intel

ISA Bus Specification and Application Notes

September 12, 1989

INTEL 386™ MICROCOMPUTER
OEM PLATFORMS

OCPD Technical Marketing

Intel ISA Bus Specification and Application Notes

Rev. 2.01

SEPTEMBER 12, 1989

With respect to information contained within this document, INTEL makes no warranties of merchantability or fitness for a particular purpose or any other warranties, representations or conditions, express or implied by statue, trade usage, or otherwise.

Furthermore, INTEL shall not be liable for any loss, claim, injury, special or consequential damages arising from the use of any information contained herein.

The information in this document is subject to change without notice.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1.0 SCOPE

The Intel ISA bus is a parallel bus derived from the IBM PC/AT memory and I/O bus. This document supplies the electrical and mechanical specifications for the Intel ISA bus as implemented at the connectors of ISA compatible platform products supplied by the Intel OEM Platforms organization. This document also describes the bus interface, connector pin-out, and electrical interface to standard IBM PC/AT[1] cards.

The main focus is to provide application information for our customers to successfully design add-on cards for Intel ISA-based platforms. The task of assembling this specification is difficult in that thousands of add-on cards have been designed without the benefit of a document of this nature. The details in this document represent the accumulation of timings, platform architecture, and good design practices developed during the design of a variety of Intel board products. There is no guarantee that an add-on card compliant to this document will work in any or all Intel or non-Intel ISA platform products. Newly developed add-on cards must be validated with a variety of actual ISA-based platforms.

Any questions, comments, or clarifications would be appreciated and can be addressed to:

Intel, OEM Platforms Operation
Technical Marketing, HF1-60
5200 N.E. Elam Young Parkway
Hillsboro, Oregon 97124

2.0 NATURE OF THIS SPECIFICATION

This specification applies to all Intel ISA bus platforms. It specifically contains application information to aid in the design of add-on cards. Any differences between this specification and the ISA bus implementation of a specific Intel baseboard must be reviewed in the hardware reference manual of the product.

Three unique headings are used to draw attention to important information. These headings are:

NOTE Special information worthy of consideration.

CAUTION Information that can cause fatal system operation if not properly

[1] PC/AT, PC/XT, and IBM are trademarks of International Business Machines Corporation.

considered.

ADD-ON CARD DESIGN FOCUS Useful information to consider when designing an add-on card.

3.0 CONVENTIONS

Throughout this specification, an asterisk (*) following a signal name indicates that the signal is in its logically true state when its voltage level is in the low state. A signal name that is not followed by an asterisk indicates that the signal is in its logically true state when its voltage level is in the high state.

To avoid confusion when referring to the logical state of signals, the terms "enabled" and "disabled" are used throughout this document. A signal is said to be enabled when it is logically true, and it is said to be disabled when it is logically false.

Many signals on the Intel ISA bus are more easily or conveniently discussed as a group, since there are many signals with identical functionality. Names for signals within these groups follow a decimal radix numbering convention, as follows.

(1) When discussed as an individual signal, the decimal number is simply appended to the signal name, e.g., **A15**.

(2) A disjoint set of signal lines with the same signal group may be collectively referenced by listing the group name and appending the decimal numbers within brackets, e.g., **A<15,12,03>**.

(3) A range of consecutive signals within the same group may be referenced by listing the group name and appending the beginning and ending signals, separated by double periods, e.g., **A<15..08>**. Ranges of signals include the beginning and ending signals in the range.

(4) Consecutive and disjoint signals within the same group are referenced by using a combination of both methods (2) and (3), e.g., **A<15,07..00>**.

(5) A signal group name with nothing appended to it refers to the entire signal group, e.g., **A** is equivalent to **A<15..00>**.

A set of brackets "[]" are used to indicate the connector size. [8] is an 8-bit resource and [8/16] supports an 8- or 16-bit resource.

Signal lines and groups of signal lines on the Intel ISA bus are always shown in bold capital letters, as in the case of the individual signal "**MEMREF***".

The use of the word "bit" always refers to singular or plural to data bits unless the adjective "address" precedes it.

4.0 ARCHITECTURE OVERVIEW

The Intel ISA bus is part of a the Intel ISA compatible platform architecture. The resources of this architecture that interact with the ISA bus are the Primary CPU, DMA Controller, Interrupt Controller, Refresh Controller, Memory, Byte Swapper, add-on cards, RTC/Timer-counter, and I/O resources (See Figure 4.0). The Primary CPU, DMA Controller, Refresh Controller and add-on cards are the only resources that can become bus owners and are defined as follows:

Primary CPU
A CPU that resides on the ISA compatible platform and is the default bus owner. The DMA and Refresh Controllers disable this resource to obtain bus ownership. The Primary CPU also responds to interrupt requests via the Interrupt Controller.

DMA Controller
This resource is connected to the DMA request and acknowledge lines. An active DMA request will enable this resource to become the bus owner and transfer data between memory and I/O resources on the platform.

Add-on Cards
A resource that is attached to the platform via the ISA bus connectors. Add-on cards can become bus owners, memory or I/O access resources, or participate in memory or I/O DMA transfers.

Refresh Controller
The Refresh Controller becomes bus owner and generates an address and read pulse to refresh DRAM resources on the platform or add-on card.

Other resources cannot become bus owners but are essential in maintaining IBM/AT compatibility. These resources are defined as follows: (See Figure 4.0)

RTC / Timer-counter
This resource consists of the real time clock (RTC) for the date and time, and an Intel 8254A. One of the

timer-counters is used to generate a pulse every 15 usec. to trigger the Refresh Controller to do a refresh cycle.

Backplane

The portion of the platform that interconnects the connectors with other platform resources.

Platform Memory

Some or all of the random access memory devices (RAM) are on the platform proper (Platform Memory). Other RAM resources may reside on the add-on cards.

Platform I/O

Some or all of the I/O resources (like serial and parallel ports) can reside on the platform. The remainder of the I/O resources can reside on the add-on cards.

Interrupt Controller

This resource is connected to the interrupt request lines of the bus and requests service from the Primary CPU. The Interrupt Controller is the Intel 8259A.

Data Swapper

This resource allows 8- and 16-bit resources to interact.

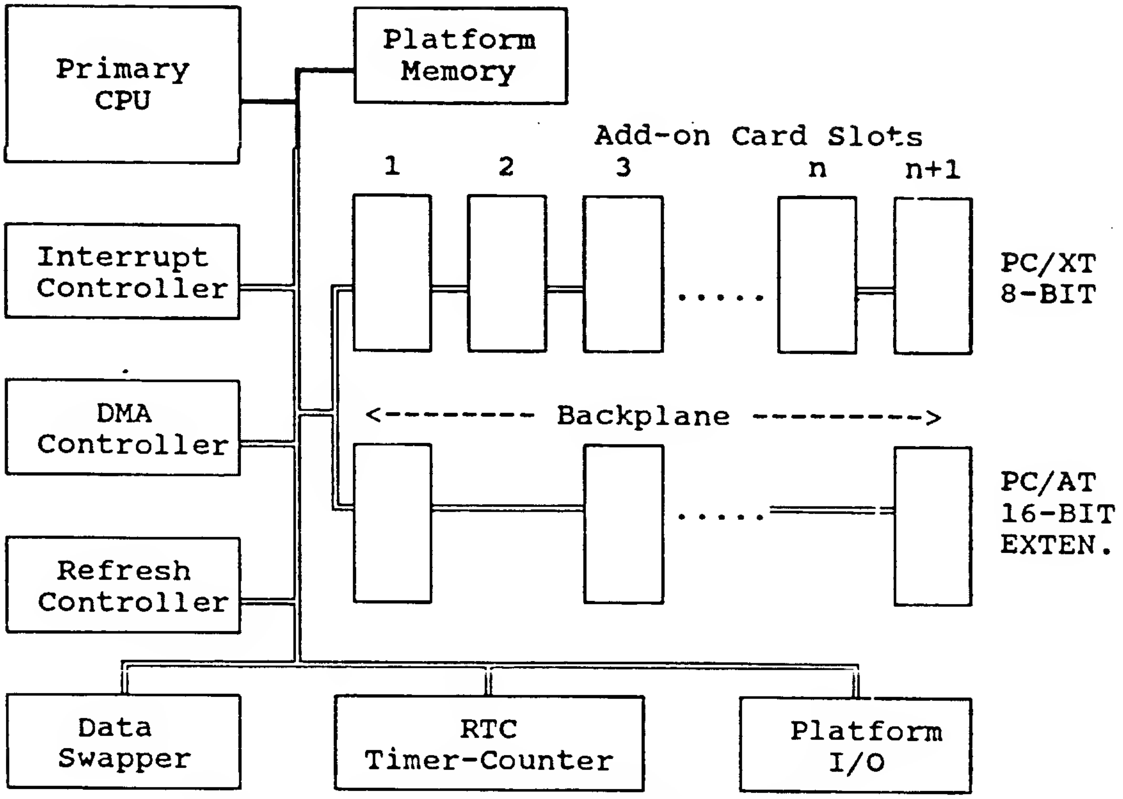


Figure 4.0 PLATFORM ARCHITECTURE

The Intel ISA bus is the combination of the aforementioned backplane and connectors that interconnect add-on card slots and platform resources. The add-on card slots can support either 8- or 16-bit add-on cards. The [8] slot contains a single connector; the [8/16] has one additional connector. The single connector slot can support only eight data bits. The double connector slot can support either 8 or 16 data bits. The total number of slots is limited to load and transmission line issues, however most implementations have eight slots due to the available DMA channels and interrupt lines.

5.0 BUS OWNER ATTRIBUTES AND DESIGN

5.1 PRIMARY CPU

The Primary CPU is the default bus owner. The Refresh and DMA Controllers (and the add-on cards by subletting from the DMA Controller) become bus owners only after disabling it. The disabling of the Primary CPU is accomplished by a handshake of its hold request and hold acknowledge lines by the DMA or Refresh

Table 5.1 outlines which signal line resources are driving or receiving when the Primary CPU is bus owner. It also specifies the type of driver.

LINE	DRIVERS & RECEIVERS						DRIVER
	PRI	ADD	DMA	REF	MEM	I/O	TYPE
AEN	-	R	D	-	-	R	TTL
BUSALE	D	R	-	-	R	R	TTL
DACK#*	-	R	D	-	-	R	TTL
DRQ# (1)	-	D	R	-	-	D	TTL
IOCHCK*	R	D	-	-	D	D	OC
IOCHRDY	R	D	-	-	D	D	OC
IOCS16*	R	D	-	-	-	D	OC
IORC*	D	R	-	-	-	R	TRI
IOWC*	D	R	-	-	-	R	TRI
IRQ# (2)	R	D	-	-	-	D	TTL
LA#	D	R	-	-	R	-	TRI
SECMAST*	-	-	-	-	-	-	OC
MCS16*	R	D	-	-	D	-	OC
MRDC*	D	R	-	-	R	-	TRI
MWTC*	D	R	-	-	R	-	TRI
84OSC	D	R	-	-	-	-	TTL
MEMREF*	-	-	-	-	-	-	OC
RSTDE(3)	D	R	R	-	-	R	TTL
A#	D	R	-	-	R	R	TRI
SBHE#(4)	D	R	-	-	R	R	TRI
D#	D/R	D/R	-	-	D/R	D/R	TRI
MEMR*	(5)	R	-	-	R	-	TRI
MEMW*	(5)	R	-	-	R	-	TRI
SYSCLK	D	R	R	-	R	-	TRI
TC	-	-	-	-	-	-	TTL
SRDY*	R	D	-	-	D	-	OC

Table 5.1 PRIMARY CPU IS BUS OWNER

NOTE: Primary CPU = PRI, Add-on card = ADD, DMA Controller = DMA, Refresh Controller = REF, Platform memory = MEM, Platform I/O = I/O, TTL = "ALS" or "LS", Open collector = OC, and Tristatable drivers/receivers = TRI.

"-" indicates that the respective signal line is not enabled or sensed by the resource.

"X" indicates a "don't care" condition. A resource can enable the signal, but it will be ignored by other resources.

(1) DRQ# can be driven but will not be acted upon until the DMA Controller is bus owner.

(2) Received by the Primary CPU via the Interrupt Controller and acted upon at the discretion of the Primary CPU when it is bus owner.

(3) This signal must be sensed at all times and immediately acted upon if enabled.

(4) Always received by the byte swapper hardware.

(5) Driven by platform resources, the address is in the first 1MB of the address and either **MRDC*** or **MWTC*** signal lines.

5.2 DMA CONTROLLER

The DMA lines of the connector are directly attached to the Intel 8237A DMA Controller. When the DMA request lines are driven by a resource, the DMA Controller obtains bus ownership by handshaking the hold request and hold acknowledge lines of the Primary CPU. Once bus ownership is granted, the associated DMA acknowledge line is driven and the DMA transfer cycle begins. When the DMA acknowledge line is attached to an add-on card, the DMA transfer cycles will not begin if the **SECMAST*** line is enabled by the add-on card (See Section 5.4 for more information).

The I/O resources involved in a DMA transfer must match in data size of the DMA channel. DMA channels 0-3 only support 8-bit I/O resources; all data must be transferred as data bits on the data lines D<07..00>. Byte swapping hardware on the platform will use **A0** and **SBHE*** to port the high byte for 16-bit memory to the low byte of the 8-bit I/O resource when needed. DMA channels 5-7 only support 16-bit I/O resources; all data must be transferred as 16 data bits on data lines **D<15..00>**. The memory involved with the transfer must have a data size of 16 bits; the platform byte swapping hardware will not compensate for data size mismatch.

NOTE

An 8-bit memory resource can only be part of a DMA transfer with an 8-bit I/O resource; the use of 8-bit memory resource with a 16-bit I/O resource is not allowed.

CAUTION

The Refresh Controller cannot become bus owner when the DMA Controller is bus owner. Thus, continuous bus ownership by the DMA Controller for transfers longer than 15 usec. may cause data to be lost in resources that have dynamic RAM and rely on ISA bus refresh cycles.

ADD-ON CARD DESIGN FOCUS

DMA request and acknowledge lines are driven by the TTL drivers, and are attached to all the slots. The add-on cards must allow the selection of different DMA channels at the time of installation to avoid conflict with already installed cards or platform resources.

The add-on cards can only be in the DMA memory or I/O modes when interacting with the DMA Controller as a DMA resource. See Section 5.3.

Table 5.2 outlines which signal lines the platform resources are driving receiving when the DMA Controller is bus owner. It also specifies the type of driver.

LINE	DRIVERS & RECEIVERS						DRIVER
	PRI	ADD	DMA	REF	MEM	I/O	TYPE
AEN	-	R	D	-	-	R	TTL
BUSALE	(6)	R	-	-	R	R	TTL
DACK#*	-	R	D	-	-	R	TTL
DRQ# (1)	-	D	R	-	-	D	TTL
IOCHCK*	R	D	-	-	D	D	OC
IOCHRDY	-	D	R	-	D	D	OC
IOCS16*	-	X	X	-	-	X	OC
IORC*	-	R	D	-	-	R	TRI
IOWC*	-	R	D	-	-	R	TRI
IRQ#(2)	R	D	-	-	-	D	TTL
LA#	-	R	D	-	R	-	TRI
SECMAST*	-	-	-	-	-	-	OC
MCS16*	-	X	X	-	X	-	OC
MRDC*	-	R	D	-	R	-	TRI
MWTC*	-	R	D	-	R	-	TRI
84OSC	D	R	-	-	-	-	TTL
MEMREF*	-	-	-	-	-	-	OC
RSTDE(3)	D	R	R	-	-	R	TTL
A#	-	R	D	-	R	R	TRI
SBHE#(4)	-	R	D	-	R	-	TRI
D#	-	D/R	-	-	D/R	D/R	TRI
MEMR*	(5)	R	-	-	R	-	TRI
MEMW*	(5)	R	-	-	R	-	TRI
SYSCLK	D	R	R	-	R	-	TRI
TC	-	R	D	-	-	R	TTL
SRDY*	-	-	-	-	-	-	OC

Table 5.2 DMA CONTROLLER IS BUS OWNER

NOTE: Primary CPU = PRI, Add-on card = ADD, DMA Controller = DMA, Refresh Controller = REF, Platform memory = MEM, Platform I/O = I/O, TTL = "ALS" or "LS", Open collector = OC, and Tristatable drivers/receivers = TRI.

"-" indicates that the respective signal line is not enabled or sensed by the resource.

"X" indicates a "don't care" condition. A resource can enable the signal, but it will be ignored by other resources.

(1) DRQ* can be driven but will not be acted upon until the DMA Controller is bus owner.
(2) Received by the Primary CPU via the Interrupt Controller and acted upon at the discretion of the Primary CPU when it is bus owner.
(3) This signal must be sensed at all times and immediately acted upon if enabled.
(4) Always received by the byte swapper hardware.
(5) Driven by platform resources, the address is in the first 1MB of the address and either **MRDC*** or **MWTC*** signal lines.
(6) Driven to the enabled level by platform hardware for the entire cycle.

5.3 ADD-ON CARDS

Add-on cards can operate in five different modes: bus owner, DMA memory or I/O, access memory or I/O resource, refresh, or reset. Add-on cards can support any combination of the first four modes, but all add-on cards must comply with the reset mode.

BUS OWNER MODE

Only add-on cards of 16-bit data size plugged into an [8/16] slot can become a bus owner. The add-on card becomes bus owner by driving a DRQ* and the **SECMAST*** line once the associated **DACK** line is driven by the DMA Controller. The add-on cards can begin ISA bus access cycles as 8- or 16-bit. The completion of the cycle as 16-bits is also dependent on the state of **MCS16*** and **IOCS16*** as driven by accessed resources.

The cycles executed by the add-on card are all access cycles. The add-on card cannot execute DMA transfer cycles because all of the DMA control lines are attached to the platform DMA Controller and can't be shared by DMA Controllers if one resides on the add-on card. When an add-on card is bus owner, the DMA Controller disables the **AEN** line when it sublets bus ownership. Disabling the **AEN** line allows I/O resources to decode the address lines and be accessed by the add-on card. With **AEN** disabled, no DMA transfers can occur. (See **AEN** description in Section 7.1.) Also, DMA transfer cycles cannot be executed because the DMA Controller has the channel that granted bus ownership active; the other DMA channels can't be activated until the channel involved with add-on card bus ownership is no longer active.

NOTE

Software supplied with the add-on card must instruct the Primary CPU to program a specific DMA channel in the cascade mode. The DMA channel must be programmed into the cascade mode in order for the associated add-on card to become bus owner.

NOTE

If the add-on card begins an access as 16-bit in size, and **MCS16*** or **IOCS16*** is not enabled, then the cycle is completed as 8-bits. The platform byte swapper will port the 8-bit byte between **D<15..08>** and **D<07..00>** as determined by **SBHE*** and **A0**.

If the add-on card begins an access as 8-bit in size, then the platform must support the appropriate byte swapping. Traditionally not all platforms have supported 8-bit add-on cards as bus owners. Please refer to the technical reference manual for the platform.

See Section 6.4 for more information.

CAUTION

The add-on card that becomes bus owner must drive the **MEMREF*** line every 15 usec. to request the Refresh Controller to execute a refresh cycle. The Refresh Controller actually executes the cycle by driving the address lines, command lines, and monitoring the **IOCHRDY** line, but the add-on card drives the **MEMREF*** line and retains bus ownership.

The add-on card must be responsible for requesting the refresh cycle because the Refresh Controller can't become bus owner if the DMA Controller is the bus owner. Recall that the add-on card becomes bus owner by subletting from the DMA Controller which is active and indirectly bus owner.

Table 5.3 outlines which signal lines the platform resources are driving receiving a signal when the add-on card is bus owner. It also specifies the type of driver.

LINE	DRIVERS & RECEIVERS							DRIVER TYPE
	PRI	ADDX	ADD	DMA	REF	MEM	I/O	
AEN	-	-	R	D	-	-	R	TTL
BUSALE	(6)	-	R	-	-	R	R	TTL
DACK#*	-	R	R	D	-	-	R	TTL
DRQ# (1)	-	D	-	D	-	-	D	TTL
IOCHCK*	R	D	D	-	-	D	D	OC
IOCHRDY	-	R	D	-	-	D	D	OC
IOCS16*	-	R	D	-	-	-	D	OC
IORC*	-	D	R	-	-	-	R	TRI
IOWC*	-	D	R	-	-	-	R	TRI
IRQ# (2)	R	D	D	-	-	-	D	TTL
LA#	-	D	R	-	-	R	-	TRI
SECMAST*	-	D	-	R	-	-	-	OC
MCS16*	-	R	D	-	-	D	-	OC
MRDC*	-	D	R	-	-	R	-	TRI
MWTC*	-	D	R	-	-	R	-	TRI
84OSC	D	R	R	-	-	-	-	TTL
MEMREF*	-	-	-	-	-	-	-	OC
RSTDE(3)	D	R	R	R	-	-	R	TTL
SA#	-	D	R	-	-	R	R	TRI
SBHE#(4)	-	D	R	-	-	-	-	TRI
SD#	-	D/R	D/R	-	-	D/R	D/R	TRI
MEMR*	(5)	-	R	-	-	R	-	TRI
MEMW*	(5)	-	R	-	-	R	-	TRI
SYSCLK	D	R	R	R	-	R	R	TRI
TC	-	-	-	-	-	-	-	TTL
SRDY*	-	-	-	-	-	-	-	OC

Table 5.3 ADD-ON CARD ACCESS PLATFORM OR ADD-ON CARD MEMORY OR I/O
ADD-ON CARD (ADDX) IS BUS OWNER

NOTE: Primary CPU = PRI, Add-on card = ADD, DMA Controller = DMA, Refresh Controller = REF, Platform memory = MEM, Platform I/O = I/O, TTL = "ALS" or "LS", Open collector = OC, and Tristatable drivers/receivers = TRI.

"-" indicates that the respective signal line is not enabled or sensed by the resource.

"X" indicates a "don't care" condition. A resource can enable the signal but it will be ignored by other resources.

(1) DRQ# can be driven but will not be acted upon until the DMA Controller is bus owner.
(2) Received by the Primary CPU via the Interrupt Controller and acted upon at the discretion of the Primary CPU when it is bus owner.
(3) This signal must be sensed at all times and immediately acted upon if enabled.
(4) Always received by the byte swapper hardware.
(5) Driven by platform resources, the address is in the first 1MB of the address and either **MRDC*** or **MWTC*** signal lines.
(6) Driven to the enabled level by platform hardware for the entire cycle.

DMA MEMORY OR I/O MODES

The add-on card can only be in the DMA modes when the DMA Controller is bus owner. The DMA memory mode allows data to transfer between other I/O resources and the add-on card's memory. The DMA I/O mode allows data to transfer between memory and add-on card's I/O by handshaking the DMA request and DMA acknowledge lines. The add-on card that responds as an 8-bit or 16-bit I/O resource must use the 8- and 16-bit DMA channels, respectively. An add-on card can support both memory and I/O modes at once, which would transfer data between add-on card memory and I/O.

The condition of the add-on card signal lines when the DMA Controller is bus owner is outlined in Table 5.2.

CAUTION

There are special considerations when the DMA Controller is executing a transfer cycle between an 8-bit I/O resource and a 16-bit add-on card memory resource. First, the add-on card knows that the transfer is with an 8-bit I/O resource because of **SBHE*** and **A0**. Second, when memory is being written, the platform byte swapper hardware will place the byte either on **D<15..08>** or **D<07..00>**; the add-on card must monitor **SBHE*** and **A0** to determine which data lines contain the correct byte. Third, when memory is being read, the platform byte swapper will port the byte from **D<15..08>** to **D<07..00>** when appropriate. The add-on card must monitor **SBHE*** and **A0** to determine when it must tristate **D<07..00>** to prevent buffer fights.

The add-on card can be a 16-bit memory resource in a DMA transfer for I/O resources of either 8- or 16-bit data size. The add-on card can be an 8-bit memory resource in a DMA transfer only if the I/O resource is 8-bits.

Another consideration is a DMA transfer cycle that writes a memory resource when an add-on card is the 8-bit I/O resource of the transfer. If the add-on card is installed in an [8/16] slot, then it must tristate the D<15..08> lines. These lines must be tristated to prevent buffer fights with the platform byte swapper when it is porting the low byte to the high byte during the transfer cycle.

See Section 6.4 for further information.

CAUTION

When the DMA Controller is bus owner it ignores the **SRDY*** signal; thus the add-on card cannot implement fast RAM DMA transfers.

ACCESS MEMORY OR I/O MODE

The add-on card can appear simply as a memory or I/O resource when the Primary CPU or another add-on card is bus owner.

CAUTION

There are special considerations when an add-on card resides in an [8/16] slot and responds as an 8-bit memory or I/O resource during an access cycle. When the add-on card resource is being read, the platform byte swapper hardware will place the byte either on D<15..08> or D<07..00> to support the 16-bit data size of the bus owner. The add-on card must tristate **D<15..08>** because these lines will be driven by the platform byte swapper hardware.

See Section 6.4 for further information.

CAUTION

When some add-on cards are bus owners they ignore the **IOCHRDY** or **SRDY*** signals and run a default 8-bit or 16-bit memory cycle. Any add-on card that returns **IOCHRDY** or **SRDY*** signals to an add-on card as it does with a Primary CPU must determine if the accessing add-on card can support these lines.

Tables 5.1 and 5.3 outline which signal lines the platform resources are driving or receiving when an add-on card is in the access memory or I/O mode and the Primary CPU or another add-on card (noted as ADDX) are bus owners, respectively. It also specifies the type of driver.

RESET MODE

The add-on card enters the reset mode whenever the **RSTDEV** is enabled, independent of what other mode it is in. All of the add-on card bus tristatable signals must be tristated and all open collector signals must be disabled within 500ns of **RSTDEV** being enabled. The board must complete its initialization within one millisecond of the **RSTDEV** being enabled and be ready for normal bus operations. Normal bus operations commence immediately when the **RSTDEV** signal line is disabled.

5.4 REFRESH CONTROLLER

The Refresh Controller executes a read cycle at a specific address to refresh the platform or add-on cards' DRAM. Every 15 usec. the Refresh Controller tries to obtain bus ownership in order to run a refresh cycle. If the present bus owner is the Primary CPU, then bus ownership is transferred to the Refresh Controller. If the present bus owner is an add-on card, then the Refresh Controller will execute a refresh cycle only when the add-on card enables the **MEMREF*** line. If the DMA Controller is the bus owner, then no refresh cycles can be executed until the DMA Controller relinquishes bus ownership.

When a refresh cycle is executed, the Refresh Controller drives address line **A<07..00>** with one of the possible 256 refresh addresses. The other address lines are undefined and should be tristated by resources that can drive them. The cycle is essentially a normal or ready type access cycle with both **MEMR*** and MRDC* enabled.

CAUTION

The refresh cycle must be executed every 15 microseconds to ensure that all DRAM addresses are accessed every 4 milliseconds. If this does not occur, DRAM data may be lost.

Tables 5.4.1 and 5.4.2 outline which signal lines the platform resources are driving and receiving a signal for a refresh cycle when the Refresh Controller and add-on card are bus owners, respectively. It also specifies the type of driver.

LINE	DRIVERS & RECEIVERS						DRIVER
	PRI	ADD	DMA	REF	MEM	I/O	TYPE
AEN	-	X	X	-	-	X	TTL
BUSALE	(6)	R	-	-	R	R	TTL
DACK#*	-	R	D	-	-	R	TTL
DRQ#-(1)	-	D	R	-	-	D	TTL
IOCHCK*	R	D	-	-	D	D	OC
IOCHRDY	-	D	-	R	D	D	OC
IOCS16*	X	X	X	X	X	X	OC
IORC*	-	-	-	-	-	-	TRI
IOWC*	-	-	-	-	-	-	TRI
IRQ# (2)	R	D	-	-	-	D	TTL
LA#	-	-	-	-	-	-	TRI
SECMAST*	-	-	-	-	-	-	OC
MCS16*	X	X	X	X	X	X	OC
MRDC*	-	R	-	D	R	-	TRI
MWTC*	-	-	-	-	-	-	TRI
84OSC	D	R	-	-	-	-	TTL
MEMREF*	-	R	-	D	R	-	OC
RSTDE(3)	D	R	R	-	-	R	TTL
A							
<07..00>	-	R	-	D	R	R	TRI
<19..08>	-	-	-	-	-	-	TRI
SBHE#(4)	-	-	-	-	-	-	TRI
D#	-	-	-	-	-	-	TRI
MEMR*	(5)	R	-	-	R	-	TRI
MEMW*	-	-	-	-	-	-	TRI
SYSCLK	D	R	R	-	R	-	TRI
TC	-	-	-	-	-	-	TTL
SRDY*	-	-	-	-	-	-	OC

Table 5.4.1 REFRESH CONTROLLER IS BUS OWNER

NOTE: Primary CPU = PRI, Add-on card = ADD, DMA Controller = DMA, Refresh Controller = REF, Platform memory = MEM, Platform I/O = I/O, TTL = "ALS" or "LS", Open collector = OC, and Tristatable drivers/receivers = TRI.

"-" indicates that the respective signal line is not enabled or sensed by the resource.

"X" indicates a "don't care" condition. A resource can enable the signal but it will be ignored by other resources.

(1) DRQ# can be driven but will not be acted upon until the DMA Controller is bus owner.
(2) Received by the Primary CPU via the Interrupt Controller and acted upon at the discretion of the Primary CPU when it is bus owner.
(3) This signal must be sensed at all times and immediately acted upon if enabled.
(4) Always received by the byte swapper hardware.
(5) Driven by platform resources, the address is in the first 1MB of the address and either **MRDC*** or **MWTC*** signal lines.
(6) Driven to the enabled level by platform hardware for the entire cycle.

LINE	DRIVERS & RECEIVERS						DRIVER
	PRI	ADD	DMA	REF	MEM	I/O	TYPE
AEN	-	X	X	-	-	X	TTL
BUSALE	(6)	R	-	-	R	R	TTL
DACK#*	-	R	D	-	-	R	TTL
DRQ# (1)	-	D	R	-	-	D	TTL
IOCHCK*	R	D	-	-	D	D	OC
IOCHRDY	-	D	-	R	D	D	OC
IOCS16*	X	X	X	X	X	X	OC
IORC*	-	-	-	-	-	-	TRI
IOWC*	-	-	-	-	-	-	TRI
IRQ#(2)	R	D	-	-	-	D	TTL
LA#	-	-	-	-	-	-	TRI
SECMAST*	-	D	R	-	-	-	OC
MCS16*	X	X	X	X	X	X	OC
MRDC*	-	R	-	D	R	-	TRI
MWTC*	-	-	-	-	-	-	TRI
84OSC	D	R	-	-	-	-	TTL
MEMREF*	-	(7)	-	R	R	-	OC
RSTDE(3)	D	R	R	-	-	R	TTL
A#	-	-	-	-	-	-	TRI
<07..00>	-	R	-	D	R	R	TRI
<19..08>	-	-	-	-	-	-	TRI
SBHE#(4)	-	-	-	-	-	-	TRI
D#	-	-	-	-	-	-	TRI
MEMR*	(5)	R	-	-	R	-	TRI
MEMW*	-	-	-	-	-	-	TRI
SYSCLK	D	R	R	-	R	-	TRI
TC	-	-	-	-	-	-	TTL
SRDY*	-	-	-	-	-	-	OC

Table 5.4.2 ADD-ON CARD IS BUS OWNER & HAS ENABLED MEMREF*

NOTE: Primary CPU = PRI, Add-on card = ADD, DMA Controller = DMA, Refresh Controller = REF, Platform memory = MEM, Platform I/O = I/O, TTL = "ALS" or "LS", Open collector = OC, and Tristatable drivers/receivers = TRI.

"-" indicates that the respective signal line is not enabled or sensed by the resource.

"X" indicates a "don't care" condition. A resource can enable the signal but it will be ignored by other resources.

(1) DRQ# can be driven but will not be acted upon until the DMA Controller is bus owner.
(2) Received by the Primary CPU via the Interrupt Controller and acted upon at the discretion of the Primary CPU when it is bus owner.
(3) This signal must be sensed at all times and immediately acted upon if enabled.
(4) Always received by the byte swapper hardware.
(5) Driven by platform resources, the address is in the first 1MB of the address and either **MRDC*** or **MWTC*** signal lines.
(6) Driven to the enabled level by platform hardware for the entire cycle.
(7) Enabled by the add-on card that is bus owner.

6.0 NON-BUS OWNER ATTRIBUTES

The Intel ISA bus has several unique attributes that are independent of bus ownership.

6.1 MEMORY ADDRESS SPACE

The maximum memory address space supported by ISA bus is 16MB (24 address lines), but not all add-on card slots can support the full address space. When a bus owner accesses platform or add-on card slot memory, it must enable **MRDC*** or **MWTC***; the platform hardware in turn enables the **MEMR*** or **MEMW*** lines if the access is in the first 1MB. Only the **MEMR***, **MEMW***, **D<07..00>**, and **A<19..00>** lines are attached to an [8] slot; thus [8] slot resources can only have an 8-bit data size and reside in the first 1MB of the memory address space. [8/16] add-on card slots receive all command, address, and data lines; thus these resources can respond as either 8- or 16-bit data resources anywhere in the memory address space. The access will be completed as a 16-bit cycle if **MCS16*** is enabled.

NOTE

The ability of the platform or add-on card memory to respond as a 16-bit memory resource requires that **MCS16*** is enabled. **MCS16*** is based on decoding **LA<23..17>**; thus the data size of each 128KB block on 128KB address boundaries must be entirely 8- or 16-bits. Different portions of each 128KB block cannot be of a different data size, as this would require decoding of the other address lines to generate **MCS16***.

CAUTION

The DRAM along with other bus resources require a refresh cycle. If the refresh operation is not executed every 15 usec., then a loss of data may occur.

ADD-ON CARD DESIGN FOCUS

The memory resource on the platform are of two types dynamic random access memory (DRAM) and EPROM space. The DRAM is either 16 or 32 data bits in size, depending the data size of the Primary CPU; but always appears as a 16-bit data resource to the add-on card. The EPROM contains the BIOS and is always 16-bits.

Refer to the Technical Reference Manual of an Intel ISA-based platform for information on memory mapping. A thorough study of memory usage is advised before attempting an add-on card design.

6.2 I/O ADDRESS SPACE

The maximum I/O address space supported by the ISA bus is 64KB (16 address lines). All of the slots support 16 address lines. The first 256 bytes are reserved for the platform resource registers of Interrupt and DMA Controllers, timer/counter, RTC, and other elements for AT compatibility. The rest of the I/O address space accesses resources on the ISA bus. Refer to the Technical Reference Manual of an Intel ISA-based platform for information on I/O address space mapping.

ADD-ON CARD DESIGN FOCUS

Even though 16 address lines are available, traditionally only the first ten address lines have been decoded by the add-on cards for an I/O access. This causes the 1KB block referenced to each and every 1KB address boundary to be a repeat of the first 1KB block. Thus, the first 256 bytes that access XT/AT compatible platform resources is repeated at the beginning of each and every 1KB address boundary. Add-on card resources must not use this portion of the 1KB blocks.

If all of the add-on cards attached to the platform and the platform itself decode all 16 address lines, then the first 1KB block is not repeated throughout the address space. The XT/AT compatible resources only reside in the first 256 bytes of the first 1KB.

6.3 INTERRUPT STRUCTURE

The interrupt lines of the connectors are directly connected to the Intel 8259A Interrupt Controller. The Interrupt Controller will react to the interrupt on a transition of low to high. There are no interrupt acknowledge lines on the ISA bus. The resource must use a memory or I/O access by the bus owner for an acknowledgement of the interrupt.

ADD-ON CARD DESIGN FOCUS

The interrupt control lines are attached to all of the slots and are edge triggered. The add-on cards must allow the selection of the interrupt line at the time of installation to avoid conflict with already installed cards or platform resources.

6.4 DATA SWAPPING

A PRIMARY CPU and an add-on card can execute either 8- or 16-bit access cycles. A cycle will be completed as 8-bits unless MCS16* or IOCS16* are enabled by the accessed resource. An 8-bit cycle is always completed as 8-bits independent of the size of the accessed resource. In either case, the platform must have the appropriate byte swapping hardware.

The byte swapper hardware resides on the platform. It adjusts for data size mismatch between resources. The mismatch can occur during access cycles as documented in Figure 6.4.1 and Table 6.4.1. The mismatch can also occur during DMA transfer cycles as documented in Figure 6.4.2 and Table 6.4.2.

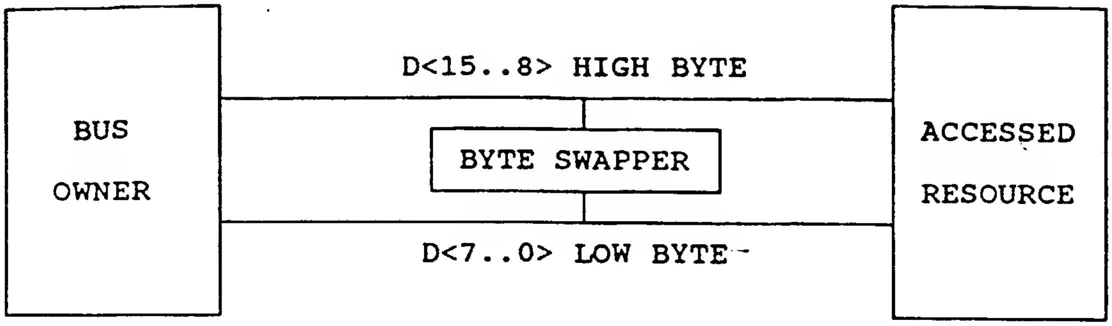


Figure 6.4.1 BYTE SWAPPER HARDWARE FOR AN ACCESS CYCLE

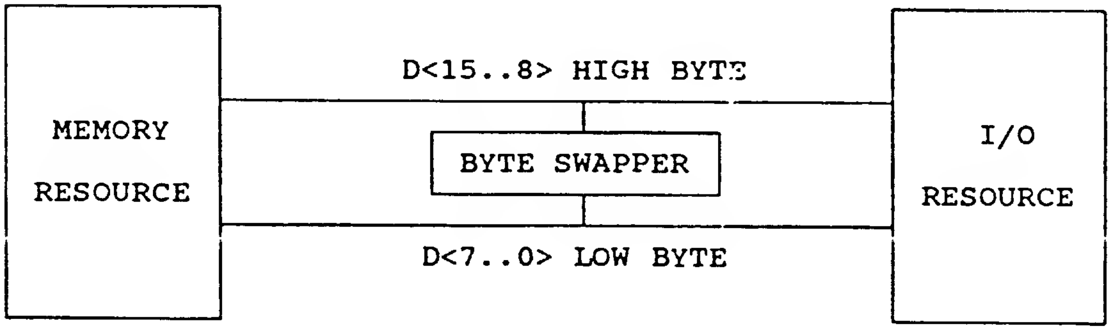


Figure 6.4.2 BYTE SWAPPER HARDWARE FOR DMA TRANSFERS

Table 6.4.1 summarizes the bytes swapped for access cycles. The byte swapping hardware allows bus owners of 16-bit data size to access resources of 8-bit data size. The swapping activity between the HIGH BYTES and the LOW BYTES is summarized in Table 6.4.1. H>L denotes HIGH BYTE lines driven onto the LOW BYTE lines by the Byte Swapper hardware; H<L denotes the opposite. HH denotes the HIGH BYTE is sent between the bus owner and accessed resource without swapping. The LL denotes that the LOW BYTE is sent between the bus owner and accessed resource without swapping.

BUS OWNER			ACCESSED RESOURCE		CYCLE COMPLETION		
DATA SIZE(1)	SBHE*	A0	DATA SIZE	XCS16*	DATA SIZE	PATH	
						RD	WR
8	1	0	8	1	8	LL	LL
	0	1	8	1	8	H<L	H>L
	1	0	16	0	8	LL	LL
	0	1	16	0	8	HH	HH
16	0	0	8	1	8	LL	LL
	0	0	16	0	16	HH/LL	HH/LL

Table 6.4.1 BYTES SWAPPED FOR AN ACCESS CYCLE

NOTE:

(1) The bus owner is actually 16-bits in size but can do an 8-bit access.

Table 6.4.2 summarizes the bytes swapped for a DMA transfer cycle. The byte swapping hardware allows memory resources of 16-bit data size to interact with I/O resources of 8-bit data size. The swapping activity between the HIGH BYTES and the LOW BYTES is summarized in Table 6.4.2. H>L denotes HIGH BYTE lines driven onto the LOW BYTE lines by the Byte Swapper hardware; H<L denotes the opposite. HH denotes the HIGH BYTE is sent between the memory resource and the I/O resource without swapping. The LL denotes that the LOW BYTE is sent between the memory resource and the I/O resource without swapping.

I/O RESOURCE	DMA		MEMORY RESOURCE		CYCLE COMPLETION		
DATA SIZE	SBHE*	A0	DATA SIZE	MCS16*	DATA SIZE	PATH	
						RD	WR
8	1	0	8	1	8	LL	LL
8	1	0	16	0	8	LL	LL
8	X	1	8	1	8	LL	LL
8	X	1	16	0	8	H>L	H<L
16	0	0	8	1		ILLEGAL	
16	0	0	16	0	16	HH/LL	HH/LL

Table 6.4.2 BYTES SWAPPED FOR A TRANSFER CYCLE

7.0 SIGNAL DESCRIPTIONS

This chapter lists and describes the seven signal groups that comprise the Intel ISA bus. The function of each signal is discussed in detail.

Each signal group has an [8] or [8/16] which denotes that this particular signal is available on an 8-bit only slot or an 8/16-bit slot, respectively.

7.1 SIGNAL GROUPS

The Intel ISA bus contains seven groups of signals: ADDRESS, DATA, CYCLE CONTROL, CENTRAL CONTROL, INTERRUPT, DIRECT MEMORY ACCESS (DMA), and POWER. The input and output direction designations for each signal are referenced to the bus owner.

7.1.1 ADDRESS SIGNAL GROUP

The address signal group consists of signals driven by the bus owner to specify data address.

A<19..00> [8][8/16]
Address signals are latched outputs driven by the bus owner. When the access is to the memory address space they represent the least significant 20 address bits and define a 1MB address space. When the access is to the I/O address space **A<15..0>** they contain the valid address and **A<19..16>** are undefined.

During refresh cycles **A<07..00>** contain the valid address **A<19..08>** are undefined and should be tristated by all resources that can drive them.

ADD-ON CARD DESIGN FOCUS

The add-on card must be the bus owner to enable the **MEMREF*** line. When the add-on card enables the **MEMREF*** line, the address lines are driven by the Refresh Controller; they must be tri-stated by the add-on card.

LA<23..17> [8/16]
Unlatched address signals are driven by the bus owner. When the Primary CPU is bus owner, the LA lines are valid when BUSALE is asserted, but are not valid for the complete cycle. When the DMA Controller is bus owner, the LA lines must be valid prior to MRDC* or MWTC* and remain valid the entire cycle. When the access is to the memory address space they represent the seven most significant address bits. When the access is to the I/O address space or during refresh cycles these lines are driven to logical "0".

During refresh cycles the unlatched address lines are undefined and should be tristated by all resources that can drive them.

ADD-ON CARD DESIGN FOCUS

When the add-on card is bus owner these lines must be valid prior to **MRDC*** or **MWTC*** and remain valid the entire cycle.
The add-on card must be the bus owner to enable the **MEMREF*** line.
When the add-on card enables the **MEMREF*** line the address lines are driven by the Refresh Controller; they must be tri-stated by the add-on card.

SBHE* [8/16].
System Bus High Enable is enabled by the Primary CPU to indicate a transfer of data on lines **D<15..8>**. **SBHE*** and **A0** are used to determine which bytes are being transferred over the bus, as shown in Figure 6.4 and Table 6.4.

SBHE* is not driven when the Refresh Controller is bus owner because no data swapping occurs; no actual data is read.

ADD-ON CARD DESIGN FOCUS

When the add-on card is bus owner **SBHE*** is used in the same fashion as it is used by the Primary CPU.
The **SBHE*** signal is tristated when the **MEMREF*** line is enabled by the add-on card that is bus owner.

BUSALE [8][8/16]
Bus Address Latch Enable is an address strobe driven by the Primary CPU to indicate when **LA<23..17>** are valid and can be latched. It also indicates when **SBHE*** and **A<19..00>** are valid.

When the DMA Controller is bus owner the **BUSALE** is driven to a logical "1" by the platform because **LA<23..17>** and **A<19..0>** are valid prior to it enabling the command lines. When the Refresh Controller is bus owner the **BUSALE** line is driven to logical "1" by the platform because **SA<19..00>** are valid prior it enabling the **MRDC*** and **MEMR*** lines.

ADD-ON CARD DESIGN FOCUS

When the add-on card is the bus owner, the **BUSALE** is driven to logical "1" by the platform for the entire time that it is bus owner. Thus **LA<23..17>** and **A<19..0>** must be valid prior the add-on card enabling the command lines.

When the Primary CPU is bus owner and it is accessing an add-on card **LA<23..19>** is valid only for a short time; **BUSALE** is used by the add-on card to latch the address. When any resource other than the Primary CPU is bus owner, the **BUSALE** line remains enabled. A proposed input address circuit design for the add-on card to accommodate both situations is shown in Figure 7.1.1.

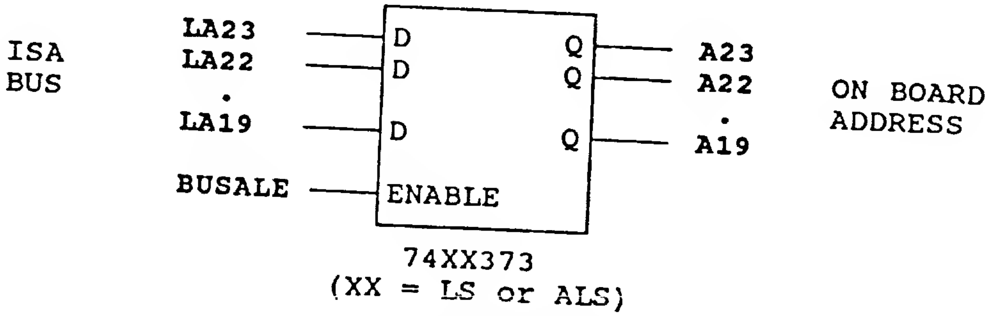


Figure 7.1.1 INPUT ADDRESS CIRCUIT

AEN [8][8/16]
Address Enable is enabled when the DMA Controller is the bus owner to indicate that a DMA transfer will occur. The enabling of the **AEN** line indicates to I/O resources to ignore the address lines which contain the memory address during DMA transfers. This line is disabled by the DMA controller when the Primary CPU or Refresh Controller are bus owners.

ADD-ON CARD DESIGN FOCUS

If an add-on card enables the **SECMAST*** line, then the **AEN** is disabled by the DMA Controller to allow the add-on cards' access to the I/O address space. When the AEN line is enabled, the add-on card should not be driving the bus unless it is part of the DMA transfer.

D<07..00> [8][8/16]
D<15..08> [8/16]
D15 is the most significant bit and **D0** is the least significant bit. All 8-bit resources can connect only to the least significant eight data lines, **D<07..00>**. To support communication between 16-bit bus owners and 8-bit resources, data swapping is supported by the BYTE SWAPPER circuitry on the baseboard.

Figure 6.4 and Table 6.4 summarize the byte swapping function.

ADD-ON CARD DESIGN FOCUS

When the **MEMREF*** line is enabled by the add-on card, the data lines must be tristated by the add-on card because no actual data is transferred during a refresh cycle.

7.1.2 CYCLE CONTROL SIGNAL GROUP

This group of signals controls the duration and type of cycles. The group consists of six command signals, two ready signals, and three signals which specify the cycle size and type.

The command signals define the address space (memory or I/O) and the data direction (read or write). The ready signals modify the command pulse widths to lengthen or shorten the default cycle timings.

MRDC* [8/16]
MEMR* [8][8/16]
The Memory Read command (**MRDC***) is enabled by the bus owner to request a memory resource to drive the data bus with the contents of the memory location specified by **LA<23..17>**, **A<19..00>**. The System Memory Read command (**MEMR***) is identical in function to **MRDC***, except it is asserted only when the memory address is within the first 1MB. The **MEMR*** signal is driven by the platform and is derived from the **MRDC*** signal; thus it lags the **MRDC*** signal by 10ns or less.

ADD-ON CARD DESIGN FOCUS

When the add-on card is bus owner it can only begin a bus cycle by enabling MRDC*; MEMR* is enabled by the platform if the access is in the first 1MB of the memory address space.
When the add-on card enables the **MEMREF*** line it must tri-state the **MRDC*** line because this line will be enabled by the Refresh Controller.

MWTC* [8/16]
MEMW* [8][8/16]
The Memory Write command (**MWTC***) is enabled when the bus owner is driving the data bus with data memory address location specified by **LA<23..17>** and **A<19..00>**. The System Memory Write (**MEMW***) is identical in function to **MWTC***, except it is asserted only when the memory address is witnin the first 1MB. The **MEMW*** signal is driven by the platform and is derived from the **MWTC*** signal; thus it lags the **MWTC*** signal by 10ns or less.

ADD-ON CARD DESIGN FOCUS

When the add-on card is bus owner, it can only begin a bus cycle by enabling **MRDC***; **MEMW*** is enabled by the platform if the access is in the first 1MB of the memory address space.

When the add-on card enables the **MEMREF*** line, it must tri-state the **MWTC*** line.

IORC* [8][8/16]
I/O Read command (**IORC***) is enabled by the bus owner to request the accessed I/O resource to drive the data bus with the contents of the address specified by **A15..00>**.

ADD-ON CARD DESIGN FOCUS

When the add-on card enables the **MEMREF*** line, it must tri-state the **IORC*** line.

IOWC* [8][8/16]
I/O Write command (**IOWC***) is enabled when the bus owner is driving the data bus for I/O address location specified by **A<15..00>**.

ADD-ON CARD DESIGN FOCUS

When the add-on card enables the **MEMREF*** line, it must tri-state the **IOWC*** line.

MCS16* [8/16]
Memory Cycle Select 16-bits is enabled by the accessed 16-bit memory resource to indicate to the bus owner that a 16-bit cycle may be executed. If this line is not enabled, then only an 8-bit cycle may be executed. The accessed resource generates **MCS16*** based on a decode of **LA<23..17>**.

NOTE

The DMA Controller and Refresh Controller will ignore **MCS16*** during DMA transfers and refresh cycles, respectively.

ADD-ON CARD DESIGN FOCUS

When the add-on card is in the access memory mode, it must enable the MCS16* line if the memory on the card is 16-bit data size. When the add-on card is in the bus owner mode, A<15..00> may contain a value which may accidentally decode to a value that enables IOCS16*; it must ignore this signal during memory operations.

IOCS16* [8/16]
I/O Cycle Select 16-bits is enabled by the accessed 16-bits I/O resource to indicate to the bus owner that a 16-bit cycle may be executed. If this line is not enabled, then only an 8-bit cycle may be executed. The accessed resource generates IOCS16* based on a decode of A<15..00>.

NOTE

The DMA Controller and Refresh Controller will ignore IOCS16* during DMA transfers and refresh cycles, respectively.

ADD-ON CARD DESIGN FOCUS

When the add-on card is in the access memory and I/O mode it must enable this line if the memory on the card is 16-bit data size. When the add-on card is in the bus owner mode LA<23..17> it may accidentally decode to a value that enables MCS16*; it must ignore this signal during I/O operations.

IOCHRDY [8][8/16]
I/O Channel Ready is an asynchronous signal driven by the accessed resource. It is disabled to force the bus owner to lengthen the bus cycle by inserting an integral number of wait states. When the bus owner is the PRIMARY CPU or add-on card, each wait state is an Intel ISA bus SYSCLK period, 125ns for an 8MHz clock rate. When the DMA Controller is bus owner, each wait state is two Intel ISA bus SYSCLK periods, or 250ns for an 8MHz clock rate. IOCHRDY is ignored during 0 wait state cycles.

NOTE

For DMA transfers I/O does not drive this line because the enabling of DRQ* by the I/O resource can only be done when valid data can be received or sent. Only the memory resource involved with the DMA transfer can enable this signal.

CAUTION

IOCHRDY must not be disabled for longer than 15 microseconds or refresh cycles will be "missed" and DRAM data failure may occur.

ADD-ON CARD DESIGN FOCUS

When the add-on card is in the bus owner it must receive **IOCHRDY** from the resource being accessed. When it is in the other modes it must drive this signal when it is ready to complete the cycle.

CAUTION

When some add-on cards are bus owners they ignore the **IOCHRDY** signal and run as default 8-bit or 16-bit normal type access memory cycle. Any add-on card that returns an **IOCHRDY** signal must determine if an accessing bus owner will be an add-on card that supports this function and thus allows the cycle to be lengthened.

SRDY* [8][8/16]

0 wait state is the only synchronous signal line on the Intel ISA bus. It is enabled by the accessed resource to request the Primary CPU or the add-on card to complete the current bus cycle without any further wait states.

NOTE

Even though this signal line is on the [8] slots, it is not used. It can only be used to access 16-bit memory resources installed in [8/16] slots when the Primary CPU or add-on card are bus owners. This signal is ignored during an I/O access or when the DMA Controller or Refresh Controller is the bus owner.

ADD-ON CARD DESIGN FOCUS

When the add-on card is in the bus owner it must receive **SRDY*** from the resource being accessed in order to execute 0 wait state access cycle. When it is in the access memory mode it must drive this signal when it can support a 0 wait state access cycle.

CAUTION

When some add-on cards are bus owners they ignore the **SRDY*** signal and run as default 8-bit or 16-bit a Normal or Ready type access memory cycle.

MEMREF* [8][8/16]
MEMREF* is enabled to indicate a DRAM refresh cycle. This signal is enabled by the Refresh Controller when it is bus owner.

ADD-ON CARD DESIGN FOCUS

When an add-on card is bus owner it can enable this signal line to request a refresh cycle. The cycle will be executed at that time by the Refresh Controller even though it is not bus owner.

7.1.3 CENTRAL CONTROL SIGNAL GROUP

The central control group consists of special timing, control, and error signals. The function of these signals is as follows.

SECMAST* [8/16]
This signal line can only be driven by the add-on card that has been granted bus ownership.

CAUTION

If **SECMAST*** is enabled for longer than 15 microseconds, the add-on card must initiate refresh cycles by enabling the **MEMREF*** line.

ADD-ON CARD DESIGN FOCUS

SECMAST* is enabled by an add-on card to become bus owner after receiving the appropriate **DACK#*** from the DMA Controller. After **SECMAST*** is enabled, the add-on card must wait at least one **SYSCLK** period before driving the address and data group signals, and it must wait at least two **SYSCLK** periods before driving the cycle control group signals.

IOCHCK* [8][8/16]
I/O Channel Check may be enabled by any resource to signal an error condition that cannot be corrected, such as a memory parity error. It must be enabled for at least 15ns. If the DMA or Refresh Controller are the bus owner, then this signal will be latched by platform hardware when enabled but not acted upon until the Primary CPU becomes bus owner.

ADD-ON CARD DESIGN FOCUS

If the add-on card is the bus owner when this signal is enabled, then the error will be latched by platform hardware when enabled but will not be acted upon until the Primary CPU becomes bus owner.

RSTDEV [8][8/16]
Reset Device is asserted by the Primary CPU to initialize all agents on the Intel ISA bus after power-up or during a low-voltage condition. The minimum enable time to ensure for reset is 1 millisecond.

ADD-ON CARD DESIGN FOCUS
The add-on card must immediately enter and remain in the reset condition when this signal is enabled. See Section 5.3 for more information about the reset condition.

SYSCLK [8][8/16]
System Clock has a frequency of 8MHz with a 50% duty cycle, and it is driven by the baseboard. Bus cycle times are directly proportional to the clock period but are not synchronous; except for a 0 wait state cycle.

ADD-ON CARD DESIGN FOCUS
When the add-on card is bus owner it can use this clock to determine the length of a cycle. But in that this clock is only synchronized to SRDY*, any clock can be used for on-card timing.

84OSC [8][8/16] 69.84ns
84OSC is a clock signal driven by the platform with a frequency of 14.31818 MHz (+/- 5- ppm) with a 45-55% duty cycle. 84OSC is not synchronous to either **SYSCLK** or any other signals on the Intel ISA bus, so it must not be used in applications which require synchronization to the bus. The presence of this particular frequency relates to the fact that it is a low-cost crystal from the color television industry. It is divided by 12 in PC's and fed into the 8254 timer.

7.1.4 INTERRUPT SIGNAL GROUP

The interrupt signal group consists of a set of signals that can be used by a resource to obtain interrupt service from the Primary CPU.

NOTE
The actual interrupt signals input to an Interrupt Controller (Intel 8259A). It is accessible by all bus owners via the I/O address, but for software compatibility only the Primary CPU should service the Interrupt Controller.

IRQ<15,14,12,11,10> [8/16]
IRQ<09,07..03> [8]
An interrupt may be requested by a platform resource or add-on card by enabling an **IRQ** line. The line must remain enabled until the interrupt is acknowledged by accessing the interrupting resource on the add-on card by the Primary CPU.

ADD-ON CARD DESIGN FOCUS
The interrupt lines are edge triggered lines and driven by TTL drivers. Thus, an add-on card must allow the user to select which ISA bus **IRQ** line is driven by the add-on card at the time of installation.

7.1.5 DIRECT MEMORY ACCESS SIGNAL GROUP

These signals support DMA transfer cycles and transfer of bus ownership to a platform resource or an add-on card.

NOTE
DMA channels <3..0> can only support 8-bit data size transfer cycles. DMA channels <7..5> can only support 16-bit data size transfer cycles.

DRQ<7..5,0> [8][8/16]
DRQ<3,2,1> [8]
DMA Request lines are enabled by a platform resource or add-on card to request DMA service or bus ownership. The **DRQ** line is enabled until the corresponding **DACK*** line is enabled by the DMA Controller.

ADD-ON CARD DESIGN FOCUS
The **DRQ** lines are driven by TTL drivers. Thus, an add-on card must allow the user to select which ISA bus **DRQ** line is driven by the add-on card at the time of installation and tri-state the others.

DACK<7..5,0>* [8][8/16]
DACK<3,2,1>* [8]
DMA Acknowledge lines are enabled by the DMA Controller to acknowledge DMA requests DRQ<7..5,3..0>. The enabling of the **DACK*** indicates that a transfer cycle will begin or an add-on card can become bus owner.

TC [8][8/16]
Terminal Count is enabled by the DMA Controller when any one of its DMA channels has reached its terminal count, indicating the end of

the DMA transfer.

7.1.6 POWER GROUP

The Intel ISA bus provides DC power at +5 volts, -5 volts, +12 volts, -12 volts, and 0 volts (Ground). All of the power lines are on the 8-bit connector except for a single +5 volts and a single ground line. These later lines are 16-bits extended connector.

The maximum amount of current for each voltage that an add-on card slot can support is summarized in Table 7.1.6.

CAUTION

The amount of current allowed for each slot as listed in Table 7.1.6 does not ensure that current will be available from the system. No AT-compatible system power supply provides enough power to allow all the card slots to draw maximum current. Refer to the System Technical Reference Manual to determine available power for add-on cards.

VOLTAGE	[8]	[8/16]
+5	3.0 AMPS	4.5 AMPS
+12	1.5 AMPS	1.5 AMPS
-5	1.5 AMPS	1.5 AMPS
-12	1.5 AMPS	1.5 AMPS

Table 7.1.6 MAXIMUM CURRENT

8.0 BUS CYCLES

ISA bus cycles are asynchronous in that bus activities are independent of the **SYSCLK**. Some signals enable and disable at any time; others respond within minimum and maximum times of the other signals being enabled or disabled. The only exception is the **SRDY*** signal which is synchronized **SYSCLK**.

There are four distinct bus cycles: Access, Transfer, Refresh, and Bus Ownership. An Access cycle occurs when the Primary CPU or add-on card is reading or writing data with another resource. A Transfer cycle occurs when the DMA Controller is bus owner and data flows between memory and I/O resources. The Refresh cycle is only executed by the Refresh Controller to refresh DRAM. The Bus Ownership cycle is executed by add-on cards to obtain bus ownership.

The exact structure of a given cycle is dependent on the bus owner and the resources involved in the cycle; the major distinction between different types of cycles is the length. There are three types of Access cycles: a minimum cycle called 0 wait state, a slightly longer cycle called Normal, and a Ready cycle. There are two types of Refresh and Transfer cycles: a default type called Normal, and a longer one called Ready.

Listed below are key features of the different cycles. Refer to Section 9 for more detailed timing diagrams.

8.1 ACCESS CYCLES ... INTRODUCTION

The Primary CPU begins an Access cycle pulsing **BUSALE** to indicate valid address on lines **A<19..00>** and for bus resources to latch address lines **LA<23..17>**. The accessed resource responds by enabling **MCS16*** or **IOCS16*** to establish a 16-bit cycle; if these signals are not enabled, the cycle is completed as a default 8-bit cycle. The Primary CPU also drives command lines **MRDC***, **MWTC***, **IORD***, and **IOWR*** to establish the address space and data direction. If the access is in the first 1MB of the memory address space, the platform hardware also enables **MEMR*** and **MEMW***. The accessed resource responds with **SRDY*** or **IOCHRDY** within a specific time to indicate to the Primary CPU the type of access cycle.

ADD-ON CARD DESIGN FOCUS

The add-on card as bus owner begins access cycles by driving the address lines. The **BUSALE** is not pulsed by the add-on card; it is enabled as a constant logical "1" by platform resources whenever the Primary CPU is not the bus owner. Thus, both **A<19..00>** and **LA<23..17>** must be valid prior to enabling the command lines and remain valid the entire cycle. The add-on card must be able to complete the cycle as 8- or 16-bits as specified by **MCS16*** or **IOCS16***.

8.1.1 ACCESS CYCLE ... 0 WAIT STATE TYPE

The 0 wait state type of access cycle is the fastest possible that can be executed. It can only be run when a Primary CPU or add-on card accesses 16-bit memory resources. The bus owner drives the **LA<23..17>** address lines to select a specific 128KB block. If the **MCS16*** is not enabled by the accessed resource, then the cycle must be completed as 8-bits. The only cycle types allowed for 8-bit are Normal and Ready; thus 0 wait state can't be executed. If the **MCS16*** is enabled by the accessed resource, then the accessed resource must enable the **SRDY*** signal within a specific time of the bus owner enabling the **MRDC*** or **MWTC*** lines to execute the cycle as a 0 wait state. If the **SRDY*** is not enabled, then the cycle is completed as either a Normal or Ready type cycle.

The enabling of **SRDY*** signal line does not require **IOCHRDY** to be enabled, in fact it is ignored by the bus owner.

NOTE

The **SRDY*** signal is only signal referenced to the system clock.

ADD-ON CARD DESIGN FOCUS

The add-on card as bus owner executes the 0 wait state type access cycle in the same fashion as the Primary CPU. Please see the **CAUTION** note in Section 7.1.2 on the **SRDY*** signal line description.

8.1.2 ACCESS CYCLE ... NORMAL TYPE

The Normal type access cycle can be run when a Primary CPU bus owner accesses 8- or 16-bits data size memory or I/O resources. The Primary CPU enables **MRDC***, **MWTC***, **IORC***, and **IOWC***. The accessed resource in response enables the **IOCHRDY** line within a specific time; otherwise the cycle becomes a Ready type access cycle. The enabling of **IOCHRDY** forces the bus owner to complete the cycle within in a fixed period of time. The fixed period of time is a multiple of the **SYSCLK** period, even though it is not synchronized to it.

The length of time that **MRDC***, **MWTC***, **IORC***, and **IOWC*** are enabled controls the length of the Normal type cycle. The length of these command lines are dependent on the data size and the address space of the access.

ADD-ON CARD DESIGN FOCUS

When the add-on card is bus owner it will execute the Normal type access cycle in the same fashion as the Primary CPU.

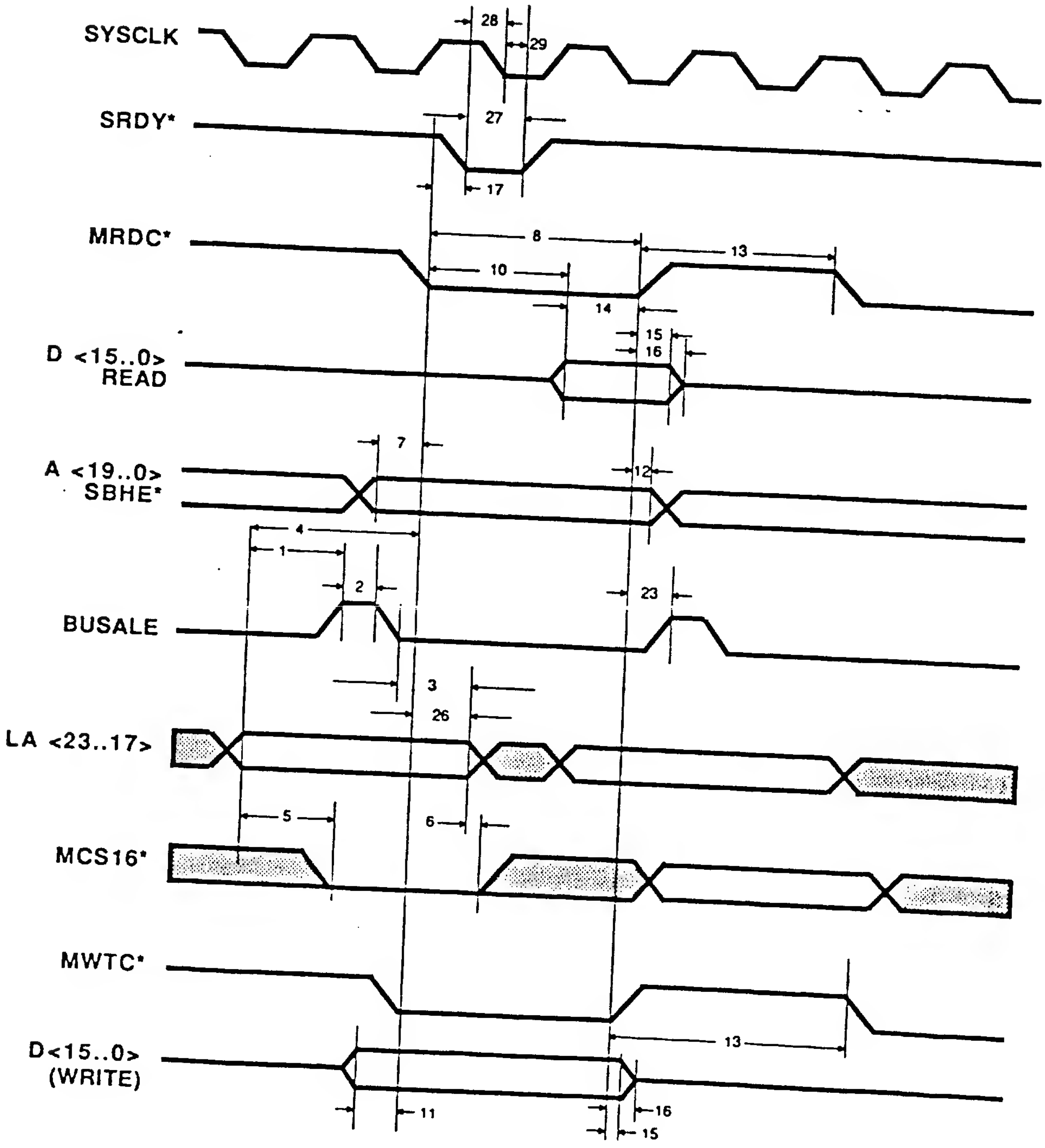
8.1.3 ACCESS CYCLE ... READY TYPE

The Ready type access cycle type is executed by the Primary CPU. The bus owner executes a Ready type access cycle if the **IOCHRDY** signal is not enabled within a specific time of a command line being enabled. The bus owner continues to enable the command line until the **IOCHRDY** signal line is enabled by the accessed resource; upon the enabling of **IOCHRDY** the bus owner disables the command line to complete the cycle.

The amount that the command line is lengthened is a multiple of the bus clock, even though none of the functions are synchronized to it.

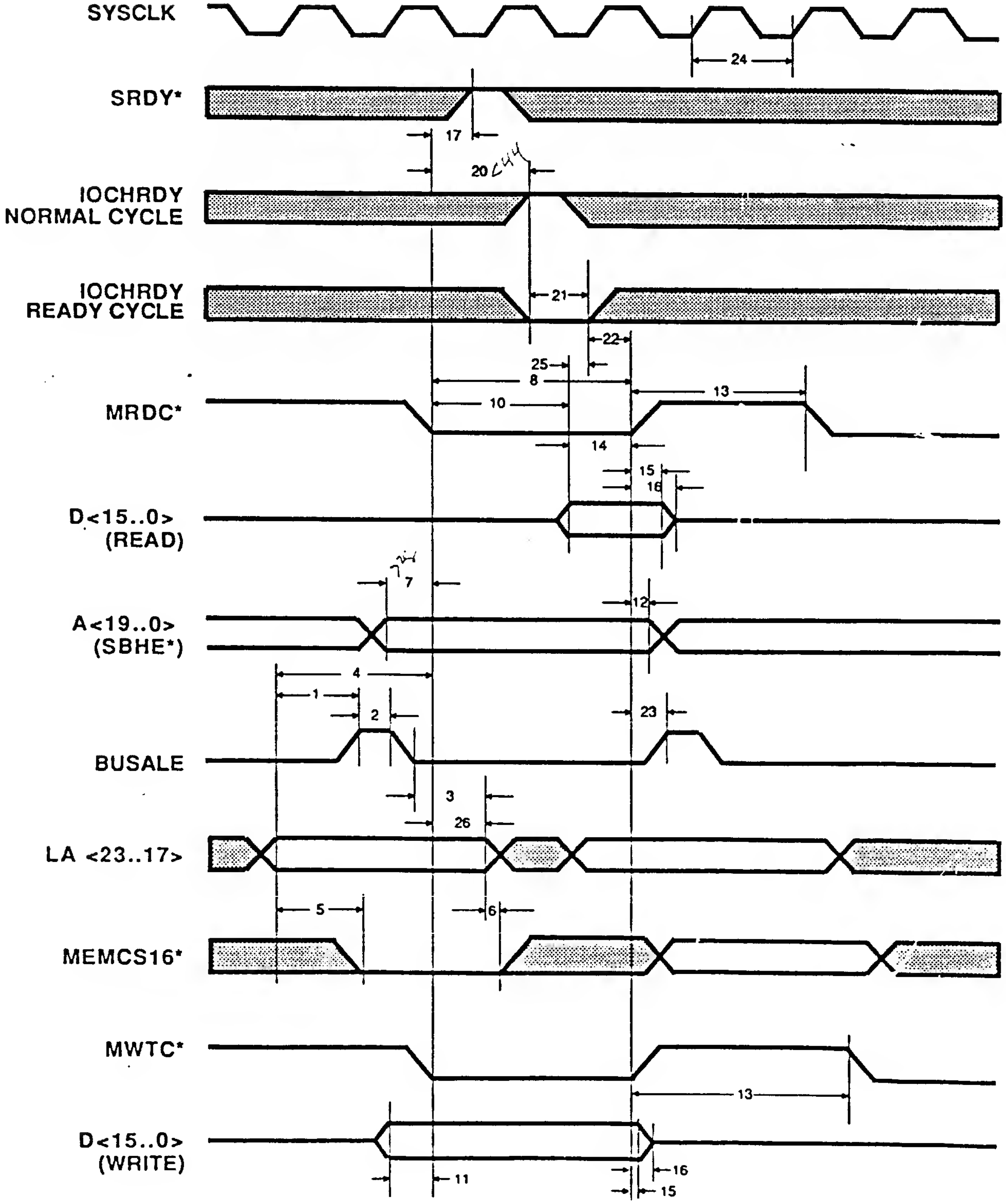
ADD-ON CARD DESIGN FOCUS

The add-on card as bus owner executes the ready type access cycle in the same fashion as the Primary CPU. Please see the **CAUTION** note in Section 7.1.2 on the **IOCHRDY** signal line description.



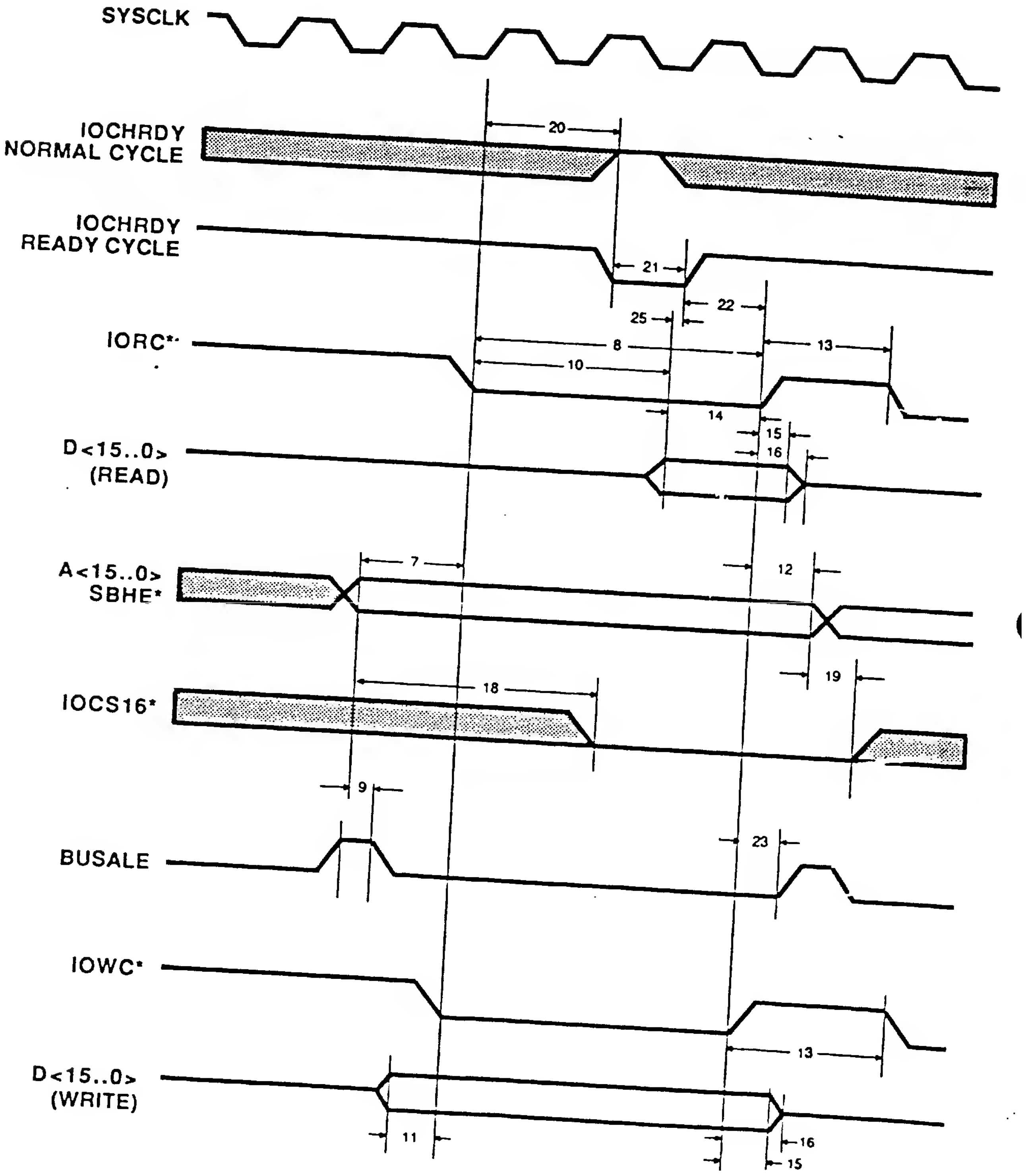
O WAIT STATE TYPE

Figure 8.1.1 16-BIT MEMORY READ & WRITE CYCLE



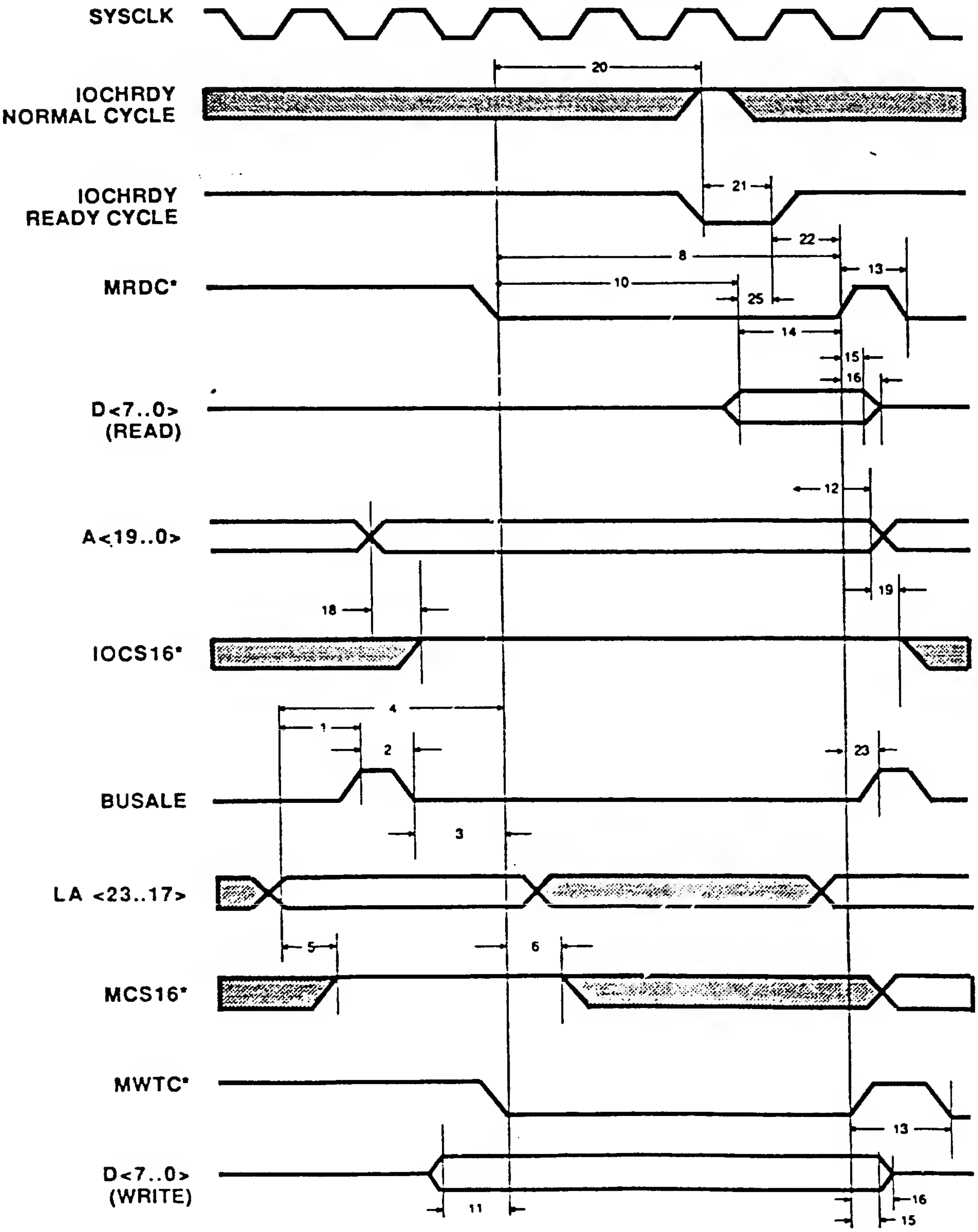
NORMAL & READY TYPE

Figure 8.1.2 16-BIT MEMORY READ & WRITE CYCLE



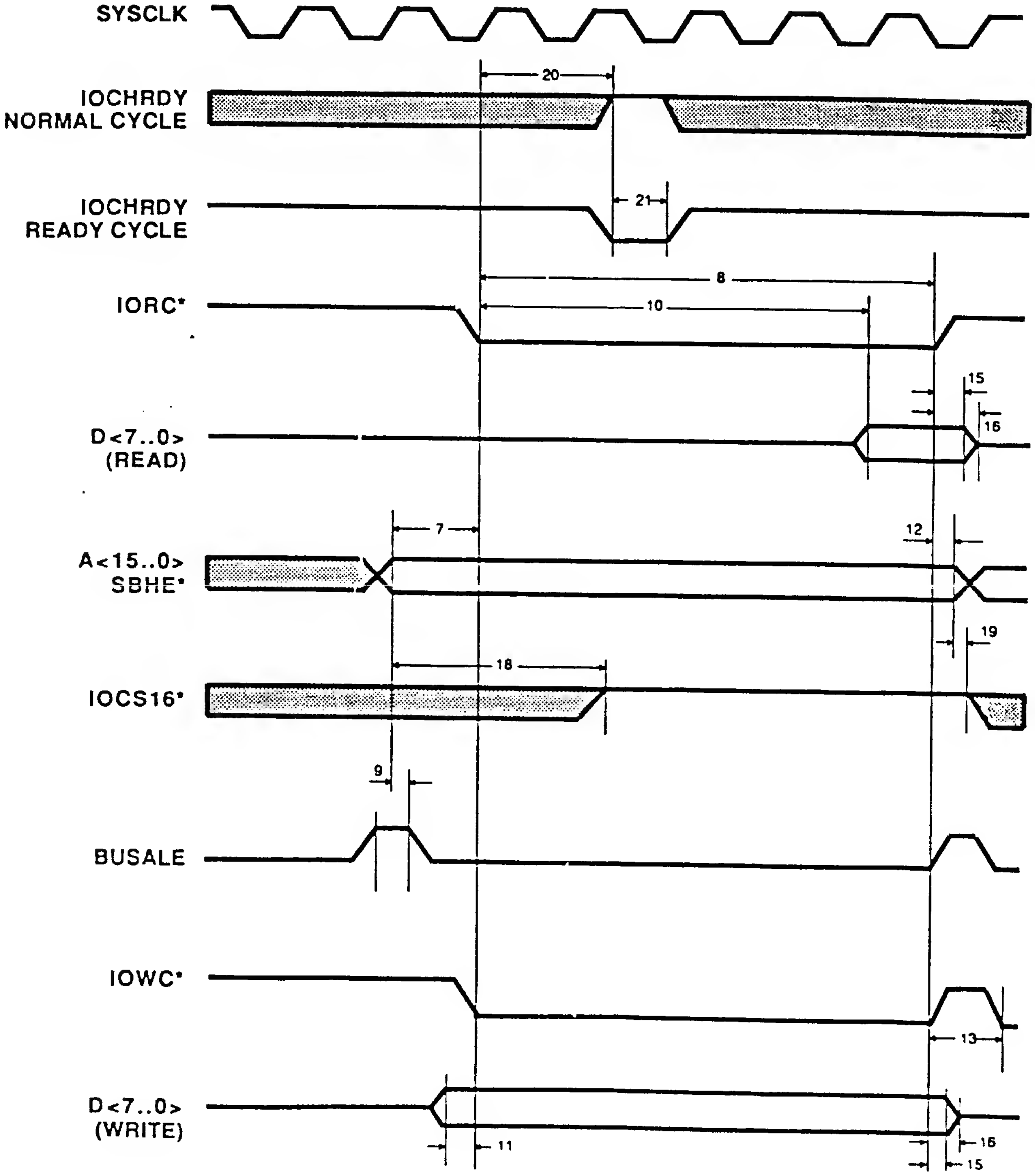
NORMAL & READY TYPE

Figure 8.1.3 16-BIT I/O READ & WRITE CYCLES



NORMAL & READY TYPE

Figure 8.1.4 8-BIT MEMORY READ & WRITE CYCLE



NORMAL & READY TYPE

Figure 8.1.5 8-BIT I/O READ & WRITE CYCLE

8.2 REFRESH CYCLES ... INTRODUCTION

The Refresh Controller becomes bus owner by two methods. Once 15 usec. has elapsed since the last refresh cycle, the Refresh Controller tries to become bus owner. If the Primary CPU is the current bus owner, ownership is immediately transferred to the Refresh Controller. If the DMA controller is bus owner, then bus ownership is not transferred until the pending DMA transfers are complete.

ADD-ON CARD DESIGN FOCUS

When an add-on card is the bus owner, it can request the Refresh Controller to run a refresh cycle by enabling the MEMREF* signal line.

The following signal lines have unique interpretations during Refresh cycles:

MEMREF* Enabling of the Refresh line begins the Refresh cycle.

ADDRESS The Refresh Controller drives SA<7..0> with the refresh address; the other address lines are undefined.

MRDC* MRDC* is enabled by the Refresh Controller. MEMR* will be enabled platform hardware.

D<15 ..00> The data lines are ignored by the Refresh Controller. They should not be driven by any resource.

SRDY*
MCS16*
IOCS16* These lines are ignored by the Refresh Controller.

8.2.1 REFRESH CYCLES ... NORMAL TYPE

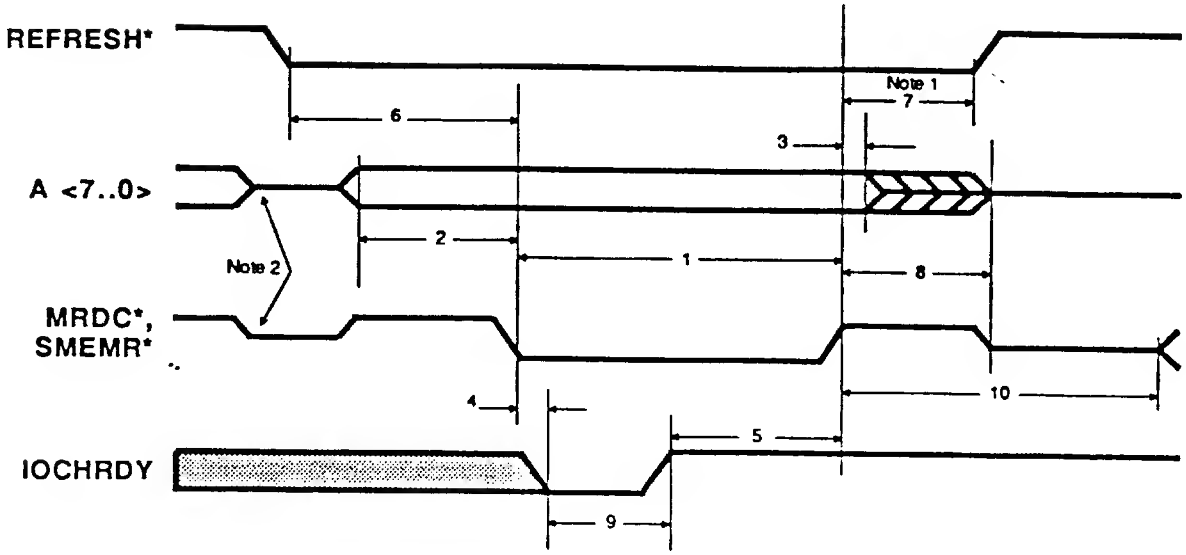
The Normal type Refresh cycle begins by the Refresh Controller enabling MRDC* and the accessed resource in response enables the IOCHRDY line within a specific time; otherwise the cycle becomes a Ready type transfer cycle.

The length of time that MRDC* is enabled determines the length of the cycle.

8.2.2 REFRESH CYCLES ... READY TYPE

The Ready type Refresh cycle is executed by the Refresh Controller. The Refresh Controller executes a Ready type access cycle if the **IOCHRDY** signal is not enabled within a specific time of the **MRDC*** a line being enabled. The Refresh Controller continues to enable the command line until the **IOCHRDY** signal line is enabled by all memory resources; upon the enabling of **IOCHRDY** the Refresh Controller disables the command line to complete the cycle.

The amount that the cycle is lengthened is a multiple of the bus clock, even though none of the functions are synchronized to it.



Notes.

1) The agent may exceed the maximum REFRESH* hold time in order to conduct another refresh cycle.

2) The secondary requesting agent, if the current bus owner must tri-state the address and command signals prior to driving REFRESH* = L.

Figure 8.2 NORMAL & READY TYPE REFRESH CYCLE

8.3 DMA TRANSFER CYCLES ... INTRODUCTION

The DMA Transfer cycle is unlike the access cycles executed by other bus owners. A DMA transfer cycle is run in response to a **DREQ*** line being enabled. The data size must be appropriate for the DMA channel; channels 0-3 are defined for 8-bit transfers, and channels 5-7 are defined for 16-bit transfers. **MCS16*** and **IOCS16*** are ignored by the DMA Controller but **MCS16*** is used by the data swapper. Also, the **SRDY*** is ignored because the 0 wait state type of access cycle is not supported by DMA transfer cycles.

The DMA transfer cycles are only between memory and I/O resources. The address lines driven by the DMA Controller contain the address of memory resource; there is no address specified for the I/O resource. The actual transfer is fly-by in nature; the data source places the data on the bus at the same time the data destination retrieves the data. The read and write command lines are enabled as pairs to support the appropriate data direction for the source and destination. The read command line is enabled prior to the write command to ensure no buffer fights between the data buffers of the two resources.

A resource requests a DMA transfer by enabling the **DRQ*** line of the appropriate channel. If the Primary CPU is bus owner, then bus ownership will be transferred to the DMA Controller. The DMA Controller notifies an I/O resource that it is part of the DMA transfer by enabling the appropriate **DACK*** line. In that the value of the address lines is for the memory resource, the I/O resource must provide and receive data based entirely on **IOWC***, **IORC***, and **DACK***.

The DMA Transfer cycle begins by the DMA Controller enabling the **DACK*** line of the appropriate channel and **AEN**. The enabling of the **AEN** line indicates to all resources that the address and control lines will be driven by DMA Controller and not the Primary CPU, Refresh Controller, or add-on card. The appropriate command lines are enabled and the DMA Controller monitors the **IOCHRDY**.

NOTE

The data to be written to a memory or I/O resource during an access cycle must be valid prior to the enabling of the write command line. In a DMA transfer cycle, the data being written must be valid prior to the disabling of the write command line.

8.3.1 DMA TRANSFER CYCLES ... NORMAL TYPE

The Normal type transfer cycle is executed by the DMA Controller for 8- or 16-bit data size transfers. The DMA Controller enables **MRDC***, **MWTC***, **IORC***, and **IOWC***. The accessed memory resource in response enables the **IOCHRDY** line within a specific time; otherwise the cycle becomes a Ready type transfer cycle. The enabling of **IOCHRDY** forces the DMA Controller to complete the cycle within in a fixed period of time. The fixed period of time is a multiple of the **SYSCLK** period even though it is not synchronized to it.

The length of time that **MRDC***, **MWTC***, **IORC***, and **IOWC*** are enabled determines the length of the cycle. The length of these command lines are also dependent on the data sizes for different address spaces.

8.3.2 DMA TRANSFER CYCLES ... READY TYPE

The Ready type DMA transfer cycle type is executed by the DMA Controller in the same fashion as the Normal type cycle. It is distinguished from the Normal type in that the **IOCHRDY** signal is not enabled within a specific time of the command lines being driven. The DMA Controller continues to enable the command line until the **IOCHRDY** signal line is enabled by the I/O resource; upon the enabling of **IOCHRDY**, the DMA Controller disables the command line to complete the cycle.

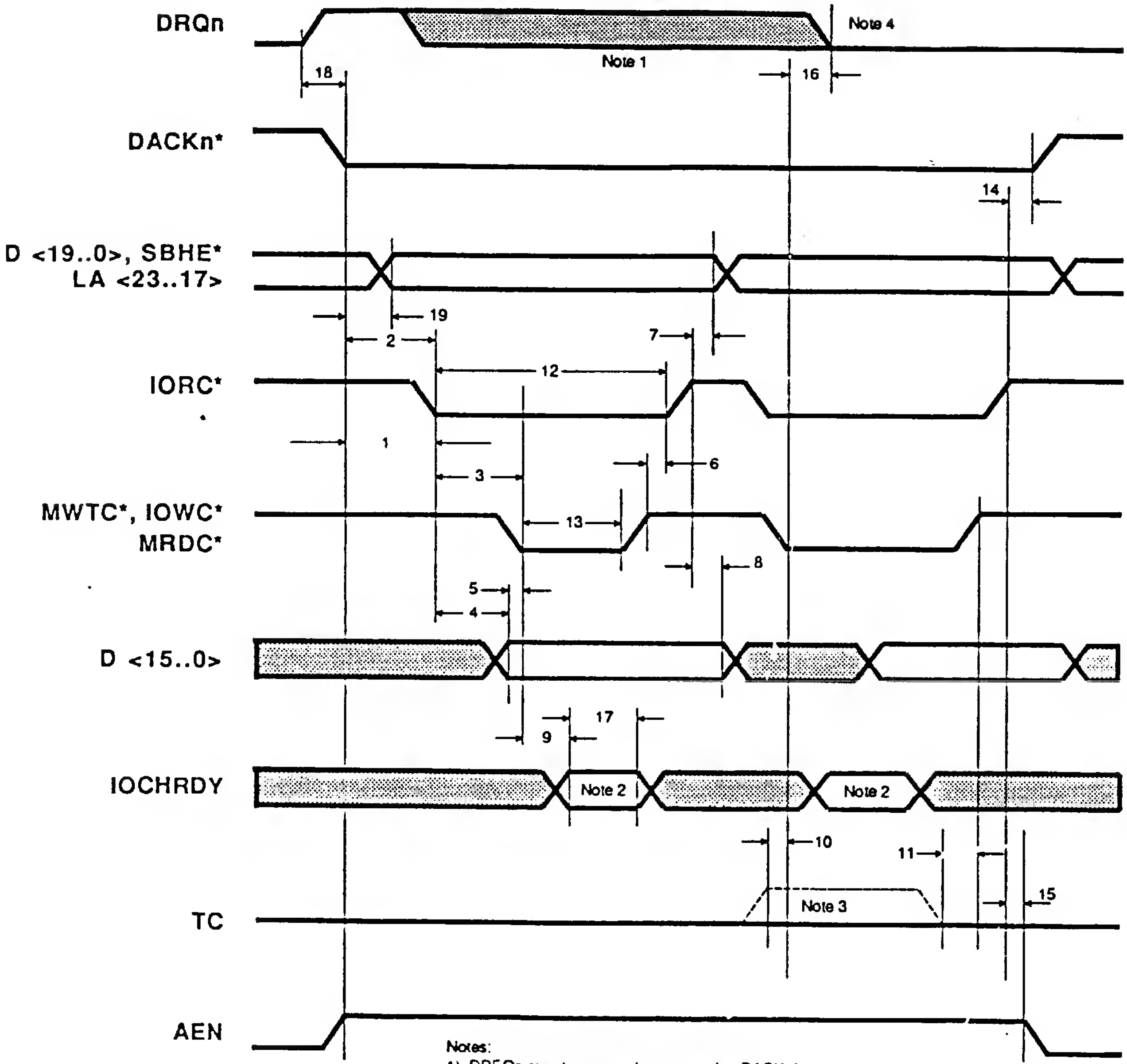
The amount that the cycle is lengthened is a multiple of two **SYSCLK** periods, even though none of the signals are synchronized to it.

NOTE

The address lines **LA<23..17>** during an access cycle must be latched by the accessed resource. In a transfer cycle these address lines are held valid for the entire transfer cycle.

CAUTION

The DMA channels that are to be used by add-on cards to obtain bus ownership must be programmed in the cascade mode.



Notes:

1) DREOn may be negated any time after DACKn* during block mode DMA cycles.

2) IOCHRDY assertion to insert additional wait states is optional. Additional bus wait states are added in units of two bus clocks.

3) The DMA controller activates TC during the last transfer of a DMA request.

4) DMA transfers may be broken up into multiple back-to-back cycles where the DMA controller removes DACKn and optionally gives up the bus to allow higher priority cycles to occur In this case DACK will be negated even though DREO is still asserted.

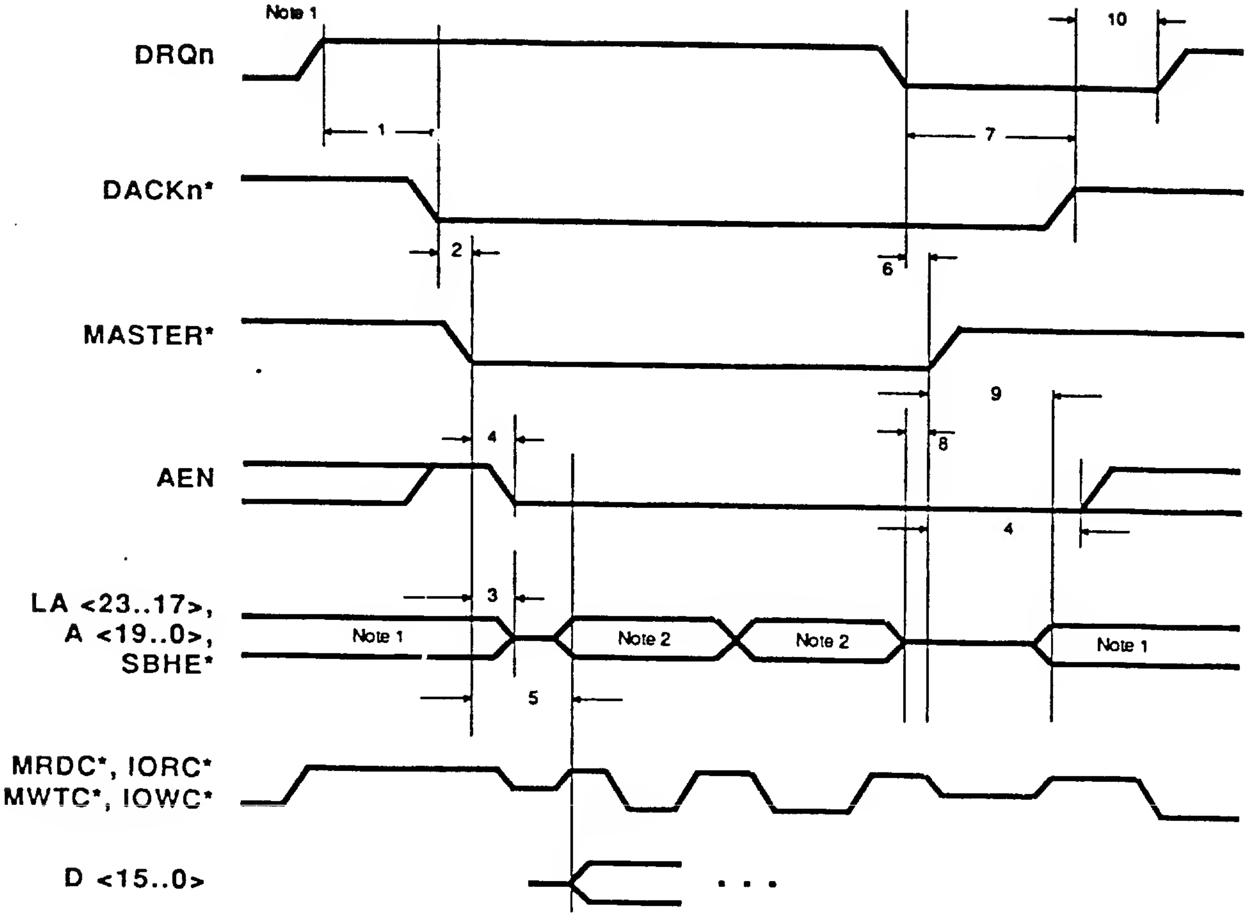
NORMAL & READY TYPE

Figure 8.3 DMA TRANSFER CYCLES

8.4 BUS OWNERSHIP CYCLES

Any add-on card that has a 16-bit data size (i.e., installed into an [8/16] slot) can become bus owner. An add-on board begins the request to become bus owner by enabling the **DRQ*** line of a DMA channel that has been programmed for the cascade mode. A DMA channel programmed in the cascade mode assumes that all cycle control is to be done externally by another resource; in this case, an add-on card. The DMA Controller responds by enabling the associated **DACK*** line; the add-on card in response enables the **SECMAST*** line.

After enabling the **SECMAST*** line, the add-on card must wait a minimum time before beginning an active access cycle.



NOTES:

1) DMA controller

2) Add-in card

Figure 8.4 BUS OWNERSHIP CYCLES

9.0 BUS CYCLE TIMING SPECS.

The following tables provide the specific timings for the bus cycles outlined in Section 8. All of the times in the tables are measured at the connector of the resource listed at the top of each column. A time of 0 - 11ns has been included in the times to cover settling times for a signal transversing the bus once. In some cases a signal is returned to a resource that was also the source of the signal that the returned signal is referenced. The referenced signal must first transverse the bus before the returned signal transverses the bus; in these cases 0 - 22ns has been included in the times. The settling time of "0" is the minimum possible time in theory; its use in timing calculations is dependent on which signals are being referenced.

NOTE

The timings only reference MRDC* or MWTC* and not MEMR* or MEMW*. MEMR* and MEMW* occur a minimum of 0ns and a maximum of 10ns relative MRDC* and MWTC* when the Primary CPU, DMA Controller, or Refresh Controller is bus owner. The maximum time increases to 32ns when an add-on card is bus owner (32 = 22 + 10).

Table 9.0.1 0 WAIT STATE, NORMAL, & READY 16-BIT MEMORY AND NORMAL & READY 16-BIT I/O NORMAL & READY 8-BIT MEMORY & I/O

Param	DESCRIPTION (Times in NANOSECONDS)	BUS OWNER		ACCESSED RESOURCE	
		Min	Max	Min	Max
1	LA<23..17> setup to BUSALE	61		50	
2	BUSALE pulse width	61		50	
3	LA<23..17> hold from BUSALE	26		15	
4	LA<23..17> setup to 16-bit memory command (1)	120		109	
5	MCS16* valid from LA<23..17>		102		66
6	MCS16* hold from LA<23..17>	0		0	

Table 9.0.1 (continued)

Param	DESCRIPTION (Times in NANOSECONDS)	BUS OWNER		ACCESSED RESOURCE	
		Min	Max	Min	Max
7a	A<19..00> setup to 16-bit memory command	39		28	
b	A<19..00> setup to 16-bit I/O or 8-bit command	102		91	
c	SBHE* setup to 16-bit memory command	49		38	
d	SBHE* setup to 16-bit I/O or 8-bit command	112		101	
8a	16-bit Normal or Ready RD/WR memory command width	250		239	
b	16-bit Normal or Ready RD/WR I/O command width	187		176	
c	16-bit SRDY RD/WR command width	125		114	
d	8-bit Normal or Ready RD/WR command width	530		519	
9	A<19..00> setup to BUSALE	40		29	
10a	16-bit memory read data access		209		187
b	16-bit I/O read data access		132		110
c	16-bit zero wait state read data access		70		48
d	8-bit read data access		489		467
11a	16-bit memory write data setup	-29		-40	
b	16-bit I/O write data setup	33		22	
c	8-bit write data setup	33		22	
12	A<19..00>, SBHE* hold from cmd	22		11	
13a	16-bit command off time	125		114	
b	8-bit command off time	187		176	
14	Read data setup	40		62	
15a	Read data hold	0		0	
b	Write data hold	32		32	
16	Cmd off to D<15..00> tri-state		30		30
17	SRDY* valid from command		40		18

Table 9.0.1 (continued)

Param	DESCRIPTION (Times in NANOSECONDS)	BUS OWNER		ACCESSED RESOURCE	
		Min	Max	Min	Max
18	IOCS16* valid from A<19..00>		126		90
19	IOCS16* hold from A<19..00>	0		0	
20a b	IOCHRDY low from 16-bit command IOCHRDY low from 8-bit command		66 378		44 356
21	IOCHRDY active pulse width	TCLK	15600	TCLK	15600
22	Command hold from IOCHRDY	TCLK		TCLK	
23	BUSALE active from command high	50		61	
24	Clock period (Tclk)	125	167	125	167
25	Data setup to IOCHRDY inactive			-63	
26	LA 17-23 hold to memory command active	41		30	
27	SRDY* pulse width	125		125	
28	SRDY* setup to SYSCLK falling edge			10	
29	SRDY* hold from SYSCLK falling			20	

NOTES:

(1) LA<23-17> timing is the same as A<19..00> when the bus owner is not the Primary CPU

Table 9.0.2 REFRESH CYCLES

Param	DESCRIPTION (Times in NANOSECONDS)	REFRESH CONTROLLER		ADD-ON CARD	
		Min	Max	Min	Max
1	MRDC*/MRDC* active pulse width	250		239	
2	A<07..00> setup to MRDC*	125		114	
3	A<07..00> hold from MRDC*	10		21	
4	IOCHRDY low from MRDC*/MRDC*		81		59
5	MRDC* inactive from IOCHRDY	125	250	125	261
6	MEMREF* setup to MRDC*	250		239	
7	MEMREF* hold from MRDC* (1)	125	250	125	211
8	A<07..00> & MRDC* tri-state from MRDC* disabled		TCLK		
9	IOCHRDY pulse width	TCLK		TCLK	
10	Add-on cards or Primary CPU ownership delay after MEMREF*	2TCLK		2TCLK	

Notes:
(1) MEMREF* may be asserted exceeding the maximum hold time in order to conduct multiple refresh cycles.

Table 9.0.3 DMA CYCLES

Param	DESCRIPTION	ADD-ON CARD AS SOURCE or DMA CONT.		ADD-ON CARD AS DESTIN.	
	(Times in NANOSECONDS)	Min	Max	Min	Max
1	DACKn*, AEN setup to IORC*,IOWC*	145		134	
2	Address setup to memory command	102		91	
3a b	IORC* setup to MWTC* MRDC* setup to IOWC*	235 0		224 0	
4a b	Data access from IORC* (1) Data access from MRDC* (1)		230 261		241 272
5a b	Data setup to MWTC* enabled Data setup to IOWC* enabled			-21 -214	
6	Read cmd hold from write cmd off	50		39	
7	Address hold from cmd off	50		39	
8	Data hold from read command (1)	11		0	
9	IOCHRDY low from memory cmd.(1)		125		90
10	Term count setup to cmd enable	-60	60	-49	49
11	Term count hold from cmd disable	-60	60	-49	49
12a b	IORC* pulse width MRDC* pulse width	700 450		689 439	
13a b	IOWC* pulse width MWTC* pulse width	400 650		389 639	
14	DACKn* hold from command off	60		49	
15	AEN hold from command off	60		49	
16	DRQn inactive from I/O command		119		141
17	IOCHRDY low pulse width	TCLK		TCLK	

NOTE:
(1) Not DMA Controller but add-on card

10.0 SLOT ATTRIBUTES

10.1 SLOT PIN ASSIGNMENTS

These are the pin assignments looking down onto the top (open card-receiving side) of the connectors.

62-pin [8] upper connector:

Table 10.1.0 PIN ASSIGNMENTS

SIGNAL	PIN	PIN	SIGNAL
<TOP>			
Ground	B1	A1	IOCHCK*
RSTDEV	B2	A2	D07
+ 5 V	B3	A3	D06
IRQ09	B4	A4	D05
-5 V	B5	A5	D04
DRQ2	B6	A6	D03
- 12 V	B7	A7	D02
SRDY	B8	A8	D01
+ 12 V	B9	A9	D00
Ground	B10	A10	IOCHRDY*
MEMW*	B11	A11	AEN
MEMR*	B12	A12	A19
IOWC*	B13	A13	A18
IORC*	B14	A14	A17
DACK3*	B15	A15	A16
DRQ3	B16	A16	A15
DACK1*	B17	A17	A14
DRQ1	B18	A18	A13
MEMREF*	B19	A19	A12
SYSCLK	B20	A20	A11
IRQ07	B21	A21	A10
IRQ06	B22	A22	A09
IRQ05	B23	A23	A08
IRQ04	B24	A24	A07
IRQ03	B25	A25	A06
DACK2*	B26	A26	A05
TC	B27	A27	A04
BUSALE	B28	A28	A03
+ 5 V	B29	A29	A02
84OSC	B30	A30	A01
Ground	B31	A31	A00
<BOTTOM>			

36-pin [8/16] lower connector:

Table 10.1.1 PIN ASSIGNMENTS

SIGNAL	PIN	PIN	SIGNAL
<TOP>			
MCS16*	D1	C1	SBHE*
IOCS16*	D2	C2	LA23
IRQ10	D3	C3	LA22
IRQ11	D4	C4	LA21
IRQ12	D5	C5	LA20
IRQ15	D6	C6	LA19
IRQ14	D7	C7	LA18
DACK0*	D8	C8	LA17
DRQ0	D9	C9	MRDC*
DACK5*	D10	C10	MWTC*
DRQ5	D11	C11	D08
DACK6*	D12	C12	D09
DRQ6	D13	C13	D10
DACK7*	D14	C14	D11
DRQ7	D15	C15	D12
+ 5 V	D16	C16	D13
SECMAST*	D17	C17	D14
Ground	D18	C18	D15
<BOTTOM>			

10.2 SIGNAL ELECTRICAL DEFINITIONS

Abbreviations used to describe the electrical characteristics of the Intel ISA bus are listed below.

TRI Tri-state driver. A driver capable of three states: active high, active low, and off.

OC Open collector driver. A driver capable of two states: active low, and off.

TTL Totem pole driver. A driver capable of two states: active high, and active low.

Iih High-level input current. The current into (*) an input when a high-level voltage is applied to that input.

Iil Low-level input current. The current into (*) an input when a low-level voltage is applied to the input.

Ioh High-level output current. The current into (*) an output with input conditions applied that, according to the device's spec, will establish a high level at the output.

Iol Low-level output current. The current into (*) an output with input conditions applied that, according to the device's spec, will establish a low level at the output.

Iozh Off-state (high-impedance state) output current (of a tri-state output) with high level voltage applied. The current flowing into (*) an output having tri-state capability with input conditions established that, according to the device's spec, will establish the high-impedance state at the output and with a high-level voltage applied to the output.

Iozl Off-state (high-impedance state) output current (of a tri-state output) with low level voltage applied. The current flowing into (*) an output having tri-state capability with input conditions established that, according to the device's spec, will establish the high-impedance state at the output and with a low-level voltage applied to the output.

NOTE: By convention, current flow into a device terminal is signed positive, while that flowing out of a device terminal is signed negative.

Vih High-level input voltage. An input voltage within the more positive of the two ranges used to represent the binary variables ON and OFF. A minimum is specified that is the least positive value of **Vih** for which proper operation of the logic device (within its spec limits) is guaranteed.

Vil Low-level input voltage. An input voltage within the less positive of the two ranges used to represent the binary variables ON and OFF. A maximum is specified that is the most positive value of **Vil** for which proper operation of the logic device (within its spec limits) is guaranteed.

Voh High-level output voltage. The voltage at an output terminal with input conditions applied that, according to the device's spec, will establish a high level at the output.

Vol Low-level output voltage. The voltage at an output terminal with input conditions applied that, according to the device's spec, will establish a low level at the output.

10.3 BUS SIGNAL VOLTAGE & CURRENT LEVELS

There are three types of drivers/receivers used in the Intel ISA bus: TTL, Tristate (TRI), and Open Collector (OC). The TTL devices must be of fixed direction; either input or output. TRISTATE devices can be either input or output at different times, and can also be of a high impedance (Tristate). The currents listed in Table 10.3 are speced per card slot.

	TTL (1)		TRISTATE (1)			O.C.(1)
	DRIVER	RECEIVER	DRIVER	RECEIVER	TRI	DRIVER
Iil	-	-.8	-	-.8	-.4	-.4 (2)
Iih	-	.04	-	.04	.04	.02 (2)
Iol	24	-	24	-	.4	24
Ioh	-3	-3	-3	-	-.04	(3)

Table 10.3 VOLTAGE & CURRENT

NOTE:

(1) **Voh= 2.4** **Vih= 2.7**
Vol= .5 **Vil= .4**
The entries in the table are in **MILLIAMPS.**
The "-" sign in front of a number indicates that current is flowing out of the add-on card into the connector on the backplane.

(2) An open collector signal line may have TTL input attached to it.

(3) The Ioh of an open collector leakage current of each slot must be less than .4 milliamps.

10.4 ADDITIONAL ADD-ON CARD DRIVER & RECEIVER REQUIREMENTS

The design of the add-on cards require additional driver and receiver considerations than those specified in Table 10.3. The additional requirements are:

Signal Stub Length: The signal stub length on an add-on card is defined as the length of the longest trace from the connector pin to any bus signal driver or receiver in that net. The maximum stub length allowed is 2.5 inches.

Driver Rise & Fall Times: To minimize crosstalk between signals and bus reflections, the driver rise/fall rates must be greater than 3ns; only LS or ALS devices are allowed. Faster parts such as F and AS cannot be used to drive the bus.

Capacitive Loads per Signal Line: 15 pF maximum capacitive load per signal per slot. This capacitive load rating includes all drivers and receivers connected to the signal pin, plus all trace capacitance used in the signal pin's on-adapter net.

Signal terminations: None are allowed; all are provided by the Intel platform.

10.5 BACKPLANE RESISTORS

The backplane has two types of resistors. One type is a pull-up which is attached from the signal line to +5 volts. The second type is a series resistance placed at one end of the signal line backplane. Table 10.5 lists the values of these two types; some lines do not have a resistor of either type.

LINE	PULL-UP	SERIES
IOCHCK*	4.7K	-
IOCHRDY	1.0K	-
IOCS16*	300	-
IORC*	4.7K	22
IOWC*	4.7K	22
SECMAST*	300	-
MCS16*	300	-
MRDC*	4.7K	22
MWTC*	4.7K	22
84OSC	-	22
MEMMEMREF*	300	-
MEMR*	4.7K	22
MEMW*	4.7K	22
SYSCLK	-	27
SRDY*	300	

Table 10.5 BACKPLANE RESISTORS

11.0 MECHANICAL SPECIFICATION

The mechanical specifications for the Intel ISA bus standard are limited to requirements for the add-on card.

11.1 ADD-ON CARD CONNECTOR-SET SPECIFICATION

This section describes the physical and electrical requirements of the platform's set of connectors implementing the Intel ISA bus.

The add-on connectors are male card-edge connectors with 0.100 inch contact spacing; PC mounted.

The platform implementation of the Intel ISA bus can support up to ten slots. All slots are on .8 inch centers. Seven of these slots offer 98-pin connections via one monolithic (62 + 36 pin) connector and support 16-bit data transfers; one slot offers only a 62-pin connection via one monolithic connector and supports just 8-bit data-transfers.

The physical characteristics of the connectors are:

98-pin connector:
  Intel part number 108768-049
  Intel-approved manufacturers:
    AMP: #645169-3
    TI: #H421023-52-2

62-pin connector:
  Intel part number 108298-031
  Intel-approved manufacturers:
    AMP: #6530843-5
    TI: #H421021-31-3 & H421023-31-3
    Viking: #3VT31/9JNK12 & 3VT31/2JNK12

Outline drawings follow.

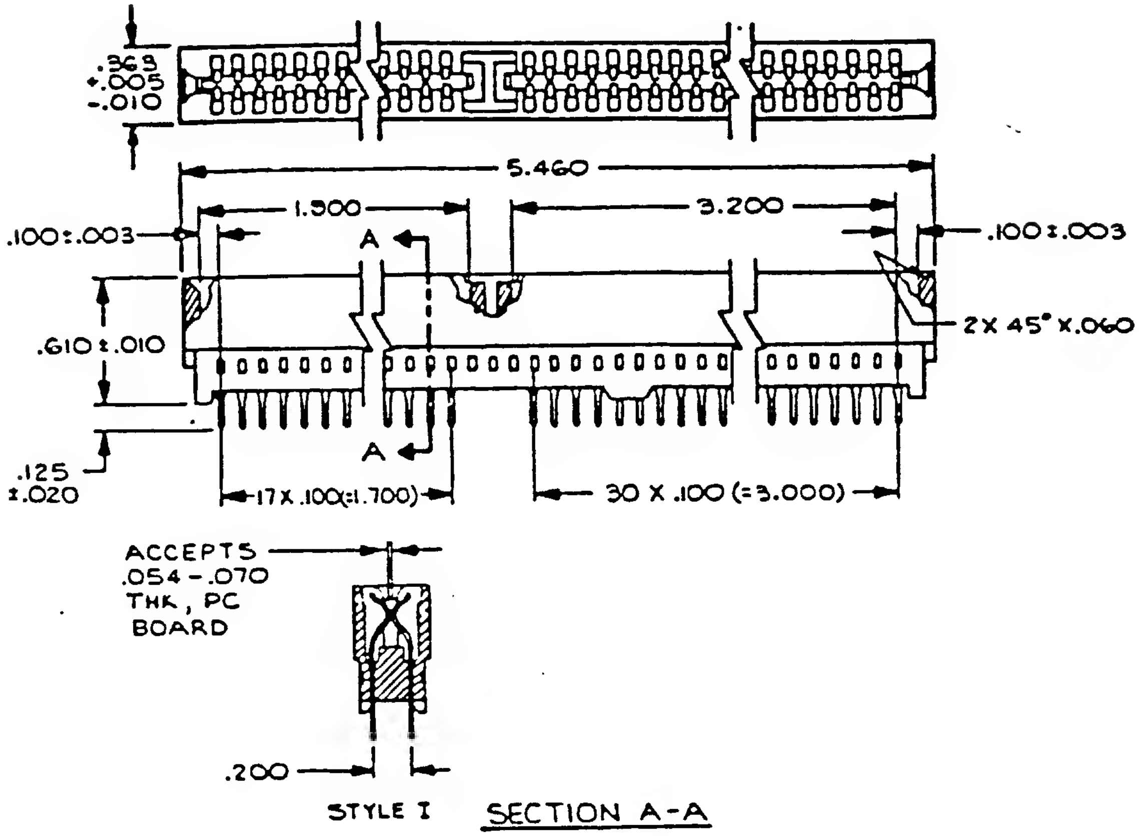


Figure 11.1.0 98-PIN CONNECTOR

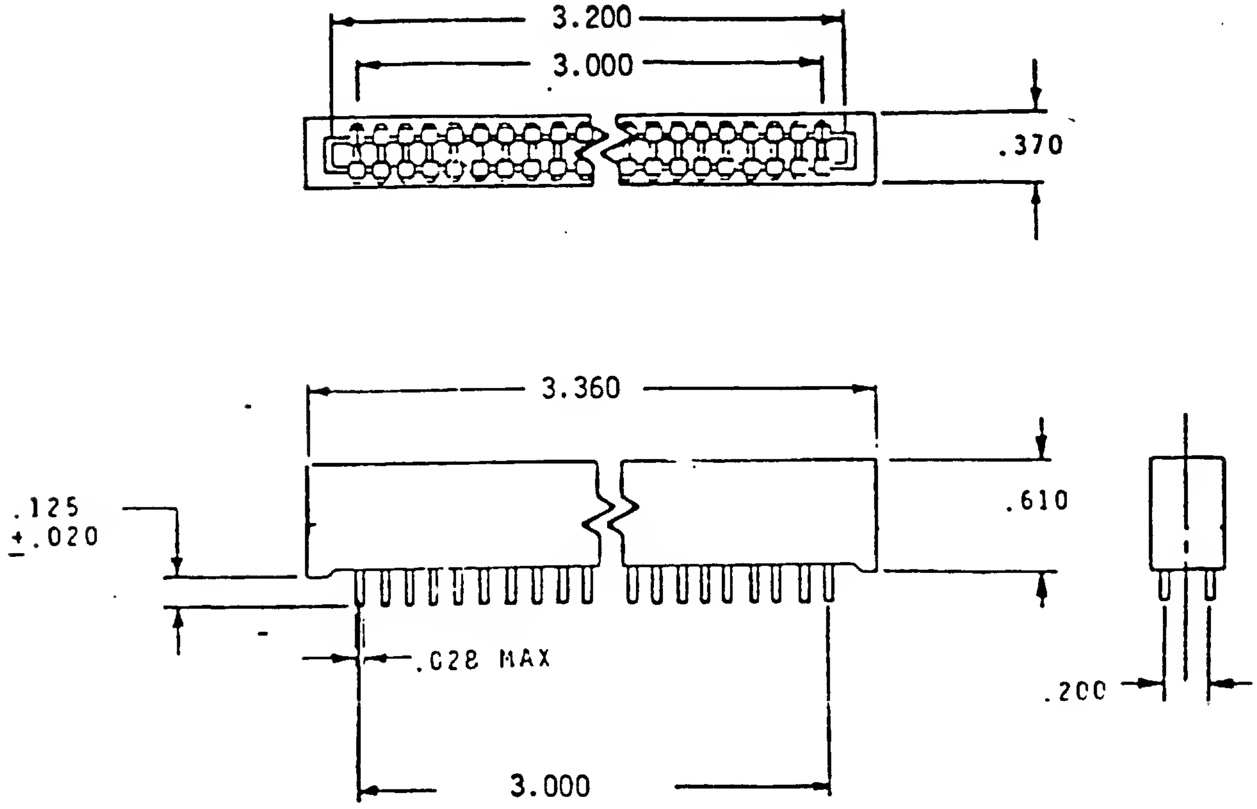


Figure 11.1.1 62-PIN CONNECTOR

12.0 ADD-ON CARDS SIZES AND DIMENSIONS

This section lists and defines the mechanical specifications with which a designer must be concerned to ensure that add-on cards will be dimensionally compatible to the Intel ISA bus as implemented.

NOTE

The designer should consult the manufacturer of the enclosure in which the add-on card is to be housed to get air flow, ambient temperature, and other data.

12.1 DEPTH & SIZE

The maximum size for an adapter card is 4.20 inches by 13.12 inches; these outside measurements include the edge connector.

The board thickr.ess spec is: 0.062 +/- 0.008 inches.

An outline drawing of the maximally-sized and maximally-pinned (16-bit data transfers using all 98 bus pins) follows.

Smaller-sized and smaller-pinned (62 pins for 8-bit data transfers) cards can be used, so long as two sets of dimensions are met:

1) The specified lower edge pinning dimensions of this specification.
2) The additional specifications of the enclosure chosen; this detail is outside the scope of this specification.

CAUTION

"Drop" card form factors also exist; their definition and use is outside the scope of this specification, since both definition and use are motherboard (Primary CPU) specific; Intel does not encourage the use of these form factors.

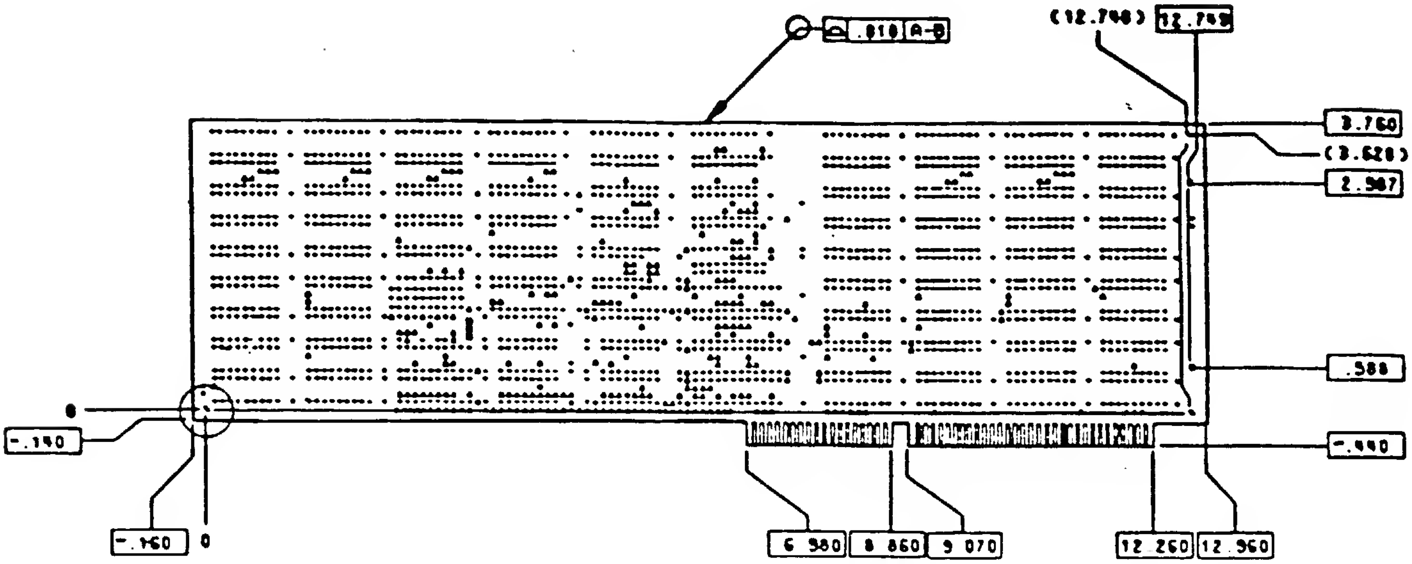


Figure 12.0 MAXIMALLY-SIZED AND PINNED ADAPTER CARD FORM FACTOR

12.2 SPACING & THICKNESS

The following guidelines should be followed if the designer wishes to occupy only one (1) slot without touching either an adjacent add-on card or parts of the platform.

The ten connectors are mounted on 0.8 inch centers. The existence of "dead areas" to the right and left of the "field" of connectors is platform-dependent (and thus outside the scope of this specification); nevertheless each add-on card is entitled to a "thickness footprint" computed thus:

Add-on card attributes:

(1) Board Thickness: 0.062 +/- 0.008 inches.

(2) Component Lead Length: 0.071 inches (max).

(3) Board Warpage: Board warpage, the deviation of the board from the plane which the card ends define, must not exceed 0.050 inches when the board edge and motherboard connector are engaged.

(4) Component Height: 0.40 inches (max).

Implied footprint: 0.641 inches

To right (primary side) of the connector centerline:
0.035 Half board thickness (max)
0.400 Component height
0.050 100% warpage allowance
0.485 inches

To left (secondary side) of the connector centerline:
0.035 Half board thickness (max)
0.071 Lead length
0.050 100% warpage allowance
0.156 inches