register Byte 3
57	—	Reserved
58	R	Channel 0 TBI Current Address Register Byte 0
59	R	Channel 0 TBI Current Address Register Byte 1
5A	—	Reserved
5B	—	Reserved
5C	—	Reserved
5D	—	Reserved
5E	—	Reserved
5F	—	Reserved

14.3 Processor Only Register Set (Continued)

Index Address	Type	Register Description
60	R/W	Channel 1 Base Count Register Byte 0
61	R/W	Channel 1 Base Count Register Byte 1
62	R/W	Channel 1 Base Count Register Byte 2
63	R/W	Channel 1 Base Address Register Byte 0
64	R/W	Channel 1 Base Address Register Byte 1
65	R/W	Channel 1 Base Address Register Byte 2
66	R/W	Channel 1 Base Address Register Byte 3
67	—	Reserved
68	R/W	Channel 1 Configuration Register
69	W	Channel 1 Transfer Strobe Register
6A	R/W	Channel 1 Status Register
6B	R/W	Channel 1 TBI Base Address Register Byte 0
6C	R/W	Channel 1 TBI Base Address Register Byte 1
6D	—	Reserved
6E	—	Reserved
6F	—	Reserved
70	R	Channel 1 Current Count Register Byte 0
71	R	Channel 1 Current Count Register Byte 1
72	R	Channel 1 Current Count Register Byte 2
73	R	Channel 1 Current Address Register Byte 0
74	R	Channel 1 Current Address Register Byte 1
75	R	Channel 1 Current Address Register Byte 2
76	R	Channel 1 Current Address Register Byte 3
77	—	Reserved
78	R	Channel 1 TBI Current Address Register Byte 0
79	R	Channel 1 TBI Current Address Register Byte 1
7A	—	Reserved
7B	—	Reserved
7C	—	Reserved
7D	—	Reserved
7E	—	Reserved
7F	—	Reserved

NOTES:
1. TBI = Transfer Buffer Interface
2. X = Slot number
3. All the reserved locations, when read, will return a value of no practical use to the user.
4. The "non BMIC" register locations (XC84h–XC87h & 04h–07h) are locations to be used by registers implemented externally on the expansion board. The BMIC will not respond to these locations (XC84h–CC87h) when accessed from the EISA side. However, the BMIC can be programmed to support the decode of the EISA addresses (XC84h–XC87h) through its I/O decode register set (refer to Section 4.8). All "non BMIC" register locations (04h–07h) when read from the local side, will return a value of no practical use to the user.

82355 Revision Summary

The following changes have been made since revision 006:

Section 1.1 EISA READ definitions has been changed from "A data transfer (burst, non-burst (two BCLK), or mismatched) from system to the expansion board across one of the two transfer channels" to "A data transfer (burst, non-burst (two BCLK), or mismatched) from system to the expansion board across Channel 1 transfer channel 1."

Section 4.2 New paragraph added after the first paragraph:

Channel 0 can be used for EISA READ operation only. Channel 1 can be used for both EISA READ and EISA WRITE operations.

Section 8.1.2 Bits 7–4 has been changed to Third hex digit of product number

Bits 4–0 has been changed to Hexadecimal digit of product revision

Section 8.2.3.3 Sentence added to Bit 22:

This is applicable only to channel 1 and not for channel 0, as channel 0 can perform EISA WRITE transfers only.

Section 10.0 Figure 10-1 TDIR, TCHAN timing diagram has changed from 10 to 11.

Figure 10-3 TDIR, TCHAN timing diagram has changed from 10 to 11.

Figure 10-7 TDIR, TCHAN timing diagram has changed from 10 to 11.

The following changes have been made since revision 005:

Section 4.3 New paragraph added at the end. This paragraph reads:

Any consecutive Peek/Poke or Locked exchanged transfers must be initiated only after the previous Peek/Poke or Locked exchange has been completed. This can be accomplished by making sure that bit 2 of the local status/control register is set to a zero before initiating the transfer.

Section 4.8 New paragraph added after the third paragraph. This paragraph reads:

The IDOEs do not go active during an IOSEL cycle outside the shared register access space.

Section 7.4 The third paragraph had two sentences deleted that is replaced with Figure 7-2, IDOEO# Connection during ID Register Access. The sentences that were deleted read as follows:

The external lines connected to the IDAT <7:0> lines should be connected to the bus between the BMIC and the external F245 data buffers. The BMIC will enable the external data buffers to drive byte lane 0 of the EISA bus upon detection of the ID address.

The following changes have been made since revision 004:

Section 4.3 New paragraph added at the end. This paragraph reads:

Whenever the BMIC is commanded to do an EISA POKE cycle, the BMIC will assert the MREQ# signal low normally, transfer up to four bytes of data, and release the bus by de-asserting MREQ# high. A potential problem exists, however, when the slave device extends the cycle by de-asserting EXRDY low. If the slave holds this signal low past the time that the BMIC is forced to release MREQ# high (it has been preempted while waiting for the slave to assert EXRDY high), then the BMIC will drive MREQ# back low again immediately after this cycle ends if there is another transfer pending (TBI, PEEK, POKE or LOCKED-EXCHANGE). Note that according to the EISA spec, "A bus master must wait at least two BCLKs after releasing the bus before re-asserting its MREQx*"(EISA spec, MREQ* signal description). To adhere to EISA specifications, it is required that LOCKED-EXCHANGE cycles be used in lieu of POKE cycles.

Section 8.2.3.4 New sentence added at the end of paragraph one:

At the end of a transfer, this register contains the value of the number of bytes transferred during the last cycle.

Section 8.2.4.2 Two paragraphs added for the CFGCL bit:

Before a channel is issued a clear command, the channel must first be suspended by writing a "1" into Bit 0 (CFGSU) of the Transfer Channel Configuration Register. Next the Transfer Channel Status Register must be read. If the TSTTC (Bit 0) is set to a "1", then the channel has already completed the transfer. The channel is then unsuspended (write a "0" into Bit 0 [CFGSU] of the Transfer Channel Configuration Register), and the TSTTC bit is then cleared.

If the TSTTC bit is a "0", then the TSTEN bit is checked. If this bit is a "1", then the channel has not returned to idle yet, and the Transfer Channel Status Register is repolled. If the TSTEN bit is a "0", then the channel has successfully returned to idle and can now be cleared with no errors. This is done by setting Bit 2 (CFGCL bit) to a 1 in the Transfer Channel Configuration Register. A flowchart for this operation is shown in the following figure.

Figure 8-1 was added to the data sheet.

Section 8.2.4.3 Bit 4 has been changed to reserved.

First paragraph under bit description has been changed to read, "Bits (7), (6), TST1K, and (4) are reserved. Any data read from these bits should be ignored".

The paragraph following this one has been deleted.

Section 11.2 Max Limits on symbols C_{OUT} and C_{CLK} have been changed.

Note 1 has been corrected.

Section 12.1 Symbol t1 has been corrected to read 2500 instead of 250 for max.

Symbol t20 has been corrected to read 3 for min.

Symbol t22 has been corrected to 4 from 7 for min.

Symbol t35 has been corrected to 5 from 7 for min.

Symbol t37 has been corrected to 3 for min.

Symbol t40 has been corrected to 1 for min.

Symbol t50 has been corrected to 124 from 125 for max.

Symbol t51 has been corrected to 1 for min.

Section 12.2 The Local CPU Interface Timing (Read Cycle/Shared Register Access) table has been corrected. The signal LDAT<7:0> has changed one portion from valid to float.