STD 32 BUS SPECIFICATION

AND

DESIGNER'S GUIDE

Version 2.1

July 3, 1996

Order Number: ZT MSTD32
Suggested Retail: $50.00

ZIATECH CORPORATION

Corporate Headquarters •1050 Southwood Drive • San Luis Obispo, CA 93401 USA
FAX (805) 541-5088 • Telephone (805) 541-0488

PREFACE

Since its inception in 1978, the STD bus has proven to be the workhorse of industrial control implementations. At that time, 8 bits of data were considered adequate for the tasks at hand. With advances in technology, including the 16- and 32-bit data paths and 24- and 32-bit addressing spaces of today's processors, the STD bus has grown to encompass these new capabilities with the STD 32 specification.

The strength of any bus structure is not in its theoretical advantages, but rather in the number and variety of cards manufactured for that bus structure. The STD 32 specification utilizes the abundance of I/O expansion cards that currently exist for the STD marketplace. Users are able to protect their investment in expansion boards while upgrading the processor for increased functionality and throughput. They can also add to the capability of their system by adding I/O cards to meet application needs. This availability of a variety of compatible products frees STD bus users from a single vendor, allowing them to select the best products from several vendors to meet their needs.

The STD 32 specification provides a solid foundation for designers of I/O cards, memory cards, and CPU cards. It defines an electrical specification, a mechanical specification, and a functional description of transactions and the modules associated with each transaction.

STD 32 is a growth path for existing STD bus users and manufacturers to better utilize today's technology. It does not further complicate the design of simple I/O expansion cards. In fact, the majority of existing cards in the marketplace can be used. At a higher level, it addresses the need for efficient multiple masters, backplane DMA, and additional interrupts.

STD 32 retains simple control bus features but with many additional capabilities. Today's users have access to the latest microprocessors, such as the 80486, and yet can still talk to I/O cards designed years ago!

CONTENTS

TABLES

TABLES

ILLUSTRATIONS

C H A P T E R 1

INTRODUCTION TO THE STD 32 BUS SPECIFICATION

CHAPTER 1

1.1. ARCHITECTURAL OVERVIEW

The STD 32 bus specification is a superset of the STD-80 series architecture. It extends the capabilities of the STD-80 series standard while remaining compatible with existing cards.

The physical form factor remains the same and the pinouts for the STD-80 series remain in effect. The additional pins required for the STD 32 bus are inserted on 0.0625 inch centers between the existing STD-80 pins and on each of two extensions.

1.1.1 STD 32 FEATURES

The STD 32 bus specification adds several new features to the bus, including:

- Compatibility with existing STD-80 series peripheral cards
- Support for EISA-like chip set extensions
- 32-bit addressing capability
- 16- and 32-bit data transfers
- Temporary Master (multiple master) capability (bus arbitration)
- Dynamic bus sizing (8-, 16-, and 32-bit data transfers)
- Backplane DMA
- Slot-specific interrupts
- Software-configurable peripheral boards
- Compliance levels

These features give users of STD-80 series systems a growth path to increase the capabilities of their application without endangering their investment in peripherals such as I/O cards and memory cards. At the same time, these users can upgrade their systems to 80386- or 80486-style microprocessors to take advantage of these architectures, both in hardware and in software. Designers of application specific cards will still enjoy the ease of design and production inherent in the STD bus.

1.2. THE STD 32 BUS

STD 32 is comprised of STD-80 series architecture with extensions to provide increased performance. The STD-80 bus remains intact, with some minor enhancements to remove existing inconsistencies. The STD 32 bus defines additional pins (E pins) between the STD-80 pins (P pins) for extended capabilities such as bus arbitration, additional interrupt lines, DMA capability, and additional address and data signals.

The E pins extend the capabilities of P by providing a set of features derived from the Extended Industry Standard Architecture (EISA) bus. Because of this, silicon support already exists for system designers who wish to utilize the features of the E pins.

E pins do more than just add additional data and address lines. The E pins allow DMA transfers in 8-, 16-, and 32-bit data widths, and also support burst transfers. STD-80 systems require frontplane connectors in order to achieve DMA transfers. STD 32 allows bus arbitration for Temporary Masters.

The E pins also allow slot-specific addressing signals to reduce the number of jumper options and to support additional interrupts.

1.2.1 STANDARD ARCHITECTURE COMPATIBLE BUS CYCLES

Standard Architecture Compatible Bus Cycles: 8-Bit Transfers

The P bus is essentially the existing STD-80 series bus. The lower data signals D0-D7 and the address signals A0-A23 are found on this bus. The manufacturers of simple I/O cards need only concern themselves with the features of this bus, as only 8-bit data is typical for their applications.

Bus transfers consist primarily of the master defining the type of transfer (I/O or memory), direction of transfer, and use of WAITRQ* to extend cycles, if necessary. Data is limited to 8 bits and addressing to 24 bits. More than one Temporary Master is permitted, but it is generally limited to one other device, typically a DMA controller. 8-bit STD-80 transfers are referred to as Standard Architecture 8 (SA8) transfers within STD 32.

16-Bit Transfers

The STD 32 bus supports extended width data transfers by using the extra data signals found on E during STD-80 series style transfers. Peripheral cards that are 16 bits wide will drive the signals MEM16* or IO16* to indicate to the master that the board is capable of 16-bit data transfers. The master, if capable, will drive BHE* and A0 to indicate the location of valid data. If the peripheral boards do not drive MEM16* or IO16*, then the default 8-bit cycle will result, with even and odd data being transferred sequentially over D0-D7. This dynamic feature allows 8- and 16-bit cards to coexist in an STD 32 system. These 16-bit transfers are referred to as SA16 transfers within the STD 32 specification.

1.2.2. EXTENDED ARCHITECTURE COMPATIBLE BUS CYCLES

An Extended Architecture (EA) compatible bus cycle operates with the same data and address signals as described above except with the possibility of increasing the number of data and address signals to 32 bits each. There may be 8-, 16-, and 32-bit EA compatible bus cycles, each defined by the data width transferred. Dynamic bus sizing is used to determine the width of the data transfer. This allows 8-, 16-, and 32-bit cards to coexist in a single STD 32 system. These Extended Architecture transfers are referred to as EA8, EA16, and EA32 transfers within the STD 32 specification.

A set of signals different from those used during STD-80 bus cycles is used to control the STD 32 cycle. This set is found on the E connector and includes START*, CMD*, M-IO, and W-R. The Permanent Master or a Temporary Master defines the type of transfer, direction of transfer, and the number of wait states to insert in the cycle, if necessary.

In an STD 32 system where the Permanent Master is controlling Slot X, each of the first 14 slots on the backplane are assigned as logical slots to the Permanent Master. Each slot is assigned a dedicated set of signals (AENx*, DREQx*, DAKx*, MREQx*, MAKx*, and IRQx). These signals are physically tied to Slot X, which must be the left-most slot on the backplane. The signals tied to it are not bussed with the rest of the signals. Slot X is dedicated to driving and receiving slot-specific signals. A full-featured Permanent Master resides in slot 0 and thus has responsibility for Slot X and Slot 0. Slots 1 through 14 are available for slot-specific cards. Backplanes can consist of more slots, but 14 are available logically for the extended features of servicing master requests and automatic configuration.

In an STD 32 system where the Permanent Master is not controlling Slot X, signals are connected as described above. The Permanent Master may reside in any slot, but Slot 0 is recommended for consistency. Additional Temporary Masters may be added in any of the 13 remaining slots, assuming a Slot X bus arbiter has been installed to support the slots.

1.3 NOTATIONS AND DEFINITIONS

The following notational conventions are used throughout this specification.

Signal Notation Slot-specific signals are shown as the signal name followed by a lower case "x", for example, AENx*.

Negative true signals are indicated by an asterisk (*) following the signal name, for example, INTRQ* (see Table 1-1 “Signal Notation” below).

Table 1-1. Signal Notation.

Signal Name	Electrical Notation	Logical Notation	State
NAME	H L	1 or True 0 or False	Active Inactive
NAME*	L H	1 or True 0 or False	Active Inactive

Radix Notation Hexadecimal numbers are indicated by a lower case "h" following the digits, for example, 3FFh.

Binary numbers are indicated by a lower case "b" following multiple digits, for example, 0101b. If a one-digit binary number is indicated, it may not have the "b" suffix.

Abbreviations	A	amp
	Kbyte	Kilobyte
	kHz	kilohertz
	Mbyte	megabyte
	MHz	megahertz
	µs	microsecond
	µA	microamp
	ms	millisecond
	mA	milliamp
	ns	nanosecond
	Ω	ohm
	V	volt
	W	watt

1.4. SUMMARY

The STD 32 bus specification is intended to ensure a planned migration path from current STD bus bandwidths to those matching technology currently available and for years to come.

Special emphasis has been placed on maintaining compatibility in two key areas:

1. Hardware compatibility with existing STD peripheral cards to provide a depth of I/O support
2. Compatibility with processors and chip sets to provide support for the abundant software already designed, and to be designed in the future, for the personal computer marketplace

These have proven to be mutually compatible goals that have been implemented without sacrificing the reliability and form factor advantages of the STD bus.

CHAPTER 2

STANDARD ARCHITECTURE BUS CYCLES

STANDARD ARCHITECTURE BUS CYCLES

CHAPTER 2

2.1. INTRODUCTION

The STD 32 bus specification defines the mechanical and electrical parameters needed to support 8, 16, and 32-bit microprocessor systems. This chapter discusses the signals used for an 8-bit or 16-bit STD-80 compatible design, referred to as Standard Architecture (SA) transfers. For 8-bit designs, these signals are nearly identical to those defined in the STD-80 series bus specification. It is this similarity that permits boards designed to the STD-80 series bus specification to be used in the STD 32 bus.

For 16-bit designs, additional pins have been defined to prevent additional multiplexing and to incorporate dynamic bus sizing. Chapter 3 expands upon the signals discussed in this chapter to support 8, 16, and 32-bit EISA-like designs.

Throughout this chapter, backplane signals are referred to as being either P or E. Signals designated as P refer to STD-80 signals and pin assignments. Signals designated as E are additional (or extension) signals and pin assignments added to the original STD-80 series bus specification. Please see Chapter 6 for a mechanical representation of E and P signals.

Additional frontplane signals have been defined to support optional DMA transfers between memory and I/O slaves. Pinout and signal descriptions are provided in the "Frontplane DMA Control" section, page 23.

2.2. SIGNAL PIN ASSIGNMENTS

The signals discussed in this chapter are organized into four functional groups as shown in Table 2-1 "Standard Architecture Functional Pin Assignments".

Table 2-1. Standard Architecture Functional Pin Assignments.

Functional Group	P Pin Locations	E Pin Locations
Power Bus	1-5, 53-56	13, 14, 17, 18, 28, 35, 45, 54, 68
Data Bus	7-14	30-44 Even
Address Bus	7-14, 15-30	--
Control Bus	6, 31-52	15, 16, 47, 53, 59

- The power bus supports +5V, +12V, -12V, 3V nominal battery voltage, and multiple separate ground returns.
- The data bus includes 8 or 16 data lines.
- The address bus includes 24 address lines. Pins 7-14 appear on both the address and data bus because the upper eight address lines are time multiplexed on the data bus.
- The control bus includes all of the remaining signals needed to manage 8- or 16-bit operations. Additional optional control signals on E pins are available for Temporary Masters, interrupts, and dynamic bus sizing.

More complete pin assignments are listed in Table 2-2 "P Pinouts", and Table 2-3 "E Pinouts". In addition to pin numbers, each table includes the standard mnemonic for each signal and the signal direction relative to the Permanent Master when in control of the bus.

CHAPTER 2

Table 2-2. P Pinouts.

COMPONENT SIDE				CIRCUIT SIDE			
Pin	**Mnemonic**	**Signal Flow**	**Description**	**Pin**	**Mnemonic**	**Signal Flow**	**Description**
1	+5VDC	In	Logic Power	2	+5VDC	In	Logic Power
3	GND	In	Logic Ground	4	GND	In	Logic Ground
5	VBAT	Bidir	Battery Power	6	DCPDN*	Bidir	DC Power Down
7	A19/D3	Out/Bidir	Address/Data	8	A23/D7	Out/Bidir	Address/Data
9	A18/D2	Out/Bidir	Address/Data	10	A22/D6	Out/Bidir	Address/Data
11	A17/D1	Out/Bidir	Address/Data	12	A21/D5	Out/Bidir	Address/Data
13	A16/D0	Out/Bidir	Address/Data	14	A20/D4	Out/Bidir	Address/Data
15	A7	Out	Address	16	A15	Out	Address
17	A6	Out	Address	18	A14	Out	Address
19	A5	Out	Address	20	A13	Out	Address
21	A4	Out	Address	22	A12	Out	Address
23	A3	Out	Address	24	A11	Out	Address
25	A2	Out	Address	26	A10	Out	Address
27	A1	Out	Address	28	A9	Out	Address
29	A0	Out	Address	30	A8	Out	Address
31	WR*	Out	Write Mem or I/O	32	RD*	Out	Read Mem or I/O
33	IORQ*	Out	I/O Address Select	34	MEMRQ*	Out	Mem Address Select
35	IOEXP	Out	I/O Expansion	36	BHE*	Out	Byte High Enable
37	INTRQ1*	In	Interrupt Request 1	38	ALE*	Out	Address Latch Enable
39	STATUS1*	Out	CPU Status 1	40	STATUS0*	Out	CPU Status 0
41	BUSAK*	Out	Bus Acknowledge	42	BUSRQ*	In	Bus Request
43	INTAK*	Out	Interrupt Acknowledge	44	INTRQ*	In	Interrupt Request
45	WAITRQ*	In	Wait Request	46	NMIRQ*	In	Non-Mask Int Request
47	SYSRESET*	Out	System Reset	48	PBRESET*	In	Push-Button Reset
49	CLOCK*	Out	Clock	50	CNTRL*	Bidir	Aux Timing
51	PCO	Out	Priority Chain Out	52	PCI	In	Priority Chain In
53	AUX GND	In	AUX Ground (bussed)	54	AUX GND	In	AUX Ground (bussed)
55	AUX +V	In	AUX Positive (+12VDC)	56	AUX -V	In	AUX Positive (-12VDC)

Notes: An asterisk (*) indicates a low level active signal.
Address lines A16-A23 are multiplexed on data lines D0-D7 on each address cycle for STD-80 compatibility.
PCO and PCI are not typically used on peripheral cards.
All boards not supporting PCO and PCI should tie these two signals together.

Table 2-3. E Pinouts.

COMPONENT SIDE				CIRCUIT SIDE			
Pin	**Mnemonic**	**Signal Flow**	**Description**	**Pin**	**Mnemonic**	**Signal Flow**	**Description**
13	+5VDC	In	Logic Power	14	+5VDC	In	Logic Power
15	DAKx*	Out	DMA Acknowledge	16	DREQx*	In	DMA Request
17	GND	In	Logic Ground	18	GND	In	Logic Ground
19	D27	Bidir	Data	20	D31	Bidir	Data
21	D26	Bidir	Data	22	D30	Bidir	Data
23	D25	Bidir	Data	24	D29	Bidir	Data
25	D24	Bidir	Data	26	D28	Bidir	Data
27	D23	Bidir	Data	28	GND	In	Logic Ground
29	D22	Bidir	Data	30	D15	Bidir	Data
31	D21	Bidir	Data	32	D14	Bidir	Data
33	D20	Bidir	Data	34	D13	Bidir	Data
35	GND	In	Logic Ground	36	D12	Bidir	Data
37	D19	Bidir	Data	38	D11	Bidir	Data
39	D18	Bidir	Data	40	D10	Bidir	Data
41	D17	Bidir	Data	42	D9	Bidir	Data
43	D16	Bidir	Data	44	D8	Bidir	Data
45	GND	In	Logic Ground	46	MASTER16*	Out	Master 16-Bit
47	IRQx	In	Interrupt Request	48	AENx*	Out	Address Enable
49	BE1*	Out	Byte Enable 1	50	BE3*	Out	Byte Enable 3
51	BE0*	Out	Byte Enable 3	52	BE2*	Out	Byte Enable 2
53	MEM16*	In	Memory 16-Bit	54	GND	In	Logic Ground
55	M-IO	Out	Memory or I/O	56	W-R	Out	Write or Read
57	DMAIOW*	Out	DMA I/O Write	58	DMAIOR*	Out	DMA I/O Read
59	IO16*	In	I/O 16-Bit	60	EX8*	In	Exchange 8-Bit
61	CMD*	Out	Command	62	START*	Out	Start
63	EX16*	In	Exchange 16-Bit	64	EX32*	In	Exchange 32-Bit
65	EXRDY	In	Exchange Ready	66	T-C	Bidir	Terminate Count
67	INTRQ3*	In	Interrupt Request 3	68	+5VDC	In	Logic Power
69	MAKx*	Out	Master Acknowledge	70	MREQx*	In	Master Request

Notes: E pins E1-12 and E71-80 are not needed for STD-80 transfers and are therefore not shown here. Refer to Chapter 3 for a complete listing of E signals.

The Permanent Master is assumed to be in control of the Slot X resources.

CHAPTER 2

2.2.1. POWER BUS

(P Pins 1-5, 53-56 and E Pins 13-14, 17-18, 28, 35, 45, 54, 68)

Table 2-4 "Power Bus Pin Assignments" shows the pin assignments for the power bus. The power bus supports four voltages: +5V for digital devices, both +12V and -12V for auxiliary devices, and battery voltage.

Table 2-4. Power Bus Pin Assignments.

P PIN	E PIN	DESCRIPTION	COMMENTS
1, 2	13, 14, 68	Digital Power	Digital Power Source (+5VDC ±0.25V)
3, 4	17, 18, 28, 35, 45, 54	Digital Ground	Digital Power Return (GND)
5		Battery Voltage	Battery Backup Source ($2.0 \leq Vbat \leq 5V$)[1]
53, 54		Auxiliary Ground	Auxiliary Power Return (AUX GND)[2]
55		Auxiliary Positive	Auxiliary Positive Supply (+12VDC ±0.5V)
56		Auxiliary Negative	Auxiliary Negative Supply (-12VDC ±0.5V)

[1]The Vbat supply is optional.

[2]AUX GND is normally tied to digital ground.

2.2.2. DATA BUS

(P Pins P7-14, E44-30 Even)

The data bus includes 16 bidirectional active-high data lines driven by a three-state driver on any board. The data bus direction is controlled by the Permanent or Temporary Master using control bus signals. Any board not currently selected must release the data bus to a high impedance state either by disabling the data bus drivers or pointing the data bus drivers away from the backplane. The Permanent Master must release the data bus when transferring control to a Temporary Master. The Temporary Master determines who drives the data bus signals when in control of bus resources. The Permanent Master must terminate all data pins with a 1.8kΩ ± 10% resistor to +5V.

Both 8-bit and 16-bit data subsets support a 24-bit address bus using 16 dedicated address lines (A0-A15) and time multiplexing the remaining eight address lines (A16-A23) with the eight data lines (D0-D7). The ALE* signal is provided to latch the multiplexed address.

8-bit data subset boards use data bus pins D0-D7 on P pins 7-14 and ignore D8-D15. 16-bit data subset boards use data bus pins D0-D7 on P pins 7-14 and D8-D15 on E pins 44-30 even. The combination of address line A0 and BHE* defines the data bus lines that are used during a data transfer (see Table 2-5 "16-Bit Byte Lane Usage").

Table 2-5. 16-Bit Byte Lane Usage.

BHE*	A0	Operation
LOW	LOW	16-bit transfer low byte on D0-7 high byte on D8-15
LOW	HIGH	8-bit transfer byte on D8-15
HIGH	LOW	8-bit transfer byte on D0-7
HIGH	HIGH	Reserved

2.2.3. ADDRESS BUS

(P Pins 7-14, 15-30)

The address bus supports 24 bidirectional active-high address lines generated by three-state drivers on the Permanent or Temporary Masters. The upper eight address lines (A16-A23) are time multiplexed with the eight data lines (D0-D7). The rising edge of the ALE* signal is used to latch the multiplexed address as shown in Figure 2-1 "8-Bit Memory Addressing". Permanent Masters must terminate all address pins with a 1.8kΩ ± 10% resistor to +5V.

The STD 32 bus masters drive all of the address lines during a memory operation but only the lower 16 address lines during an I/O operation. Memory slaves should decode 24 bits of address. I/O slaves should decode 16 bits of address. The Permanent Master must release the address bus when transferring control to a Temporary Master. The Temporary Master must manage all 24 address lines on the address bus when in control of bus resources.

Permanent or Temporary Masters should multiplex a cascade address for slave interrupt controllers on address lines A8-A10, as defined in the timing diagrams at the end of this chapter, if slave (cascaded) interrupt controllers are supported.

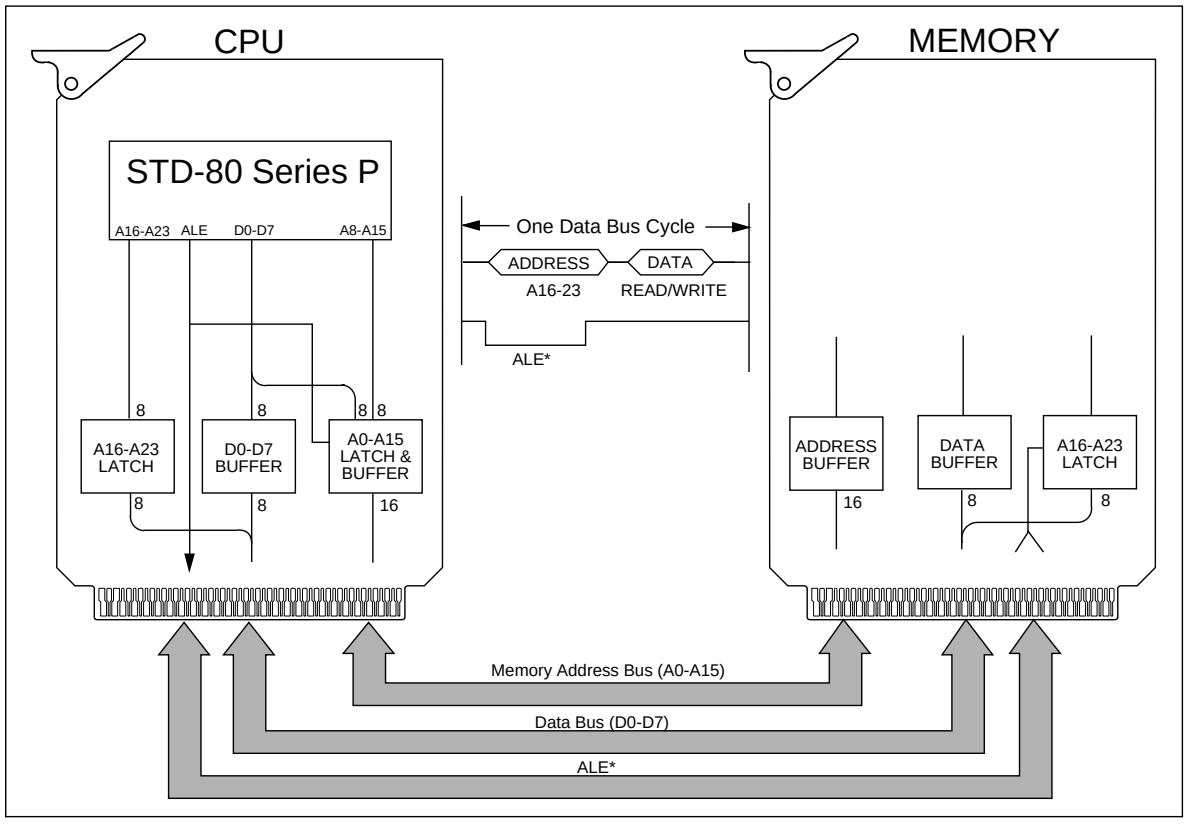


Figure 2-1. 8-Bit Memory Addressing.

2.2.4. CONTROL BUS

(P Pins 6, 31-52)

The control bus includes all the signals needed to coordinate STD bus activity. The control bus can be divided into four standard functional groups and one optional functional group.

- Memory and I/O control
- Peripheral timing control
- Interrupt and bus control
- Clock and reset
- Frontplane DMA control (optional)

The signals supported by these functional groups are discussed in detail below.

Memory and I/O Control

Memory and I/O control signals direct the transfer of data between the boards in the system. These signals are described in detail below. There is only one read and one write signal used for both memory and I/O. Therefore, it is not possible to support direct memory access (DMA) on the 8-bit STD 32 bus without additional frontplane signals.

WR*
(P Pin 31)

Write is an active-low signal generated by a three-state driver on the Permanent or Temporary Master. The WR* signal controls the transfer of data from the Permanent or Temporary Master to a memory or I/O board. Data is not valid on the falling edge of WR*. The Permanent Master must release WR* in response to a bus request (BUSRQ*) from a Temporary Master. The Temporary Master must manage WR* when in control of bus resources. The Permanent Master must terminate WR* with a 1.8kΩ ±10% resistor to +5V.

RD*
(P Pin 32)

Read is an active-low signal generated by a three-state driver on the Permanent or Temporary Master. The RD* signal controls the transfer of data from a memory or I/O board to a Permanent or Temporary Master. The Permanent Master must release RD* in response to a bus request (BUSRQ*) from a Temporary Master. The Temporary Master must manage RD* when in control of bus resources. The Permanent Master must terminate RD* with a 1.8kΩ ±10% resistor to +5V.

IORQ*
(P Pin 33)

I/O Request is an active-low signal generated by a three-state driver on the Permanent or Temporary Master. An active IORQ* indicates that the current cycle is a data transfer between the Permanent or Temporary Master and a slave I/O board. I/O boards typically use IORQ* to qualify the 16-bit board select address. The Permanent Master must release IORQ* in response to a bus request (BUSRQ*) from a Temporary Master. The Temporary Master must manage IORQ* when in control of bus resources.

It is possible on the STD-80 series bus for I/O boards to improperly select during an interrupt acknowledge cycle when IORQ* is active and the address bus is floating. To prevent this from happening, it is necessary to include INTAK* in the I/O board decoding logic to disqualify selection during interrupt acknowledge. Particular attention must be given to wait state generation in this situation. The Permanent Master must terminate IORQ* with a 1.8kΩ ±10% resistor to +5V.

MEMRQ*
(P Pin 34)

Memory Request is an active-low signal generated by a three-state driver on the Permanent or Temporary Master. An active MEMRQ* indicates that the current STD bus cycle is a data transfer between the Permanent Master or Temporary Master and a Memory Slave. Memory boards typically use MEMRQ* to qualify the board select address. The Permanent Master must release MEMRQ* in response to a bus request (BUSRQ*) from a Temporary Master. The Temporary Master must manage MEMRQ* when in control of bus resources. During Permanent and Temporary Master memory refresh cycles, MEMRQ* must not be driven active. The Permanent Master must terminate MEMRQ* with a 1.8kΩ ±10% resistor to +5V.

IOEXP
(P Pin 35)

I/O Expansion is an active-high signal generated by a three-state driver on the Permanent or Temporary Master. The IOEXP low signal is used as an added qualifier for I/O boards that decode fewer than 16 address lines. The Permanent Master must release IOEXP in response to a bus request (BUSRQ*) from a Temporary Master. The Temporary Master must manage IOEXP when in control of bus resources. The Permanent Master must terminate IOEXP* with a 1.8kΩ ±10% resistor to +5V.

The Permanent or Temporary Master will drive IOEXP low during accesses to existing 8-bit I/O addressable boards. Software will access these boards in the I/O address range FC00-FFFFh, which is the slot 15 address space as defined by the STD 32 bus specification. During all other accesses, IOEXP remains not driven and is pulled high by the master. Refer to the timing diagrams at the end of this chapter for the proper setup time required for IOEXP. ALE* is irrelevant to I/O transfers.

An I/O board decoding fewer than 10 address lines should decode IOEXP. An I/O board decoding 16 address lines does not need to decode IOEXP. Some of the older I/O boards designed for STD bus applications decode fewer than 16 address lines. This means that current systems may include some I/O boards that decode all 16 bits of I/O address and some I/O boards that decode fewer than 16 bits. It is necessary to ensure that all boards decoding 16 bits are not mapped into a redundant address location occupied by a board decoding fewer than 16 bits, unless IOEXP is included in their address selection logic.

As an example, assume that there are two I/O boards in a system. One decodes 16 address lines while the other decodes only 8 address lines. The board decoding 8 address lines is mapped at I/O address 80-8Fh. This board is redundantly mapped in a 16-bit I/O address space at 180-18Fh, 280-28Fh, 380-38Fh, and so on. This means the board decoding 16 address lines cannot be mapped into any of these ranges. The IOEXP signal provides a second option for handling this situation by forcing the 8 address line board to appear only at addresses FC80h-FC8Fh.

BHE*
(P Pin 36)

Byte High Enable is an active-low signal generated with a three-state driver by Permanent or Temporary Masters. During a write cycle, the BHE* signal is driven active to inform a 16-bit board that data is available on D8 through D15. During a read cycle, the BHE* signal is driven active to instruct the 16-bit board to provide data on D8 through D15. The Permanent Master must release BHE* in response to a bus request (BUSRQ*) from a Temporary Master. A Temporary Master must manage BHE* when in control of bus resources. 8-bit boards ignore BHE*. The Permanent Master must terminate BHE* with a 1.8kΩ ±10% resistor to +5V.

IO16*
(E Pin 59)

I/O 16-bit is an active-low signal generated by an open-collector driver on I/O boards. IO16* is a dynamic bus size determination signal. If IO16* is active, it signals the Permanent or Temporary Master that the I/O board is capable of 16-bit I/O data transfers. The slave I/O board decodes enough of address lines A0-15 to assert IO16* for its I/O size and must include IORQ* in the decode. This is compatible with STD-80 series boards because they do not drive this signal active and the inactive level defines an 8-bit data transfer. A Temporary Master must manage IO16* when in control of bus resources. The Permanent Master must terminate IO16* with a 316Ω ± 7% resistor to +5V.

MEM16*
(E Pin 53)

Memory 16-bit is an active-low signal generated by memory boards with an open-collector driver. MEM16* is a dynamic bus size determination signal that when active, signals to the Permanent or Temporary Master that the memory board is capable of 16-bit memory data transfers. The slave memory board decodes enough of address lines A14-23 to assert MEM16* for its memory size, which is a minimum of 16 Kbyte. This is compatible with STD-80 series boards because they do not drive this signal active and the inactive level defines an 8-bit data transfer. A Temporary Master must manage the MEM16* signal when in control of bus resources. The Permanent Master must terminate MEM16* with a 316Ω ± 7% resistor to +5V.

Note: Master implementations must allow sufficient time for MEM16* to be generated by a memory slave. A14 through A23 must be generated in time for tD2a and tD2b to be satisfied by a memory slave. Meeting only the setup time to ALE* rising (tS1, tS2) is not sufficient for satisfying address generation requirements.

CHAPTER 2

Peripheral Timing Control

Peripheral Timing Control lines provide control signals that enable Permanent or Temporary Masters to use the P bus to service their own peripheral devices. The P bus includes four control signals that are tailored to interface the STD 32 series bus master with associated STD-80 series peripherals.

ALE*
(P Pin 38)

Address Latch Enable is an active-low signal generated by a three-state driver on the Permanent or Temporary Master. The primary function of ALE* is to provide memory boards with a signal for latching the time multiplexed address from the data bus at the start of a memory cycle. The Permanent Master must release ALE* in response to a bus request (BUSRQ*) from a Temporary Master. The Temporary Master must drive ALE* when in control of bus resources. ALE* does not indicate the start of a machine cycle. Slave boards should not latch A0-7 with ALE*. The Permanent Master must terminate ALE* with a 1.8kΩ ±10% resistor to +5V.

Note that when a processor is accessing an 8-bit device with a 16-bit data instruction, it must perform two sequential 8-bit transfers. The processor will typically increment A0 (from 0 to 1) on the second transfer but will not necessarily issue an ALE* if the A16-23 has not changed.

STATUS1*
(P Pin 39)

Status line 1 is an active-low signal generated by a three-state driver on the Permanent or Temporary Master. The STATUS1* signal, combined with STATUS0*,IORQ*, and MEMRQ*, defines the type of bus cycle currently executing. Refer to Table 2-6 "STD-80 Bus Cycle Types" for the possible cycle types and the state of STATUS0*, STATUS1*, IORQ*, and MEMRQ* for each. The Permanent Master must release STATUS1* in response to a bus request (BUSRQ*) from a Temporary Master. The Temporary Master must drive STATUS1* when in control of bus resources. The Permanent Master must terminate STATUS1* with a 1.8kΩ ±10% resistor to +5V.

STATUS0*
(P Pin 40)

Status line 0 is an active-low signal generated by a three-state driver on the Permanent or Temporary Master. The STATUS0* signal, combined with STATUS1*, IORQ*, and MEMRQ*, defines the type of machine cycle currently executing. Refer to Table 2-6 “STD-80 Bus Cycle Types” for the possible machine cycle types and the state of STATUS0*, STATUS1*, IORQ*, and MEMRQ* for each. The Permanent Master must release STATUS0* in response to a bus request (BUSRQ*) from a Temporary Master. The Temporary Master must drive STATUS0* when controlling bus resources. The Permanent Master must terminate STATUS0* with a 1.8kΩ ±10% resistor to +5V.

Table 2-6. STD-80 Bus Cycle Types.

CYCLE TYPE:	STD BUS SIGNAL (PIN)			
	IORQ* (33)	**MEMRQ* (34)**	**STATUS1* (39)**	**STATUS0* (40)**
Interrupt Acknowledge	low	high	low	low
Read I/O Port	low	high	low	high
Write I/O Port	low	high	high	low
Halt	low	high	high	high
Code Access (Fetch)	high[1]	low	low	low
Read Memory	high[1]	low	low	high
Write Memory	high[1]	low	high	low
Reserved	all remaining combinations			

[1]Memory boards should not use inverted IORQ* as a substitute for MEMRQ*.

Interrupt and Bus Control

Interrupt and bus control lines provide a mechanism for implementing the control schemes listed below.

- Multiprocessing
- Interrupt driven data transfers
- Machine cycle extension for slower devices

BUSAK*
(P Pin 41)

Bus Acknowledge is an active-low signal driven by the current bus master. BUSAK* can be used to signal a bus ownership change to another bus master or to indicate a DMA acknowledge. BUSAK* is generated in response to BUSRQ*. BUSAK* can be driven by a totem-pole driver or three-state driver. BUSAK* must be actively driven to either logic level and not allowed to float high. The Permanent Master must terminate BUSAK* with a 316Ω ± 7% resistor to +5V.

BUSRQ*
(P Pin 42)

Bus Request is an active-low signal driven by either a Temporary Master for bus ownership or a DMA slave for DMA request. BUSRQ* is driven by an open collector driver. This signal may be driven asynchronously, and it is the responsibility of the receiving agent to synchronize BUSRQ*. Temporary Masters must manage the signals shown in Table 2-7 "Signals Standard Architecture Temporary Masters Must Manage". The Permanent Master must terminate BUSRQ* with a 316Ω ± 7% resistor to +5V.

Table 2-7. Signals Standard Architecture Temporary Masters Must Manage.

8-Bit (SA8) Signal Mnemonics		16-Bit (SA16) Signal Mnemonics	
D0-D7	A0-A23	D0-D16	A0-A23
WR*	RD*	WR*	RD*
IORQ*	MEMRQ*	IORQ*	MEMRQ*
BHE*	ALE*	BHE*	ALE*
STATUS1*	STATUS0*	STATUS1*	STATUS0*
BUSRQ*	BUSAK*	BUSRQ*	BUSAK*
IOEXP	INTAK*	IOEXP	INTAK*
WAITRQ*		WAITRQ*	
		IO16*	MEM16*

INTAK*
(P Pin 43)

Interrupt Acknowledge is an active-low signal generated by a three-state driver on the Permanent or Temporary Master. Any board can request interrupt service by generating an interrupt request (INTRQ* or INTRQ1*). The interrupt request is routed to the Permanent or Temporary Master to invoke an interrupt procedure designed to service the request. The master responsible for the request responds by generating an INTAK*. In a vectored interrupt system, the master also generates a unique cascade address if the interrupt request is configured for cascade operation.

IORQ* should be driven active during the INTAK* cycle for STD-80 compatibility.

INTRQ*
(P Pin 44),
INTRQ1*
(P Pin 37),
and
INTRQ3*
(E Pin 67)

Interrupt Request, Interrupt Request 1, and Interrupt Request 3 are active-low signals generated by an open-collector driver on any board. It is the responsibility of the Permanent or Temporary Master to synchronize INTRQ*, INTRQ1*, and INTRQ3*. Any board can generate an active INTRQ*, INTRQ1*, or INTRQ3* to invoke an interrupt procedure typically designed to service an asynchronous event. A common practice is to connect INTRQ*, INTRQ1*, and INTRQ3* to maskable interrupt request inputs on an interrupt controller located on the Permanent or Temporary Master. The Permanent Master must terminate INTRQ*, INTRQ1*, and INTRQ3* with 316Ω ± 7% resistors to +5V.

Note that the CNTRL* signal may be defined as INTRQ2* (P pin 50).

Since it is possible to have multiple interrupt sources on the same interrupt request, it is recommended that each interrupt source include a status bit to determine if the interrupt is active. It is also recommended that each shared interrupt signal include readback to ensure that all sources have been serviced. These design practices eliminate the possibility of missing one interrupt request that is generated while another interrupt request is being serviced in an edge-triggered mode of operation. Although multiple sources may share an interrupt request, the same interrupt request should not be shared between multiple masters in a system.

WAITRQ*
(P Pin 45)

Wait Request is an active-low signal generated by an open-collector driver on memory or I/O boards. It is the responsibility of the Permanent and Temporary Masters to synchronize WAITRQ*. The asserted (low) condition must be synchronized by two flip-flops and the ready (high) condition must be synchronized by one flip-flop (similar to an 8284A). See Figure 2-2 "WAITRQ* Synchronization". Memory or I/O boards can assert WAITRQ* to extend the amount of time data is valid during a write cycle or to extend the amount of time allowed to return the data during a read cycle.

Warning: With WAITRQ* active, the Permanent Master may not be able to service interrupt requests, DMA requests, asynchronous data transfers, or dynamic RAM refresh. Masters must be able to complete on-board cycles with WAITRQ* asserted on the STD bus. For these reasons, it is best not to keep wait request asserted for more than 10µs. The Permanent Master must terminate WAITRQ* with a 316Ω ± 7% resistor to +5V. It is recommended that peripheral boards drive WAITRQ* only during SA cycles.

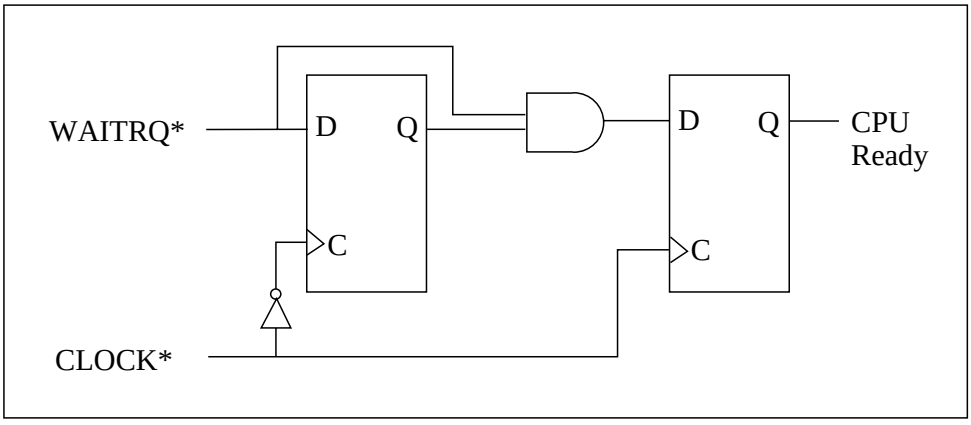


Figure 2-2. WAITRQ* Synchronization.

NMIRQ*
(P Pin 46)

Non-Maskable Interrupt Request is an active-low signal generated by an open-collector driver on any board other than the Permanent Master. NMIRQ* is an edge-triggered signal that must be held active a minimum of 1µs. It is the responsibility of the Permanent Master to synchronize NMIRQ* if required by the processor. Any board other than the Permanent Master generates an active NMIRQ* to invoke an interrupt procedure typically designed to service a catastrophic system error such as a power failure or a DRAM parity error. The Permanent Master should terminate NMIRQ* with a 316Ω ± 7% resistor to +5V.

Since it is possible to have multiple interrupt sources on the same non-maskable interrupt request, it is recommended that each interrupt source include a status bit to determine if the interrupt is active. It is also recommended that NMIRQ* include readback to ensure that all sources have been serviced. These design practices eliminate the possibility of missing one interrupt request that is generated while another interrupt request is being serviced in an edge-triggered mode of operation.

PCO
(P Pin 51)

Priority Chain Out is an active-high signal generated by a totem-pole driver on any board supporting the serial priority resolution scheme of STD-80. STD 32 peripherals do not use the PCI-PCO feature. PCI and PCO should be connected together on STD 32 designs.

PCI
(P Pin 52)

Priority Chain In is an active-high signal received by any board supporting the serial priority resolution scheme of STD-80. STD 32 peripherals do not use the PCI-PCO feature. PCI and PCO should be connected together on STD 32 designs.

DCPDN*
(P Pin 6)

DC Power Down is an active-low signal generated by an open-collector driver on any board in the system. The DCPDN* signal should become active when the digital supply drops below an acceptable operating range of approximately 4.75V. Once active, DCPDN* must monotonically decrease (continuously drop without glitching) until the digital supply drops below 0.8V. Typical applications of DCPDN* include memory protection to prevent corruption of battery-backed devices and shutdown of low power CMOS devices into a sleep mode. The board generating DCPDN* must terminate it with a 10kΩ ± 10% resistor to +5V.

Clock and Reset

The P connector includes several signals with clock and reset information. The clock signals are not necessarily synchronized to the operation of the Permanent Master. The reset signals are used to place hardware into a known state.

SYSRESET*
(P Pin 47)

System Reset is an active-low signal generated by a totem-pole driver on the Permanent Master.

SYSRESET is normally reset:

- In response to a switch closure (refer to PBRESET*)
- In response to a watchdog timer time out
- When digital power falls below an acceptable operating level of approximately 4.75V

A board generating SYSRESET* should activate this signal before the digital power supply reaches 1V and keep it active for a minimum of 1ms after the digital power supply reaches an acceptable operating level of approximately 4.75V.

PBRESET*
(P Pin 48)

Push Button Reset is an active-low signal generated by a switch closure or an open-collector driver from any board in the system. It is the responsibility of all boards receiving PBRESET* to debounce it. The Permanent Master must terminate PBRESET* with a 316Ω ± 7% pullup resistor to +5V.

CLOCK*
(P Pin 49)

The Clock signal is an active-low signal generated by a totem-pole driver on the Permanent Master. CLOCK* is an 8MHz nominal, 8.33MHz maximum, periodic signal and is not necessarily synchronized to the operation of the Permanent or Temporary Master's processor.

CNTRL*
(P Pin 50)

Control is an auxiliary circuit for special clock timing. Implementations of CNTRL* include:

- A clocking signal that is equal in frequency to some multiple of the CLOCK* signal
- A third maskable interrupt request, INTRQ2*

It is recommended that boards using this signal make the connection through a jumper or cuttable trace. If INTRQ2* is supported, the Permanent Master must terminate this signal with a 316Ω ± 7% resistor to +5V, and should be generated with an open collector driver.

Frontplane DMA Control

Frontplane DMA control signals manage the transfer of data between memory and I/O slaves in the system. These optional signals are described in detail below. All signals are implemented in a 10-pin latching ribbon cable header located on the top (frontplane) of the board with the pin assignments illustrated in Table 2-8 "Frontplane DMA Pinouts". Signal flow is relative to the DMA control logic.

Table 2-8. Frontplane DMA Pinouts.

ODD PINS				EVEN PINS			
Pin	**Mnemonic**	**Signal Flow**	**Description**	**Pin**	**Mnemonic**	**Signal Flow**	**Description**
1	$\overline{GND}$	In	Logic Ground	2	$\overline{DRQ}$	In	DMA Request
3	$\overline{GND}$	In	Logic Ground	4	$\overline{DAK}$	Out	DMA Acknowledge
5	$\overline{GND}$	In	Logic Ground	6	$\overline{DMAW}$	Out	DMA Write
7	$\overline{GND}$	In	Logic Ground	8	$\overline{DMAR}$	Out	DMA Read
9	$\overline{GND}$	In	Logic Ground	10	$\overline{T/C}$	Out	Terminal Count

$\overline{DRQ}$ (Pin 2)

DMA Request is an active-low signal generated by a totem-pole driver on any board, typically an I/O slave. $\overline{DRQ}$ is driven by an I/O slave to request a DMA transfer. The I/O slave must hold $\overline{DRQ}$ active until the frontplane DMA acknowledge signal $\overline{DAK}$ is received. Each byte transferred requires a $\overline{DRQ}$/$\overline{DAK}$ sequence.

$\overline{DAK}$ (Pin 4)

DMA Acknowledge is an active-low signal generated by a totem-pole driver from the system DMA control logic, typically a Permanent or Temporary Master. $\overline{DAK}$ is driven in response to a DMA request $\overline{DRQ}$ to signal the beginning of a DMA operation.

$\overline{DMAW}$ (Pin 6)

DMA Write is an active-low signal generated by a totem-pole driver from the system DMA control logic, typically a Permanent or Temporary Master. An active $\overline{DMAW}$ indicates a DMA read cycle, which is a data transfer from memory to I/O. The I/O slave must qualify $\overline{DMAW}$ with $\overline{DAK}$ before participating in the transfer.

$\overline{DMAR}$ (Pin 8)

DMA Read is an active-low signal generated by a totem-pole driver from the system DMA control logic, typically a Permanent or Temporary Master. An active $\overline{DMAR}$ indicates a DMA write cycle, which is a data transfer from I/O to memory. The I/O slave must qualify $\overline{DMAR}$ with $\overline{DAK}$ before participating in the transfer.

$\overline{T/C}$ (Pin 10)

Terminal count is an active-low signal generated by a totem-pole driver from the system DMA control logic. An active $\overline{T/C}$ indicates that a predetermined number of data bytes have completed transferring. An I/O slave must qualify $\overline{T/C}$ with $\overline{DAK}$ to determine if $\overline{T/C}$ is active.

2.3. IMPLEMENTATIONS

2.3.1. PERMANENT AND TEMPORARY MASTERS

Memory Refresh

Masters in control of the bus must not drive MEMRQ* or ALE* during on-board refresh cycles. Masters may drive other bus control signals such as RD*, STATUS1*, and STATUS0* and may drive the address bus during an onboard refresh cycle.

Bus Termination

The Permanent Master must resistively terminate all signals that could be managed by a Temporary Master in response to BUSRQ* or DREQx* as listed in Table 2-7 “Signals Standard Architecture Temporary Masters Must Manage”. Resistive termination should be 1.8kΩ ± 10% pullup to +5V, unless otherwise specified.

Optional Signal Management

Only one additional Temporary Master is allowed to gain control of the bus in a traditional STD-80 system utilizing BUSRQ*. A true multiple SA master system that is STD-80 compatible is possible through the use of DREQx*, DAKx*, and a Slot X arbiter. A Permanent or Temporary Master utilizing these optional signals would have the DX compliance designation. See Chapter 3 for a signal description and Figure 2-10 "(SA) Master Request Acknowledge Timing", for a timing diagram.

Temporary Masters may also manage INTRQ*, INTRQ1*, CNTRL* (INTRQ2*), or INTRQ3*. In a vectored interrupt system, care must be taken not to assign the same interrupt request level to more than one interrupt source. This eliminates the possibility of two interrupt sources responding to the same cascade address.

Table 2-9. Signals Standard Architecture Temporary Masters Optionally Manage.

Pin Number	Signal Mnemonic
E16	DREQx*
E15	DAKx*
P44	INTRQ*
P37	INTRQ1*
P50	CNTRL* (INTRQ2*)
E67	INTRQ3*

2.3.2. I/O SLAVES

I/O slaves desiring to be compatible with future systems should implement interrupts with IRQx as well as INTRQ*, INTRQ1*, CNTRL* (INTRQ2*), or INTRQ3*. IRQx is a positive true edge-triggered or low-level asserted Slot X interrupt request. See Chapter 3 for a signal description.

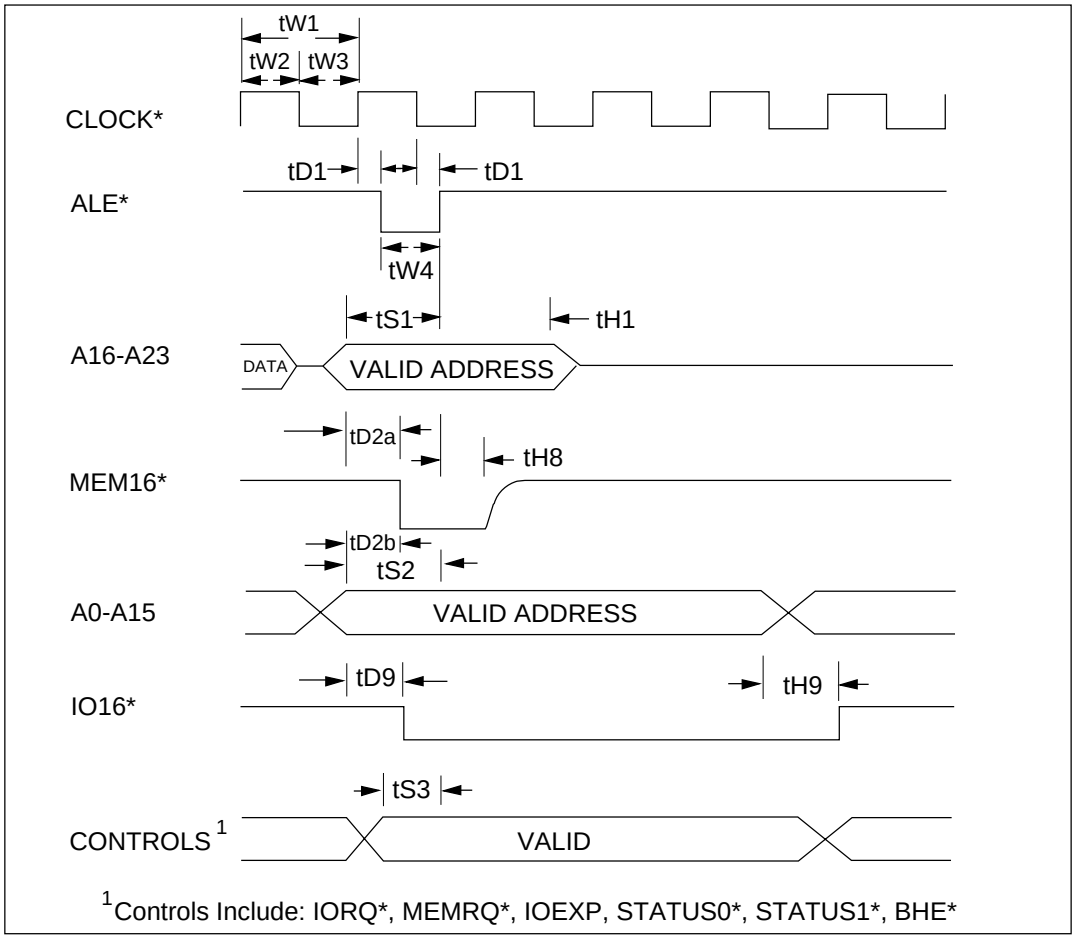
2.3.3. 32-BIT SA DATA TRANSFERS

SA compatible bus cycles may include 8- or 16-bit data transfers. 32-bit SA data transfers are not allowed.

2.4. TIMING

The following figures illustrate the critical timing parameters from the point of view of the Permanent Master. The timing diagrams are divided into the following groups for ease of use:

- Status Timing
- Read Timing
- Write Timing
- Interrupt Timing
- Fly-By DMA Timing
- Wait State Timing
- Bus Exchange Timing
- SA Master Request/Acknowledge Timing

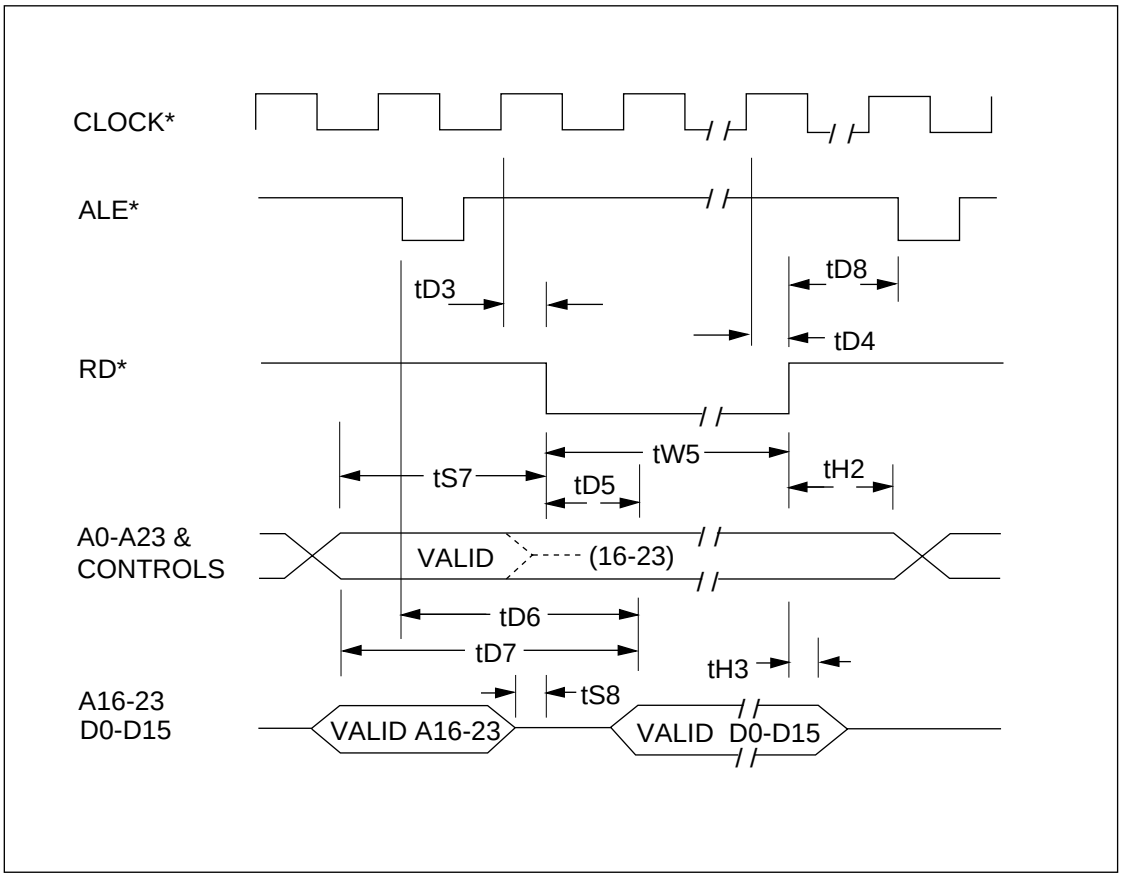


	MIN	MAX	DESCRIPTION
tD1	1	70	DELAY FROM CLOCK* TO ALE* LOW OR HIGH
tD2a		150	DELAY FROM A17-23 TO MEM16* VALID (128 KBYTE BLOCK)
tD2b		150	DELAY FROM A14-16 TO MEM16* VALID (16 KBYTE BLOCK)
tD9		150	DELAY FROM A0-A15 VALID TO IO16*
tH1	25		A16-A23 HOLD AFTER ALE* ↑
tH8	5		MEM16* HOLD AFTER ALE* ↑
tH9	5		IO16* HOLD AFTER A0-15 INVALID
tS1	85		A16-A23 SETUP TO ALE* ↑ (SEE PAGE 2-12 FOR MEM16* DETAILS)
tS2	85		A0-A15 SETUP TO ALE* ↑
tS3	75		CONTROLS SETUP TO ALE* ↑
tW1	124	126	CLOCK* PERIOD (8MHz)
tW2	52	73	CLOCK* HIGH WIDTH
tW3	52	73	CLOCK* LOW WIDTH
tW4	50		ALE* LOW WIDTH

All times given in nanoseconds.

Figure 2-3. Status Timing.

CHAPTER 2

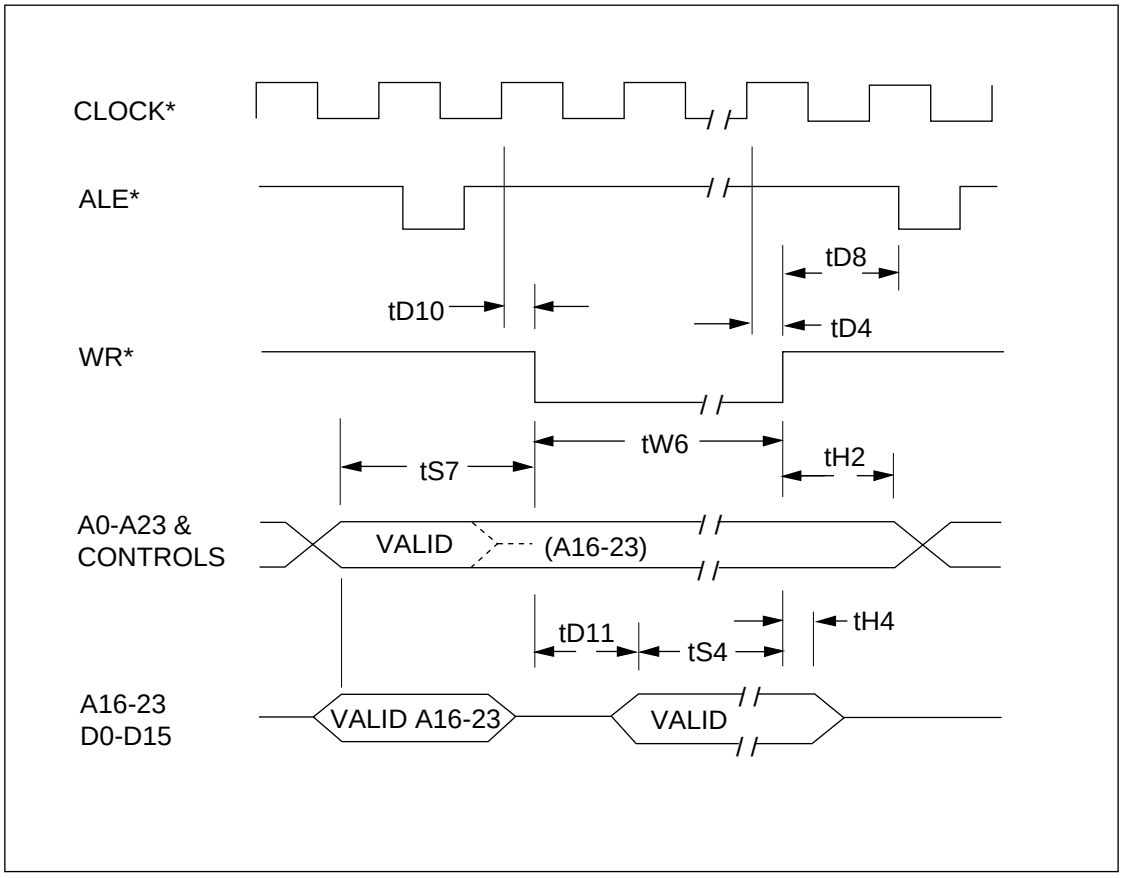


	MIN	MAX	DESCRIPTION
tD3	1	80	DELAY FROM CLOCK* ↑ TO RD* LOW
tD4	1	110	DELAY FROM CLOCK* ↑ TO RD* HIGH
tD5		140	DELAY FROM RD* LOW TO D0-D15 VALID
tD6[1]		350	DELAY FROM ALE* LOW TO D0-D15 VALID
tD7		400	DELAY FROM A0-A23 VALID TO D0-D15 VALID
tD8	175		DELAY FROM RD* HIGH TO ALE* LOW
tH2	55		A0-A15 HOLD AFTER RD* HIGH
tH3	10	125	D0-D15 HOLD AFTER RD* HIGH
tS7	150		A0-23 SETUP TO RD* LOW
tS8	1		A16-23 FLOAT BEFORE RD* LOW
tW5	210		RD* LOW WIDTH

All times given in nanoseconds.

[1] tD6 provided for reference only since ALE* may not occur on cycles that do not change or use A16-A23. Designs using A16-23 should use the smaller of tD6 or tD7 for timing.

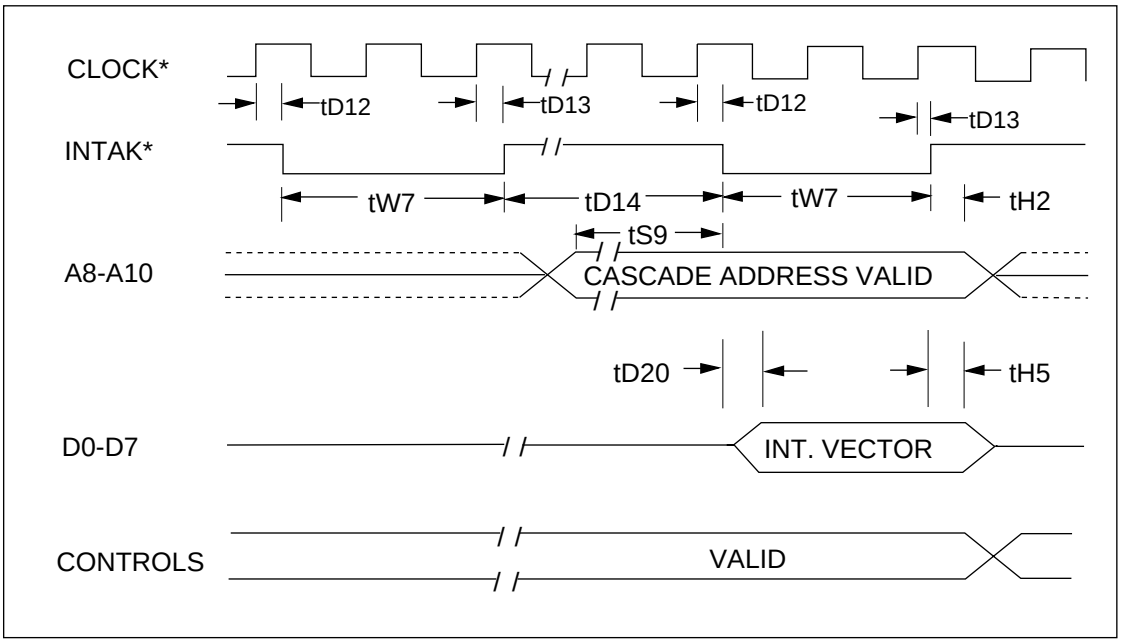
Figure 2-4. Read Timing.



	MIN	MAX	DESCRIPTION
tD4	1	110	DELAY FROM CLOCK* ↑ TO WR* HIGH
tD8	175		DELAY FROM WR* HIGH TO ALE* LOW
tD10	1	70	DELAY FROM CLOCK* ↑ TO WR* LOW
tD11		50	DELAY FROM WR* LOW TO D0-15 VALID
tH2	55		A0-A15 HOLD AFTER WR*
tH4	25		D0-15 HOLD AFTER WR* ↑
tS4	160		D0-D15 SETUP TO WR* ↑
tS7	150		A0-23 SETUP TO WR* LOW
tW6	210		WR* LOW WIDTH

All times given in nanoseconds.

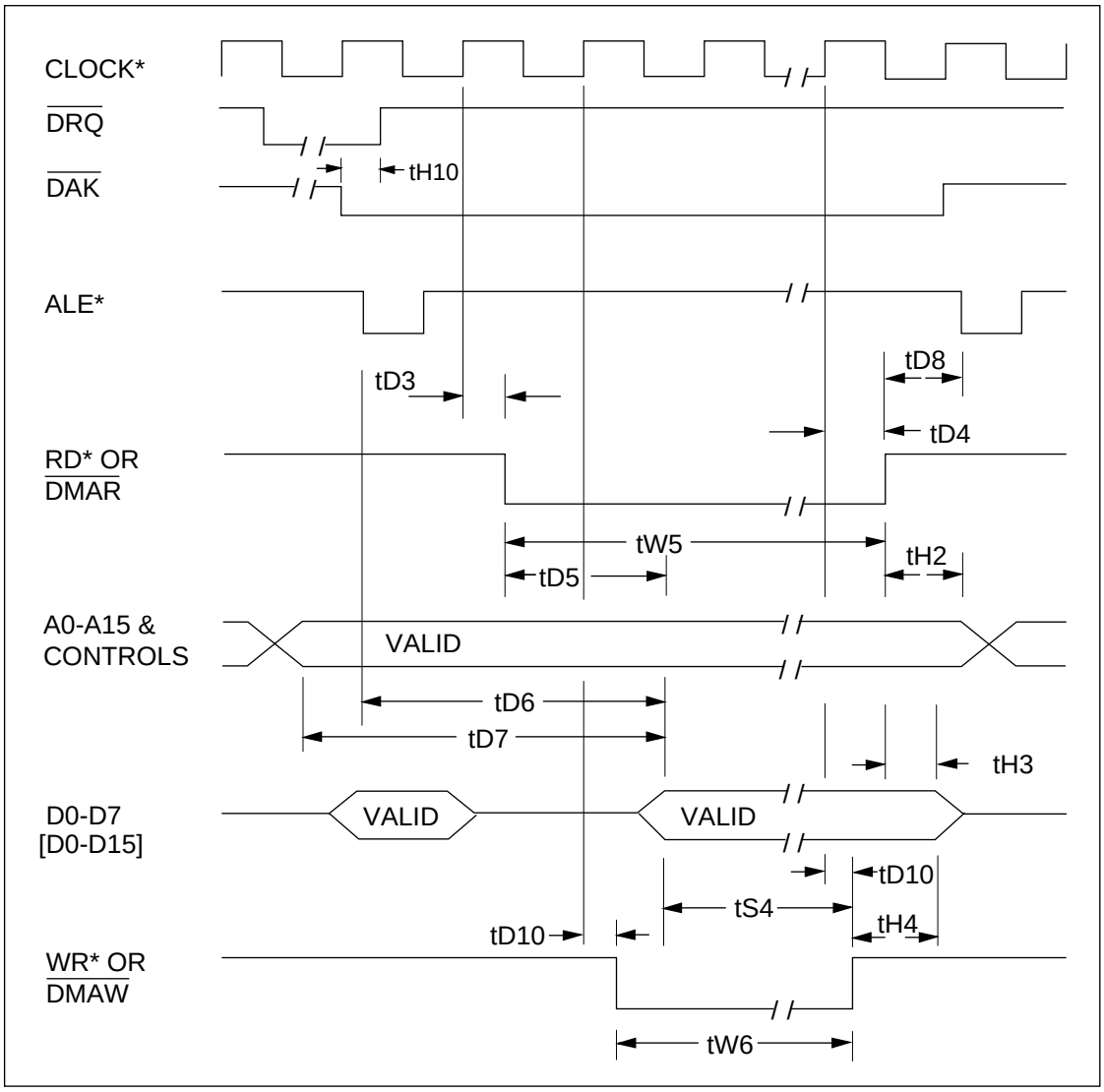
Figure 2-5. Write Timing.



	MIN	MAX	DESCRIPTION
tD12	1	70	DELAY FROM CLOCK* ↑ TO INTAK* LOW
tD13	1	70	DELAY FROM CLOCK* ↑ TO INTAK* HIGH
tD14	400		FIRST INTAK* HIGH TO SECOND INTAK* LOW
tD20		125	DELAY FROM SECOND INTAK ↓ TO INTERRUPT VECTOR
tH2	55		CASCADE ADDRESS HOLD AFTER INTAK* ↑
tH5	10	125	INTERRUPT VECTOR HOLD AFTER INTAK* ↑
tS9	75		CASCADE ADDRESS VALID TO SECOND INTAK* ↓
tW7	210		INTAK* WIDTH

All times given in nanoseconds.

Figure 2-6. Interrupt Acknowledge Timing.



	MIN	MAX	DESCRIPTION
tH10	1	50	DRQ HOLD AFTER DACK ↓

All times given in nanoseconds.

Notes:

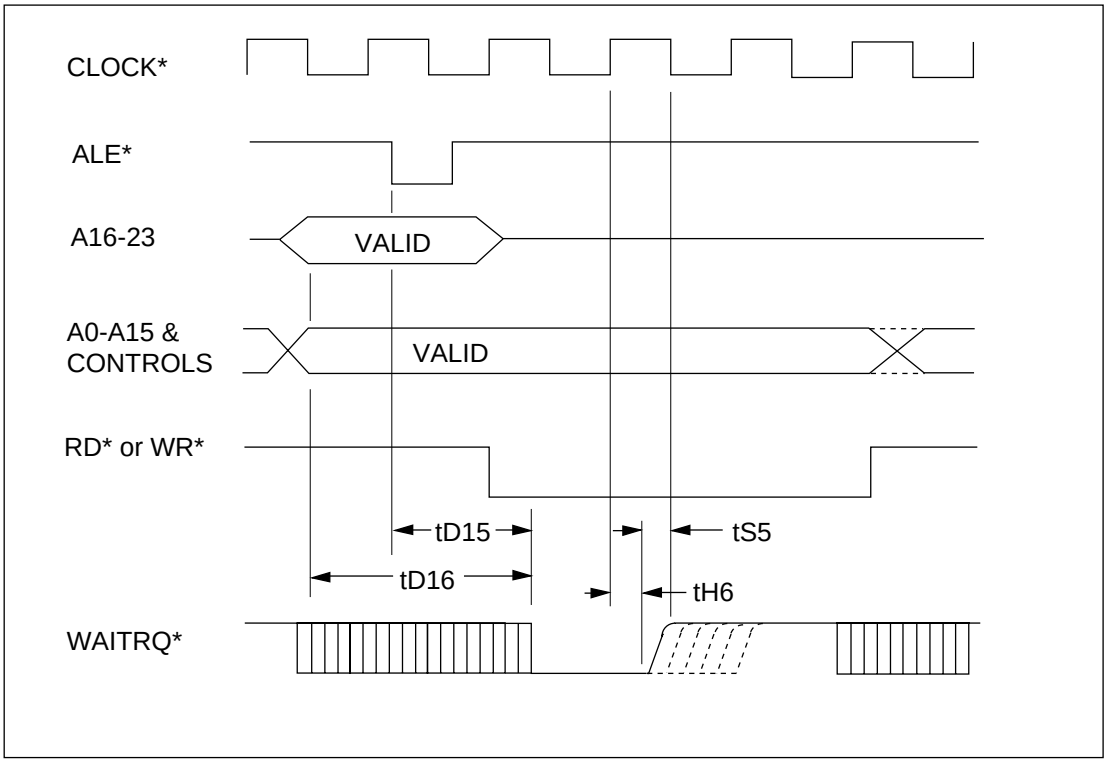
The master must provide sufficient wait states to allow the device being written-to to assert WAITRQ* in the normal manner (with respect to WR* or DMAW).

For memory-to-I/O transfer, RD* and DMAW will be used. For I/O-to-memory transfers, DMAR* and WR* will be used.

DMAR and DMAW are frontplane signals.

Figure 2-7. Fly-By DMA Timing.

CHAPTER 2



	MIN	MAX	DESCRIPTION [1]
tD15		60	SLAVE DELAY FROM ALE* TO WAITRQ* LOW [2, 4]
tD16		130	SLAVE DELAY FROM A0-A23 VALID TO WAITRQ* LOW [2, 4]
tH6	5		WAITRQ* HOLD AFTER CLOCK* ↑
tS5	45		WAITRQ* SETUP TO CLOCK* ↓ [3, 4]

All times given in nanoseconds.

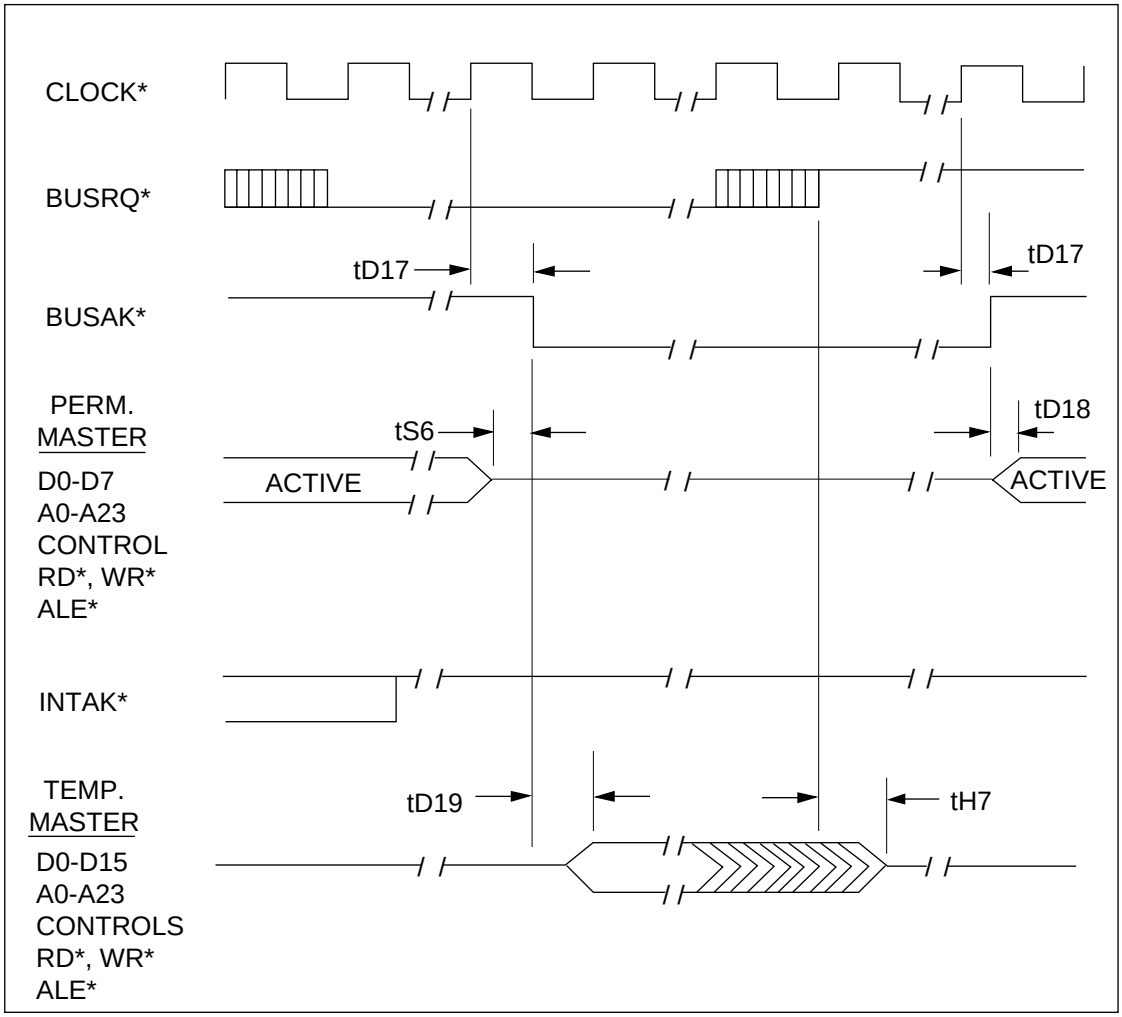
[1] Masters must be capable of inserting two wait states to guarantee PC channel ready timing (CHRDY) compatibility.

[2] tD15 provided for reference only since ALE* may not occur on cycles that do not change or use A16-A23. Designs using A16-23 should use tD15 or tD16 for timing, depending on how fast the logic is. WAITRQ* must by synchronous to CLOCK*.

[3] If this time is met, the slave is guaranteed to not get an extra wait state. However, the masters provide two flip-flop (or equiv.) synchronization, so no metastability errors will occur if tS5 is violated (dashed lines).

[4] tS5, tD15, tD16 include 36ns backplane delay and is based on a 300 Ohm pullup to +5V and a maximum of 175 pF on the WAITRQ* line (5 boards @ 20pF/board + 24 connectors @ 2pF/connector + 18 inch trace @ 1.5 pF/inch).

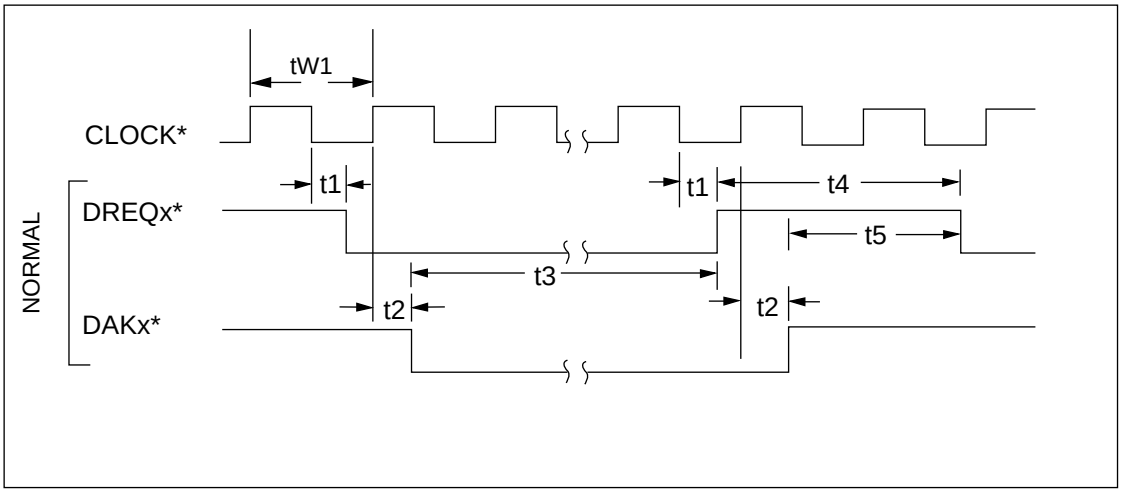
Figure 2-8. Wait State Timing.



	MIN	MAX	DESCRIPTION
tD17	1	50	DELAY FROM CLOCK* ↑ TO BUSAK* ↓
tD18	1		DELAY FROM BUSAK* HIGH TO PERM. MASTER ACTIVE
tD19	1		DELAY FROM BUSAK* LOW TO TEMP. MASTER ACTIVE
tH7		125	TEMP. MASTER ACTIVE AFTER BUSRQ* HIGH
tS6	1		PERM. MASTER FLOAT BEFORE BUSAK* LOW

All times given in nanoseconds.

Figure 2-9. Bus Exchange Timing.



	MIN	MAX	DESCRIPTION
tW1	124	126	CLOCK* PERIOD (8 MHz)
t1	2	33	CLOCK* ↓ TO DREQx* HIGH OR LOW
t2	2	40	CLOCK* ↑ TO DAKx* HIGH OR LOW
t3		64tW1	DAKx* ↑ TO DREQx* HIGH (MAX TIME TO RELEASE BUS)
t4	2tW1		DREQx* ↑ TO DREQx* LOW
t5	0		DAKx* ↑ TO DREQx* LOW

All times given in nanoseconds.

Figure 2-10. (SA) Master Request Acknowledge Timing.

C H A P T E R 3

EXTENDED ARCHITECTURE BUS CYCLES

3.1. INTRODUCTION

The STD 32 Bus Specification defines the mechanical and electrical parameters needed to support 8-, 16-, and 32-bit EISA-like microprocessor systems, referred to as Extended Architecture (EA) transfers. This chapter expands on the STD-80 subset discussed in Chapter 2 to include the signals required for a 16- or a 32-bit STD 32 compatible design. These signals are summarized in Table 3-1 "Additional Signals for Extended Architecture Bus Cycle Implementation".

EXTENDED ARCHITECTURE BUS CYCLES

CHAPTER 3

Table 3-1. Additional Signals for Extended Architecture Bus Cycle Implementation.

Power	Logic Ground (E Pins 1, 17, 18, 28, 35, 45, 54) +5VDC (E Pins 13, 14, 68)
Data	D16-D31 (E Pins 43, 41, 39, 37, 33, 31, 29, 27, 25, 23, 21, 19, 26, 24, 22, 20)
Address	XA24*-XA31* (E Pins 79, 77, 75, 73, 80, 78, 76, 74) XA16-XA23 (E Pins 9, 7, 5, 3, 10, 8, 6, 4)[1]
Control	NOWS* (E Pin 11) RSVD (E Pin 12) MAKx* (E Pin 69) MREQx* (E Pin 70) MASTER16* (E Pin 46) IRQx (E Pin 47) AENx* (E Pin 48) BE0*-BE3* (E Pins 51, 49, 52, 50) MEM16* (E Pin 53) M-IO (E Pin 55) W-R (E Pin 56) DMAIOW* (E Pin 57) DMAIOR* (E Pin 58) IO16* (E Pin 59) EX8* (E Pin 60) CMD* (E Pin 61) START* (E Pin 62) EX16* (E Pin 63) EX32* (E Pin 64) EXRDY (E Pin 65) T-C (E Pin 66) LOCK* (E Pin 2) DAKx* (E Pin 15) DREQx* (E Pin 16) SLBURST* (E Pin 71) MSBURST* (E Pin 72)

[1]These signals have been relocated from their multiplexed location on the STD-80 bus for pipelined operation.

3.2. SIGNAL PIN ASSIGNMENTS

The complete set of STD 32 signals is shown in Table 3-2 "STD 32 Bus P Pinout" and Table 3-3 "STD 32 Bus E Pinout." This chapter discusses the signals that expand the STD-80 subset to an STD 32 implementation.

Table 3-2. STD 32 Bus P Pinout.

COMPONENT SIDE				CIRCUIT SIDE			
Pin	**Mnemonic**	**Signal Flow**	**Description**	**Pin**	**Mnemonic**	**Signal Flow**	**Description**
1	+5VDC	In	Logic Power	2	+5VDC	In	Logic Power
3	GND	In	Logic Ground	4	GND	In	Logic Ground
5	VBAT	Bidir	Battery Power	6	DCPDN*	Bidir	DC Power Down
7	A19/D3	Out/Bidir	Address/Data	8	A23/D7	Out/Bidir	Address/Data
9	A18/D2	Out/Bidir	Address/Data	10	A22/D6	Out/Bidir	Address/Data
11	A17/D1	Out/Bidir	Address/Data	12	A21/D5	Out/Bidir	Address/Data
13	A16/D0	Out/Bidir	Address/Data	14	A20/D4	Out/Bidir	Address/Data
15	A7	Out	Address	16	A15	Out	Address
17	A6	Out	Address	18	A14	Out	Address
19	A5	Out	Address	20	A13	Out	Address
21	A4	Out	Address	22	A12	Out	Address
23	A3	Out	Address	24	A11	Out	Address
25	A2	Out	Address	26	A10	Out	Address
27	A1	Out	Address	28	A9	Out	Address
29	A0	Out	Address	30	A8	Out	Address
31	WR*	Out	Write Mem or I/O	32	RD*	Out	Read Mem or I/O
33	IORQ*	Out	I/O Address Select	34	MEMRQ*	Out	Mem Address Select
35	IOEXP	Out	I/O Expansion	36	BHE*	Out	Byte High Enable
37	INTRQ1*	In	Interrupt Request 1	38	ALE*	Out	Address Latch Enable
39	STATUS1*	Out	CPU Status 1	40	STATUS0*	Out	CPU Status 0
41	BUSAK*	Out	Bus Acknowledge	42	BUSRQ*	In	Bus Request
43	INTAK*	Out	Interrupt Acknowledge	44	INTRQ*	In	Interrupt Request
45	WAITRQ*	In	Wait Request	46	NMIRQ*	In	Non-Mask Int Request
47	SYSRESET*	Out	System Reset	48	PBRESET*	In	Push-Button Reset
49	CLOCK*	Out	Clock	50	CNTRL*	Bidir	Aux Timing
51	PCO	Out	Priority Chain Out	52	PCI	In	Priority Chain In
53	AUX GND	In	AUX Ground (bussed)	54	AUX GND	In	AUX Ground (bussed)
55	AUX +V	In	AUX Positive (+12VDC)	56	AUX -V	In	AUX Neg. (-12VDC)

Notes: An asterisk (*) indicates a low level active signal.
Address lines A16-A23 are multiplexed on data lines D0-D7 on each address cycle for STD-80 compatibility.
PCO and PCI are not typically used on peripheral cards. All boards not supporting PCO and PCI should tie these two signals together.

CHAPTER 3

Table 3-3. STD 32 Bus E Pinout.

COMPONENT SIDE				CIRCUIT SIDE			
Pin	**Mnemonic**	**Signal Flow**	**Description**	**Pin**	**Mnemonic**	**Signal Flow**	**Description**
1	GND	In	Logic Ground	2	LOCK*	Out	Lock
3	XA19	Out	Address	4	XA23	Out	Address
5	XA18	Out	Address	6	XA22	Out	Address
7	XA17	Out	Address	8	XA21	Out	Address
9	XA16	Out	Address	10	XA20	Out	Address
11	NOWS*	In	No Wait States	12	RSVD	--	Reserved
13	+5VDC	In	Logic Power	14	+5VDC	In	Logic Power
15	DAKx*	Out	DMA Acknowledge	16	DREQx*	In	DMA Request
17	GND	In	Logic Ground	18	GND	In	Logic Ground
19	D27	Bidir	Data	20	D31	Bidir	Data
21	D26	Bidir	Data	22	D30	Bidir	Data
23	D25	Bidir	Data	24	D29	Bidir	Data
25	D24	Bidir	Data	26	D28	Bidir	Data
27	D23	Bidir	Data	28	GND	In	Logic Ground
29	D22	Bidir	Data	30	D15	Bidir	Data
31	D21	Bidir	Data	32	D14	Bidir	Data
33	D20	Bidir	Data	34	D13	Bidir	Data
35	GND	In	Logic Ground	36	D12	Bidir	Data
37	D19	Bidir	Data	38	D11	Bidir	Data
39	D18	Bidir	Data	40	D10	Bidir	Data
41	D17	Bidir	Data	42	D9	Bidir	Data
43	D16	Bidir	Data	44	D8	Bidir	Data
45	GND	In	Logic Ground	46	MASTER16*	Out	Master 16-Bit
47	IRQx	In	Interrupt Request	48	AENx*	Out	Address Enable
49	BE1*	Out	Byte Enable 1	50	BE3*	Out	Byte Enable 3
51	BE0*	Out	Byte Enable 0	52	BE2*	Out	Byte Enable 2
53	MEM16*	In	Memory 16-Bit	54	GND	In	Logic Ground
55	M-IO	Out	Memory or I/O	56	W-R	Out	Write or Read
57	DMAIOW*	Out	DMA I/O Write	58	DMAIOR*	Out	DMA I/O Read
59	IO16*	In	I/O 16-Bit	60	EX8*	In	Exchange 8-Bit
61	CMD*	Out	Command	62	START*	Out	Start
63	EX16*	In	Exchange 16-Bit	64	EX32*	In	Exchange 32-Bit
65	EXRDY	In	Exchange Ready	66	T-C	Bidir	Terminate or Count
67	INTRQ3*	In	Interrupt Request 3	68	+5VDC	In	Logic Power
69	MAKx*	Out	Master Acknowledge	70	MREQx*	In	Master Request
71	SLBURST*	In	Slave Burst	72	MSBURST*	Out	Master Burst
73	XA27*	Out	Address	74	XA31*	Out	Address
75	XA26*	Out	Address	76	XA30*	Out	Address
77	XA25*	Out	Address	78	XA29*	Out	Address
79	XA24*	Out	Address	80	XA28*	Out	Address

3.2.1. POWER BUS ADDITIONS

(E Pins 1, 13, 14, 17, 18, 28, 35, 45, 54, 68)

Several ground pins have been added to prevent ground bounce. All boards implementing the STD 32 compatible edge fingers should utilize these additional grounds even if no other STD 32 signals are used.

3.2.2. DATA BUS ADDITIONS

(E Pins 43, 41, 39, 37, 33, 31, 29, 27, 25, 23, 21, 19, 26, 24, 22, 20)

The 32 data lines are bidirectional, active-high, and generated with three-state drivers by any board in the system. The data bus direction is controlled by the Permanent or Temporary Master using control bus signals. Any board not currently selected must release the data bus to a high impedance state either by disabling the data bus drivers or pointing the data bus drivers away from the backplane. The Temporary Master manages the data bus when in control of bus resources. The BE0*-BE3* signals define which data bus lines are used during a data transfer. The Permanent Master must provide 1.8kΩ ± 10% pullup resistors to +5V on all data lines.

The 32-Bit extension relocates address lines A16-A23 of the STD-80 bus. This is done to eliminate multiplexing and to support pipelined transfers.

3.2.3. ADDRESS BUS ADDITIONS

(E Pins 9, 7, 5, 3, 10, 8, 6, 4, 79, 77, 75, 73, 80, 78, 76, 74)

The 32 address lines are bidirectional and generated with three-state drivers by the Permanent or Temporary Masters. The upper eight address lines (XA24*-XA31*) are active-low. This permits the use of a resistive termination to set the address bus in a known state for Permanent or Temporary Masters unable to drive all 32 address lines.

The Permanent Master must release XA24*-XA31* to a Temporary Master in response to a master request (MREQx*) and not release XA24*-XA31* in response to a bus request (BUSRQ* or DREQx*). The Temporary Master must drive the address bus when in control of bus resources. A Permanent Master supporting the high order address byte must terminate the signals with 1kΩ ± 10% resistors to +5V.

EA bus cycles relocate address lines A16-A23 of the STD-80 bus to E pins 3-10. This is done to eliminate multiplexing and to support pipelined transfers. ALE* is not used.

3.2.4. CONTROL BUS ADDITIONS

(E Pins 11, 15, 16, 46, 47, 49-53, 55-67, 69-72)

The control bus includes the signals needed to coordinate STD bus activity. The control signals discussed below are added to the STD-80 subset signals to manage 16- and 32-bit data STD 32 bus cycles.

Some of the control bus signals are "slot-specific". A slot-specific signal is a backplane connection from the Slot X connector to one and only one other connector. An example of a slot-specific STD 32 backplane is shown in Figure 3-1 "STD 32 Slot-Specific Backplane". The slot-specific signals are AENx*, MREQx*, MAKx*, DREQx*, DAKx*, and IRQx. In this example, the slot-specific signals are routed from Slot X to Slots 0 through n. Up to 14 slots are supported by the slot-specific signals.

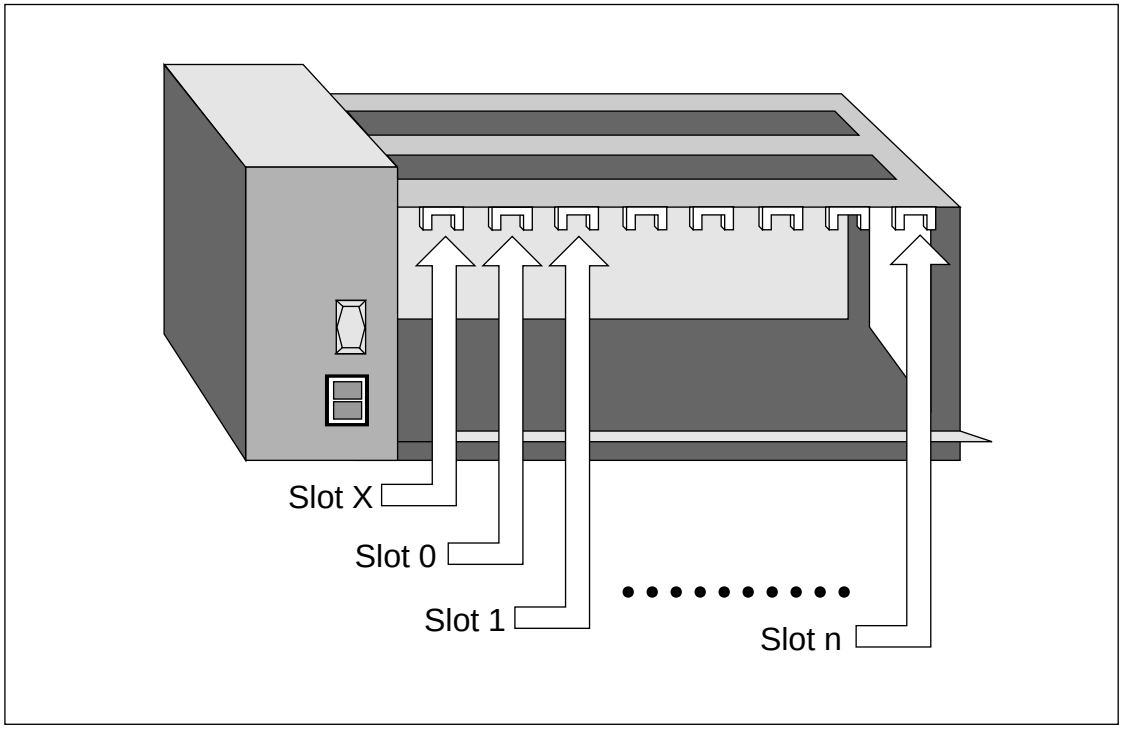


Figure 3-1. STD 32 Slot-Specific Backplane.

Memory and I/O Control

MASTER16*
(E Pin 46)

Master 16-bit is an active-low signal generated with an open-collector driver by Permanent and Temporary Masters. MASTER16* is asserted by a master to indicate a 16-bit data transfer. The Permanent Master must terminate MASTER16* with a 316Ω ± 7% resistor to +5V.

AENx*
(E Pin 48)

Address Enable is an active-low, slot-specific signal generated by a totem-pole driver on Slot X controlled by the Permanent Master as an I/O address qualifier. AENx* is driven low for I/O address ranges x100h - x3FFh, x500h - x7FFh, x900h - xBFFh, and xD00h - xFFFh, where "x" is any value. It is driven active to slots 1 through 14 for the above I/O ranges. AENx* is also driven active on a slot-by-slot basis for I/O addresses z000h - z0FFh, z400h - z4FFh, z800h - z8FFh, and zC00h - zCFFh, where "z" corresponds to the specific slot. This allows the Permanent Master to perform slot-specific configuration. Extended Architecture I/O peripherals should decode AENx* as part of its I/O address, but should also provide an option to ignore AENx* for simple Permanent Masters that do not have Slot X capability (and therefore cannot drive AENx*).

BE0*-BE3*
(E Pins 49-52)

The four Byte Enable signals are active-low signals generated with a three-state driver by Permanent or Temporary Masters. The BE0*-BE3* signals identify the data bus lane usage. The Permanent Master must release BE0*-BE3* to a Temporary Master in response to a master request (MREQx*) and not release BE0*-BE3* in response to a bus request (BUSRQ*). The Temporary Master must drive BE0*-BE3* when in control of bus resources. These signals are synchronous to CLOCK*. The Permanent Master must terminate the BE0* - BE3* signals with a 1.8kΩ ± 10% resistor to +5V.

MEM16*
(E Pin 53)

Memory 16-bit is an active-low signal generated by memory boards with an open-collector driver. MEM16* is a dynamic bus size determination signal that when active, signals to the Permanent or Temporary Master that the memory board is capable of 16-bit memory data transfers. The slave memory board decodes enough of address lines A14-23 to assert MEM16* for its memory size, which is a minimum of 16Kbyte. This is compatible with STD-80 Series boards because they do not drive this signal active and the inactive level defines an 8-bit data transfer. A Temporary Master must manage the MEM16* signal when in control of bus resources. The Permanent Master must terminate MEM16* with a 316Ω ± 7% resistor to +5V.

M-IO
(E Pin 55)

Memory or I/O is generated with a three-state driver by the Permanent or Temporary Masters to indicate the type of cycle currently in progress. M-IO is driven to a logical high state to indicate a memory cycle and driven to a logical low state to indicate an I/O cycle. The Permanent Master must release M-IO to a Temporary Master generating a master request (MREQx*) and not release M-IO in response to a bus request (BUSRQ*). The Temporary Master must drive M-IO when in control of bus resources. M-IO is synchronous to CLOCK*. The Permanent Master must terminate M-IO with a 1.8kΩ ±10% resistor to +5V.

W-R
(E Pin 56)

Write or Read is generated with a three-state driver by the Permanent or Temporary Master to indicate the direction of data transfer. W-R is driven to a logical high state to indicate a write cycle and driven to a logical low state to indicate a read cycle. The Permanent Master must release W-R to a Temporary Master generating a master request (MREQx*) and not release W-R in response to a bus request (BUSRQ*). The Temporary Master must drive W-R when in control of bus resources. W-R is synchronous to CLOCK*. The Permanent Master must terminate W-R with a 1.8kΩ ±10% resistor to +5V.

IO16*
(E Pin 59)

I/O 16-bit is an active-low signal generated by an open-collector driver on I/O boards. IO16* is a dynamic bus size determination signal. If IO16* is active, it signals the Permanent or Temporary Master that the I/O board is capable of 16-bit I/O data transfers. The slave I/O board decodes enough of address lines A0-15 to assert IO16* for its I/O size and must not include IORQ* in the decode (in contrast to STD-80 requirements). This is compatible with STD-80 Series boards because they do not drive this signal active and the inactive level defines an 8-bit data transfer. A Temporary Master must manage IO16* when in control of bus resources. The Permanent Master must terminate IO16* with a 316Ω ± 7% resistor to +5V.

EX8*
(E Pin 60)

Exchange 8-bit is an active-low signal generated with an open-collector driver on memory and I/O Slave boards. EX8* is a dynamic bus size determination signal. If EX8* is active, it signals the Permanent or Temporary Master that the slave board is capable of 8-bit EA data transfers. The Permanent Master must terminate EX8* with a 316Ω ± 7% resistor to +5V. EX8* is synchronous to CLOCK*.

EX16*
(E Pin 63)

Exchange 16-bit is an active-low signal generated with an open-collector driver on memory and I/O Slave boards. EX16* is a dynamic bus size determination signal. If EX16* is active, it signals the Permanent or Temporary Master that the slave board is capable of 16-bit EA data transfers. The Permanent Master must terminate EX16* with a 316Ω ± 7% resistor to +5V. EX16* is synchronous to CLOCK*.

EX32*
(E Pin 64)

Exchange 32-bit is an active-low signal generated by an open-collector driver on memory and I/O Slave boards. EX32* is a dynamic bus size determination signal. If EX32* is active, it signals the Permanent or Temporary Master that the memory board is capable of 32-bit EA data transfers. The Permanent Master must terminate EX32* with a 316Ω ± 7% resistor to +5V. EX32* is synchronous to CLOCK*.

SLBURST*
(E Pin 71)

Slave Burst is an active-low signal generated with an open-collector driver by any board other than a Permanent Master. SLBURST* indicates that the board supports burst cycles. The Permanent Master must terminate SLBURST* with a 316Ω ± 7% resistor to +5V.

MSBURST*
(E Pin 72)

Master Burst is an active-low signal generated with a three-state driver by Permanent or Temporary Masters. MSBURST* indicates that the master can provide DMA burst cycles. The Permanent Master must release MSBURST* to a Temporary Master in response to a master request (MREQx*) and not release MSBURST* in response to a bus request (BUSRQ*). The Temporary Master must manage MSBURST* when in control of bus resources. MSBURST is synchronous to CLOCK*. The I/O Slave must terminate MSBURST* with a 1.8kΩ ± 10% resistor to +5V.

Peripheral Timing Control

CMD*
(E Pin 61)

Command is an active-low signal generated with a totem-pole driver by the Permanent Master to provide cycle timing information. CMD* is asserted synchronous to the backplane clock simultaneously with the negation of START*. CMD* is removed synchronously with CLOCK*. CMD* is always driven by the Permanent Master.

START*
(E Pin 62)

Start is an active-low signal generated with a three-state driver by permanent or Temporary Masters. START* is asserted after address and M-IO are valid. Please note that BE0*-BE3* and W-R are not guaranteed valid when START* is asserted. The Permanent Master must release START* to a Temporary Master in response to a master request (MREQx*) and not release START* in response to a bus request (BUSRQ*). The Temporary Master must drive START* when in control of bus resources. START* is synchronous to CLOCK*. The Permanent Master must terminate START* with a 1.8kΩ ± 10% resistor to +5V.

Interrupt and Bus Control

NOWS*
(E Pin 11)

No Wait States is an active-low signal generated with an open-collector driver by any memory or I/O slave board. NOWS* indicates that the remaining clock cycles of the data transfer are not required. The Permanent Master must terminate NOWS* with a 316Ω ± 7% resistor to +5V.

MAKx*
(E Pin 69)

Master Acknowledge is an active-low slot-specific signal generated by a totem-pole driver on a board plugged into Slot X. An EA Master generates a master request (MREQx*) to Slot X to gain control of the STD 32 bus resources. Slot X responds to a MREQx* by arbitrating between all requests and awarding the bus resources to the requestor with the highest priority. MAK* is synchronized to the rising edge of CLOCK* by the arbiter.

MREQx*
(E Pin 70)

Master Request is an active-low signal generated with a totem-pole driver by an EA master to gain control of the STD 32 bus resources. It is the responsibility of the requesting master to synchronize MREQx* to the falling edge of CLOCK*. SA masters do not use this signal for acquiring the bus.

A Temporary Master in control of bus resources is responsible for the signals shown in Table 3-4 "Signals That Temporary Masters Must Manage." The Slot X bus arbiter must terminate MREQx* with a 10kΩ ± 10% resistor to +5V.

IRQx
(E Pin 47)

Interrupt request is a slot-specific signal generated with a totem-pole driver by any board other than the Permanent Master. IRQx may be either positive edge-triggered or low-level asserted, depending on the Permanent Master interrupt controller. It is the responsibility of Slot X to synchronize IRQx. Any board other than the Permanent Master generates an active IRQx to invoke an interrupt procedure typically designed to service an asynchronous event. A common practice is to connect IRQx to a maskable interrupt request of an interrupt controller located on Slot X controlled by the Permanent Master. The Permanent Master must terminate IRQx with a 10kΩ ± 10% resistor to +5V.

EXRDY
(E Pin 65)

Expansion Ready is an active-high signal generated by memory or I/O boards with an open-collector driver to the Permanent or Temporary Masters. It is the responsibility of the memory and I/O to synchronize EXRDY to CLOCK*. Memory or I/O boards can negate EXRDY to extend the amount of time data is valid during a write cycle or to extend the amount of time allowed to return the data during a read cycle. With EXRDY inactive, the current master is not able to service interrupt requests, DMA requests, asynchronous data transfers, or dynamic RAM refresh. For these reasons, it is best not to keep EXRDY low for more than 10µs. The Permanent Master must terminate EXRDY with a 316Ω ± 7% resistor to +5V. EXRDY must be synchronous to CLOCK*.

LOCK*
(E Pin 2)

Lock is an active-low signal generated by an open-collector driver on Permanent and Temporary Masters. LOCK* is asserted to guarantee exclusive backplane access to a resource shared between multiple masters. The exclusive access continues as long as LOCK* is asserted. This protocol is useful for read/modify/write operations needed for semaphore management when multiple masters share common resources. LOCK* is also an input to the Permanent and Temporary Masters. The Permanent Master must terminate LOCK* with a 1.8kΩ ± 10% resistor to +5V.

Table 3-4. Signals That Temporary Masters Must Manage.

Required STD-80 Compatible (SA) Signal Mnemonics		**Required STD 32 Compatible (EA) Signal Mnemonics**	
D0-D7	A0-A23		
WR*	RD*	D0-D7, D8-D15	
IORQ*	MEMRQ*	A0-A15, XA16-23	
WAITRQ*	ALE*	MREQx*	BE0*-BE3*
STATUS 1*	STATUS 0*	IO16*	MEM 16*
BUSRQ*	BUSAK*	M-IO	LOCK*
IOEXP	INTAK*	DMAIOW*	DMAIOR*
		EX8*	EX16*
		EX32*	EXRDY
D8-D15	BHE*	LOCK*	NOWS*
IO16*	MEM 16*	START*	W-R
Optional STD-80 Compatible (SA) Signal Mnemonics		**Optional STD 32 Compatible (EA) Signal Mnemonics**	
DREQx*	DAKx*	D16-D31	XA24*-XA31*
INTRQ*	INTRQ 1*	IRQx	DREQx*
CNTRL* (INTRQ 2*)		SLBURST*	MSBURST*
		MASTER 16*	T-C

DMA Control

DMAIOW*
(E Pin 57)
DMA I/O Write Cycle is an active-low signal generated with a three-state driver by Permanent or Temporary Masters. An active DMAIOW* indicates a DMA read cycle, which is a data transfer from memory to I/O. The I/O Slave must qualify DMAIOW* with DAKx* before participating in the transfer.

DMAIOR*
(E Pin 58)
I/O Read Cycle is an active-low signal generated with a three-state driver by Permanent or Temporary Masters. An active DMAIOR* indicates a DMA write cycle which is a data transfer from I/O to memory. The I/O Slave must qualify DMAIOR* with DAKx* before participating in the transfer.

T-C
(E Pin 66)
Terminal Count is an active-high bidirectional signal generated by any board in the system. In one direction, T-C is driven active by an I/O Slave to terminate a DMA block mode transfer. In the other direction, T-C is driven by the Master to signal that the transfer count has rolled over. An I/O Slave decodes DAKx* to qualify an active T-C.

DAKx*
(E Pin 15)
DMA Acknowledge is an active-low slot-specific signal generated by a totem-pole driver on Slot X controlled by the Permanent Master. A Temporary Master or I/O Slave generates a DMA request (DREQx*) to Slot X to initiate a DMA transfer. Slot X responds to DREQx* by generating DAKx* and performing the DMA operation. See DREQx* description.

DREQx*
(E Pin 16)
DMA Request is an active-low slot-specific signal generated with a totem-pole driver on any board other than the Permanent Master. It is the responsibility of Slot X controlled by the Permanent Master to synchronize DREQx*. DREQx* is driven by an I/O Slave to request DMA transfers. The I/O Slave must hold DREQx* active until the DMA acknowledge (DAKx*) is received. SA Bus Masters may also use DREQx* to gain access to system resources in a multiple master architecture. In this mode, SA bus masters generate DREQx* synchronous to the system clock. The Slot X Arbiter generates DAKx* synchronous to the system clock after arbitration. The Permanent Master must terminate DREQx* with a 10kΩ ± 1% resistor to +5V.

3.3. IMPLEMENTATION

3.3.1 PERMANENT AND TEMPORARY MASTERS

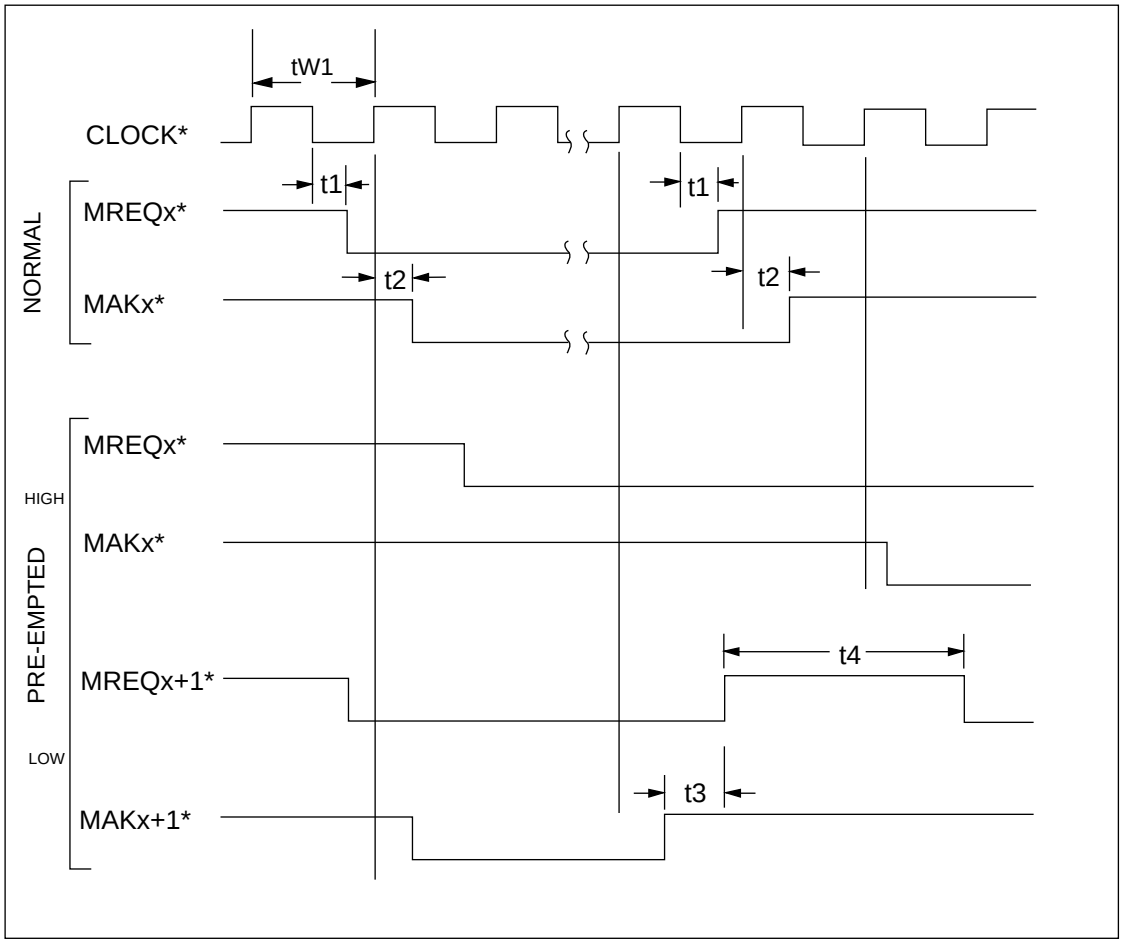
Bus Termination

The Permanent Master must resistively terminate all signals that could be managed by a Temporary Master in response to a BUSRQ*, DREQx*, or MREQ*, as listed in Table 3-4 "Signals That Temporary Masters Must Manage." Resistive termination should be a 1.8kΩ ± 10% pullup to +5V.

3.4. TIMING

Figure 3-2 "Master Request / Acknowledge Timing" illustrates the critical timing parameters from the point of view of the Permanent Master.

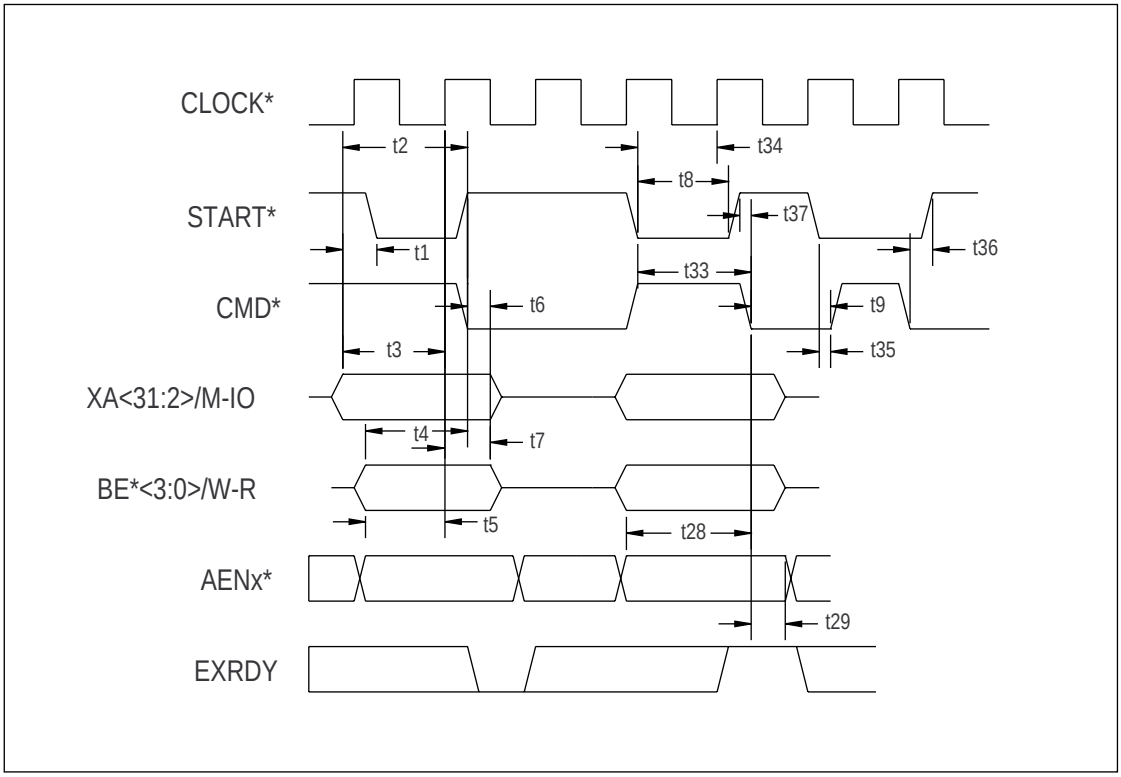
Figures 3-3 through 3-5 illustrate Extended Architecture STATUS, EXRDY / EX32* / EX16* / EX8*, and DATA ACCESS timing, respectively.



	MIN	MAX	DESCRIPTION
tW1	124	126	CLOCK* Period (8 MHz)
t1	2	33	CLOCK* ↓ to MREQx* high or low
t2	2	40	CLOCK* ↑ to MAKx* high or low
t3		64tW1	MAKx* ↑ to MREQx* high
t4	2tW1		MREQx* ↑ to MREQx* low

All times given in nanoseconds.

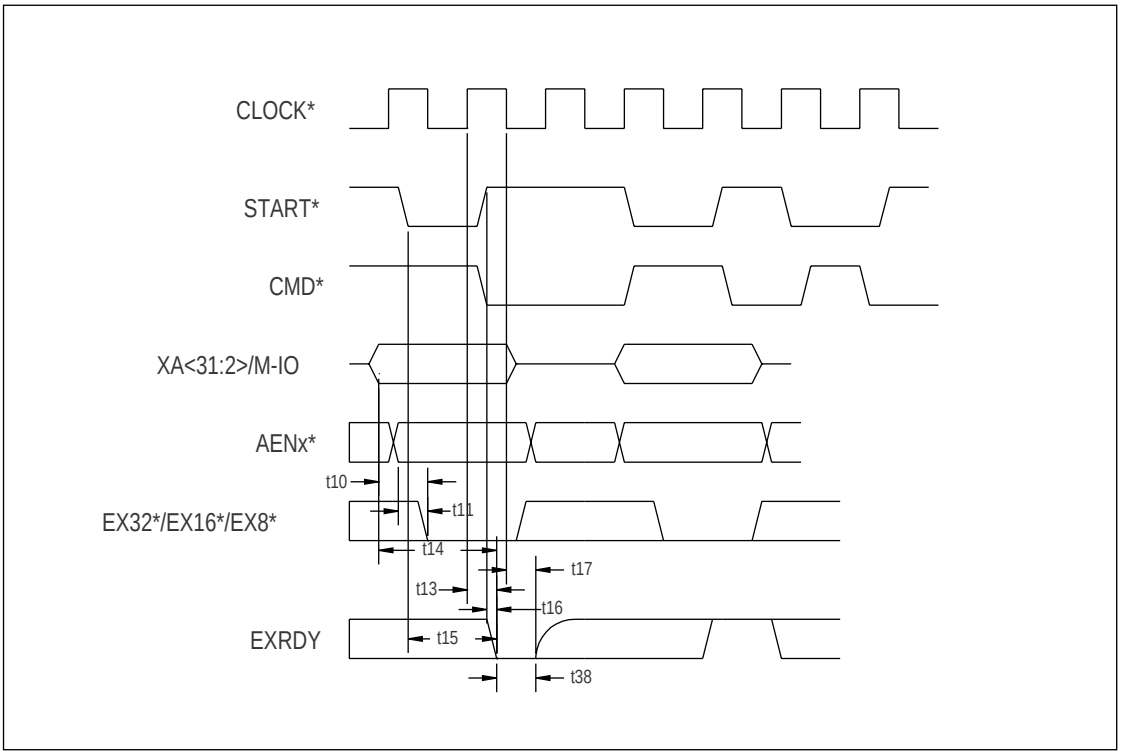
Figure 3-2. Master Request/Acknowledge Timing.



	MIN	MAX	DESCRIPTION
t1	10		Address, M-IO setup to START* asserted
t2	120		Address, M-IO setup to START* negated / CMD* asserted
t3	118		Address, M-IO setup to CLOCK* rising
t4	80		Byte Enables, W-R setup to START* negated / CMD* asserted
t5	80		Byte Enables, W-R setup to CLOCK* rising
t6	15		Address, M-IO, Byte Enables, W-R held from START* negated / CMD* asserted
t7	20		Address, M-IO, Byte Enables, W-R held BCLK rising
t8	115		START* pulse width
t9	115	2900	CMD* pulse width
t28	95		AENx* setup to START* negated / CMD* asserted
t29	25		AENx* hold time from START* negated / CMD* asserted
t33	90		START* asserted to CMD* asserted setup
t34	88		START* asserted to CLOCK* rising
t35		30	START* asserted to CMD* negated overlap
t36		25	CMD* asserted to START* negated underlap
t37		25	START* negated to CMD* asserted gap

All times given in nanoseconds.

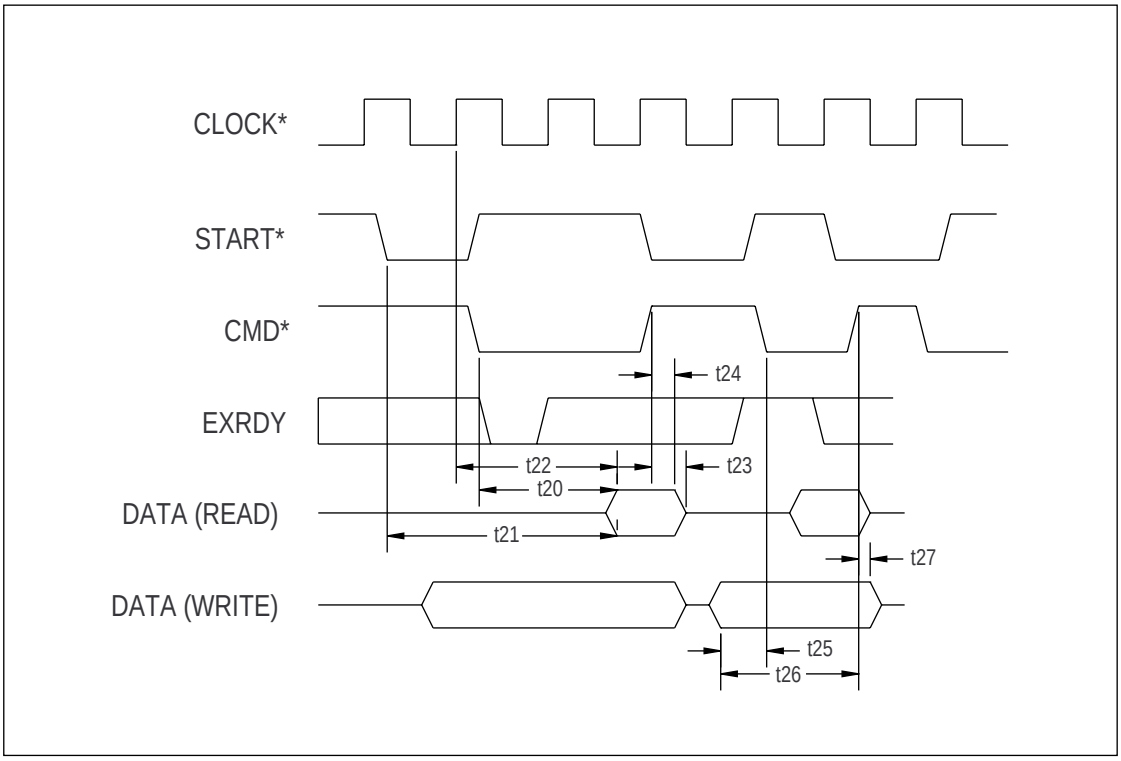
Figure 3-3. Extended Architecture Status Timing.



	MIN	MAX	DESCRIPTION
t10	2	54	EX32*, EX16*, EX8* delay from Address, M-IO
t11	2	34	EX32*, EX16*, EX8* delay AENx* asserted (I/O cycles using AENx*)
t13		35	EXRDY negated delay from BCLK rising
t14		143	EXRDY negated delay from Address, M-IO, (AENx* if decoded)
t15	2	124	EXRDY negated delay from START* asserted
t16		4	EXRDY negated delay from START* negated / CMD* asserted
t17	2	30	EXRDY float delay from BCLK falling
t38		2500	EXRDY negated pulse width

All times given in nanoseconds.

Figure 3-4. Extended Architecture EXRDY, EX32*, EX16*, EX8* Timing.



	MIN	MAX	DESCRIPTION
t20		47	Data delay from CMD* asserted (Read cycle)
t20(8-bit)		167	Data delay from CMD* asserted (Read cycle 8-bit)
t21		167	Data delay from START* asserted (Read cycle)
t21(8-bit)		287	Data delay from START* asserted (Read cycle 8-bit)
t22		77	Data delay from BCLK* rising when CMD* asserted (Read cycle)
t22(8-bit)		197	Data delay from BCLK* rising when CMD* asserted (Read cycle 8-bit)
t23		28	Data float delay from CMD* negated (Read cycle)
t24	16.5		Data hold time from CMD* negated (Read cycle 8-bit)
t25	-12		Data setup to CMD* asserted (Write cycle)
t25(8-bit)	-46		Data setup to CMD* asserted (Write cycle 8-bit)
t26	108		Data setup to CMD* negated (Write cycle)
t26(8-bit)	228		Data setup to CMD* negated (Write cycle 8-bit)
t27	25		Data hold time from CMD* negated (Write cycle)

All times given in nanoseconds.

Figure 3-5. Extended Architecture Data Access Timing.

EXTENDED ARCHITECTURE BUS CYCLES

CHAPTER 3

C H A P T E R 4

STD 32 COMPLIANCE LEVELS

4.1. INTRODUCTION

Any bus based system, including STD 32, needs a method for specifying compliance to the specification in order to provide easy system integration for its users. STD 32 extends the STD bus with additional transfer classes that dramatically improve the performance of STD systems. This chapter provides a structure that must be used by manufacturers who advertise a board as STD 32 compatible. This structure is intended to provide STD 32 users with a method for easily identifying the capabilities of a particular board. A product descriptor will be used to specify the capabilities of the board.

4.1.1 MASTER IMPLEMENTATIONS

Masters have the option, but are not required, to support all transfer classes. If a manufacturer chooses to support only certain classes of STD 32, this chapter gives him the mechanism to specify only those classes supported. It is suggested that manufacturers support all classes of transfers below and equal to the maximum class of transfers that the board supports. It is permissible for masters to support limited subsets of a lower class. Of particular importance is support for STD-80 class I/O (Class SA8). It is recommended that Class EA masters support SA8 I/O, if not full SA8 transfers. Masters that support full SA8 I/O will benefit from the support of the majority of STD bus I/O boards.

4.1.2. SLAVE IMPLEMENTATIONS

Slave board design can be approached in two ways. The first approach, generally most useful for simple slaves, is to design the board as a Class SA8 board and limit the transfer modes to the slower STD-80 variety. The second, more flexible design approach incorporates both SA and EA transfer classes. Appendix A demonstrates a method for designing I/O boards that support SA and EA transfers. As STD 32 matures, systems will rely primarily on the higher performance EA transfers, and new slaves will be designed around these capabilities.

4.1.3. REQUIREMENTS

All manufacturers who produce STD 32 compatible boards and advertise a board as STD 32 compatible will specify the capabilities of that board by the methods described in this chapter. This will allow the end users of STD 32 systems to purchase boards from various manufacturers with confidence that the boards will work together. It is the responsibility of the manufacturer who advertises the board as STD 32 compatible to verify that the board meets the letter and the spirit of the STD 32 specification. The major goal of this specification is to meet the concerns of end users and manufacturers regarding the capabilities of STD 32 while maintaining STD 32 as a reliable structure around which systems can be designed.

Boards that are compatible with a class may use only those signals associated with that class during transfers of that class level. Signals not used by that class are reserved for other transfer types and may not be used for other purposes. For example, SA8 masters do not drive EA control signals and are not allowed to use these pins for other signals.

4.2. PRODUCT DESCRIPTORS

Boards that are STD 32 compatible are required to be specified as such with a product descriptor. The descriptor indicates the STD 32 board modes that are STD 32 compatible, the class of transfers that are supported within each mode, and any options that the board supports. Each class has certain implicit requirements that are listed in the description of the class. These are documented in table form at the end of this chapter.

Product descriptors have the following format:

BOARD MODE: CLASSES SUPPORTED-{REQUIREMENTS} OPTIONS

The items within braces are those features that must be supported for the board to function. Items after the braces are options that may be left unsupported at the cost of some performance or functionality. Multiple items in each category are separated by commas.

4.2.1. BOARD MODES

There are five board modes within STD 32. Any combination of modes may be supported by a board. The five STD 32 board modes are:

Permanent Master -	Master that acknowledges requests from BUSRQ* or MREQx*
Temporary Master -	Master that drives either BUSRQ* or MREQx*
I/O Slave -	Slave that decodes and responds to I/O cycles
Memory Slave -	Slave that decodes and responds to memory cycles
Arbiter (Mn, Dn) -	Slot X arbiter that supports Mn MREQx* requests and/or Dn DREQx* requests for bus ownership

Board modes define the top level function of the board. For example, a typical processor board is a Permanent Master. A board can have more than one mode and each mode is defined in the product descriptor.

4.2.2. CLASSES

The types of transfers that a board supports are specified within STD 32 by a transfer class. Each class represents an increase in performance. The five classes are:

Class				**Special Notes**
SA8 -	Standard Architecture	(SA)	8-bit	STD-80 compatible boards (Chapter 2)
SA16 -	Standard Architecture	(SA)	16-bit	(backward compatible with SA8)
EA8 -	Extended Architecture	(EA)	8-bit	(slaves only)
EA16 -	Extended Architecture	(EA)	16-bit	(supports EA8 transfers)
EA32 -	Extended Architecture	(EA)	32-bit	(supports EA16 transfers)

Section 4.3 contains additional descriptions and examples.

4.2.3. REQUIREMENTS AND OPTIONS

Each class has certain requirements and may have several options. If a board absolutely needs to support a particular class option to function then that option is placed within the requirement braces in the product descriptor. Other options follow, allowing the manufacturer to inform the user of the complete capabilities of the board.

4.2.4. PRODUCT DESCRIPTOR SUGGESTED STYLE (DATA SHEETS)

Data sheets and advertisements for STD 32 compatible boards should include a product descriptor with a consistent format. This format states that the board complies with STD 32 and all board modes, transfer classes, and required or optional features in the following format:

STD 32 Compliance

Permanent Master:	Classes supported - {Requirements} Options
Temporary Master:	Classes supported - {Requirements} Options
Memory Slave:	Classes supported - {Requirements} Options
I/O Slave:	Classes supported - {Requirements} Options
Arbiter (Mn, Dn):	Classes supported - {Requirements} Options

4.3. STD 32 BOARD CLASSES

This section defines five different board classes and also specifies the requirements and optional features to be found in each class.

Each class represents a performance level. Notice that classes SA8 and SA16 revolve around the original Standard Architecture (SA), and that EA8, EA16, and EA32 transfers are Extended Architecture (EA) classes.

4.3.1. CLASS SA8 BOARDS

Class SA8 boards provide a mechanism for manufacturers of I/O boards that meet the STD-80 bus specification to become integrated into the STD 32 family. In order for an I/O board to meet Class SA8 requirements, the board must meet STD-80 specifications for timings and mechanical dimensions. Class SA8 masters must meet the requirements of Chapter 2 of this specification, with the exception of the mechanical requirements.

SA8 Requirements

Class SA8 boards have several implicit features that are not specified in the product descriptor. It is agreed that SA8 masters and SA8 slaves all support this core set of features. It is the responsibility of the manufacturer to ensure complete compatibility for this set of features if SA8 support is claimed.

The following features are implicit requirements of Class SA8 boards:

Permanent Masters	
D8	8-bit data (D0-D7) Standard Architecture (SA) transfer
A24	24-bit memory address generation
A16	16-bit I/O address generation and must drive IOEXP low during accesses to I/O addresses FC00h-FFFFh
Temporary Masters	
D8	8-bit data (D0-D7) Standard Architecture (SA) transfer
A24	24-bit memory address generation
A16	16-bit I/O address generation and must drive IOEXP low during accesses to I/O addresses FC00h-FFFFh
I/O Slaves	
D8	8-bit data (D0-D7) Standard Architecture (SA) transfer
A16	16-bit I/O address decoding[1]
Memory Slaves	
D8	8-bit data (D0-D7) Standard Architecture (SA) transfer
A24	24-bit memory address generation

[1] I/O Slaves that do not meet this specification may be qualified as STD 32 Class SA8 I/O if they meet the following criteria:

- 8-bit I/O address decoders must decode IOEXP low in order to locate these boards at FC00h-FFFFh in the STD 32 I/O map
- 10-bit I/O address decoders must decode (redundantly) only within the range X100h-X3FFh, meaning that these boards will show up at 0100h-03FFh, 1100h-13FFh, etc.

CHAPTER 4

SA8 Options

Boards manufactured for Class SA8 systems will have varying options that can also be defined in the product descriptor.

I	STD-80 interrupt generation by slaves, servicing by masters of any of the interrupts NMIRQ*, INTRQ*, INTRQ1*, CNTRL* (INTRQ2*), or INTRQ3*
ICA	Cascadable interrupt address support, masters / slaves
IXP	Slot-specific interrupt support, Permanent Masters / Slaves, positive edge-triggered Slaves drive and Slot X capable Permanent Masters decode this signal
IXL	Slot-specific interrupt support, Permanent Masters / Slaves, low-level asserted Slaves drive and Slot X capable Permanent Masters decode this signal
SDMA8	8-bit Standard Architecture DMA as defined in Chapter 2 of this specification
SDMA16	16-bit Standard Architecture DMA as defined in Chapter 2 of this specification
SDMABP	Standard Architecture DMA using BUSRQ* / BUSAK* for request and acknowledge and the control signals DMAIOR*, DMAIOW*, and T-C
MB	Permanent Master with BUSRQ* / BUSAK* support Temporary Master that requests bus via BUSRQ* and receives BUSAK*

SA8 Recommendations

It is suggested that future revisions of Class SA8 boards that were designed to the STD-80 specification incorporate the edge finger contacts defined in this specification. This edge finger contact layout is compatible with both STD-80 backplanes and STD 32 backplanes, and will help to reduce contact wear on unused pins when these boards are installed in STD 32 backplanes.

It is also suggested that STD-80 boards, when revised, meet the tighter mechanical specifications of STD 32, which guarantee more uniformity among manufacturers.

SA8 Product Descriptors

Class SA8 boards have the following product descriptor possibilities:

Permanent Master:	SA8-[I], [ICA], [IXP], [IXL], [SDMA8], [SDMABP], [MB]
Temporary Master:	SA8-[I], [ICA], [SDMA8], [MB], [MD], [IXP], [IXL]
I/O Slave:	SA8-[I], [ICA], [IXP], [IXL], [SDMA8], [SDMABP]
Memory Slave:	SA8-[I], [ICA], [IXP], [IXL]

Designators in brackets are optional.

SA8 Product Descriptor Examples

A Permanent Master that supports interrupts and frontplane 8-bit DMA would have the following product descriptor:

Permanent Master:	SA8-I, SDMA8

If this board also supported access to on-board memory during the control of the bus by another master, then this board would also have a class definition for the type of memory that the Temporary Master would see:

Permanent Master:	SA8-I, SDMA8, MB
Memory Slave:	SA8

An I/O Slave that needs 8-bit DMA support and interrupt support in order to work properly would have the following product descriptor:

I/O Slave:	SA8-{I, SDMA8}

4.3.2. CLASS SA16 BOARDS

Class SA16 boards meet the specifications in Chapter 2 of this document. The primary difference between Class SA8 and Class SA16 is that the additional data signals D8-D15 are transferred over E pins E44-E30, and the signals MEM16*, IO16* and BHE* are used for dynamic bus sizing.

CHAPTER 4

SA16 Requirements

Class SA16 boards have several implicit features that are not specified in the product descriptor. Class SA16 masters and SA16 slaves must support this core set of features. It is the responsibility of the manufacturer to ensure complete compatibility for this set of features if SA16 support is claimed. All Class SA16 boards must use the STD 32 edge finger specifications defined in Chapter 6.

The following features are implicit requirements of Class SA16 boards:

Permanent Masters	
D8	8-bit data (D0-D7) transfer as defined in Chapter 2
D16	16-bit data (D0-D15) transfer as defined in Chapter 2
A24	24-bit memory address generation
A16	16-bit I/O address generation and must drive IOEXP low during accesses to I/O addresses FC00h-FFFFh
Temporary Masters	
D8	8-bit data (D0-D7) transfer as defined in Chapter 2
D16	16-bit data (D0-D15) transfer as defined in Chapter 2
A24	24-bit memory address generation
A16	16-bit I/O address generation and must drive IOEXP low during accesses to I/O addresses FC00h-FFFFh
I/O Slaves	
D16	16-bit data (D0-D15) transfer as defined in Chapter 2
A16	16-bit I/O address decoding
Memory Slaves	
D16	16-bit data (D0-D15) transfer as defined in Chapter 2
A24	24-bit memory address decoding

SA16 Options

Boards manufactured for Class SA16 systems can have the following options. These options can also be defined in the product descriptor.

I	STD-80 interrupt generation by slaves, servicing by masters of NMIRQ*,INTRQ*, INTRQ1* and optionally CNTRL* (INTRQ2*)
ICA	Cascadable interrupt address support, masters / slaves
IXP	Slot-specific interrupt support, Permanent Masters / slaves, positive edge-triggered Slaves drive and Slot X capable Permanent Masters receive this signal
IXL	Slot-specific interrupt support, Permanent Masters / slaves, low-level asserted Slaves drive and Slot X capable Permanent Masters receive this signal
SDMA8	8-bit Standard Architecture DMA as defined in Chapter 2 of this specification
SDMA16	16-bit Standard Architecture DMA as defined in Chapter 2 of this specification
SDMABP	Standard Architecture DMA using BUSRQ* / BUSAK* for request and acknowledge and the control signals DMAIOR*, DMAIOW*, and T-C
MB	Permanent Master with BUSRQ* / BUSAK* support Temporary Master that requests bus via BUSRQ* and receives BUSAK*
MD	Permanent Master that requests the bus via DREQx* and receives DAKx*; requires Slot X capabilities Temporary Master that requests the bus via DREQx* and receives DAKx*

SA16 Product Descriptors

Class SA16 boards have the following product descriptor possibilities:

Permanent Master:	SA16, SA8-[I], [ICA], [IXP], [IXL], [SDMA8], [SDMA16], [SDMABP], [MB], [MD]
Temporary Master:	SA16, SA8-[I], [ICA], [SDMA8], [SDMA16], [MB], [MD], [IXP], [IXL]
I / O Slave:	SA16-[I], [ICA], [IXP], [IXL], [SDMA8], [SDMA16], [SDMABP]
Memory Slave:	SA16-[I], [ICA], [IXP], [IXL]

Designators within brackets are optional.

SA16 Product Descriptor Examples

A Permanent Master that supports interrupts and 16-bit frontplane DMA would have the following product descriptor:

Permanent Master: SA16, SA8-I, SDMA16

If this board also supports access to on-board memory during control of the bus by another master, then this board would also have a class definition for the type of memory that the Temporary Master would see:

Permanent Master: SA16, SA8-I, SDMA16, MB

Memory Slave: SA16, SA8

An I/O Slave that needs 16-bit DMA support and interrupt support would have the following product descriptor:

I/O Slave: SA16, SA8-{I, SDMA16}

A Temporary Master that uses BUSRQ* (MB) and has frontplane 16-bit DMA support (as defined in Chapter 2) would have the following product descriptor:

Temporary Master: SA16, SA8-{MB} SDMA16

4.3.3. CLASS EA8 BOARDS

Class EA8 boards have 8-bit Extended Architecture (EA) data transfer capabilities as specified in Chapter 3 of this specification. EA8 boards support 8-bit EA type transfers as part of EA16 and EA32 capability. Class EA8 masters are not allowed. 8-bit masters should be implemented as SA8 masters.

EA8 and SA8 boards differ in the mechanism for data transfer. EA transfers are nominally two bus clock transfers and use EA control signals (refer to Chapter 3).

Unlike Class SA8 and Class SA16, addresses A16-A23 are not multiplexed with D0-D7. Class EA8 address signals XA16-XA23 replace A16-A23 and are found on E pins E3-E10 in order to support higher performance pipelined transfers. All Class EA8 boards will use the STD 32 edge finger specifications as defined in Chapter 6.

EA8 Requirements

Class EA8 boards have several implicit features that are not specified in the product descriptor. It is agreed that EA8 slaves all support this core set of features. It is the responsibility of the manufacturer to ensure complete compatibility for this set of features if EA8 support is claimed. All Class EA8 boards will use the STD 32 edge finger specifications defined in Chapter 6. Class EA8 I/O Slaves do not use XA16-XA23 or XA24*-XA31* and may be designed without the two outermost portions of the E connector. EA8 I/O Slaves use AENx* for slot-specific addressing. Slot-specific addressing allows the permanent master to configure each peripheral board at power up. Normal I/O accesses do not use AENx* in the decode logic. EA8 boards must support 8-bit EA transfers as defined in Chapter 3.

The following features are implicit requirements of Class EA8 boards:

I/O Slaves	
XD8	8- bit EA data transfer capability
XA16	16-bit EA I/O address fully decoded with AENx*
Memory Slaves	
XD8	8-bit EA data transfer capability
XA32	32-bit EA memory address fully decoded

CHAPTER 4

EA8 Options

Boards manufactured for Class EA8 systems can have the following options. These options can also be defined in the product descriptor. These options are in addition to any options that may be supported for Class SA8 transfers.

I	STD-80 interrupt generation by slaves
ICA	Cascadable interrupt address support
IXP	Slot-specific interrupt support, Permanent Masters / slaves, positive edge-triggered Slaves drive and Slot X capable Permanent Masters receive this signal
IXL	Slot-specific interrupt support, Permanent Masters / slaves, low-level asserted Slaves drive and Slot X capable Permanent Masters receive this signal
GAX	Geographic Addressing (slot-specific) support
NOWS	No wait state support
EBURST	Extended Architecture burst transfers supported
EDMAA	Extended Architecture type A DMA transfers supported
EDMAB	Extended Architecture type B DMA transfers supported
EDMAC	Extended Architecture type C DMA transfers supported

EA8 Product Descriptors

Class EA8 boards have the following product descriptor possibilities:

I / O Slave:	EA8, [SA16], [SA8]-[I], [ICA], [IXP], [IXL], [NOWS], [GAX], [EBURST], [EDMAA], [EDMAB], [EDMAC]
Memory Slave:	EA8, [SA16], [SA8]-[I], [ICA], [IXP], [IXL], [NOWS], [GAX], [EBURST]

Designators within brackets are optional.

EA8 Product Descriptor Examples

A Class EA8 I/O Slave that supports both 16- and 8-bit STD-80 transfers, interrupts, and 8-bit frontplane DMA would have the following product descriptor:

I/O Slave: EA8,SA16,SA8-SDMA8

4.3.4. CLASS EA16 BOARDS

Class EA16 boards have 16-bit Extended Architecture (EA) data transfer capabilities as specified in Chapter 3 of this specification. EA16 boards support both 16-bit and 8-bit EA type transfers via dynamic bus sizing.

Class EA16 boards may also support Class SA8 and Class SA16 transfers and would be so designated in their product descriptor. EA16 and SA16 boards differ in the mechanism for data transfer. EA transfers are nominally two bus clock transfers and use EA control signals (refer to Chapter 3).

Unlike Class SA8 and Class SA16, addresses A16-A23 are not multiplexed with D0-D7. Class EA16 address signals XA16-XA23 replace A16-A23 and are found on E pins E3-E10 in order to support high performance pipelined transfers. All Class EA16 boards will use the STD 32 edge finger specifications as defined in Chapter 6.

Classes SA8 and SA16 require memory and I/O support. Due to the requirement of multiplexed data during SA8 and SA16 memory cycles, memory support for these classes adds complexity for which there is little payback. In order to allow Extended Architecture masters to pick up the large marketplace of simple STD-80 I/O boards and forego supporting STD-80 memory transfers, a master will specify this within the product descriptor. EA16 and EA32 masters can optionally specify SA8(I/O) as a class in lieu of SA8, and SA16(I/O) in lieu of SA16, if they choose to support SA I/O only.

EA master systems generally will have local memory or use STD 32 EA16 or EA32 memory because of its speed. An EA master that executes out of SA8 or SA16 memory will be considerably slower than if EA16 or EA32 transfers were being used for the same application. In all cases, the product descriptor allows the manufacturer to fully indicate the capabilities of the master.

EA Masters use MREQx* and MAKx* to gain control of the bus. BUSRQ* and DREQx* are for SA Master use only. However, an EA Permanent Master may service SA Temporary Masters by acknowledging BUSRQ* with BUSAK*.

CHAPTER 4

EA16 Requirements

Class EA16 boards have several implicit features that are not specified in the product descriptor. It is agreed that EA16 masters and EA16 slaves all support this core set of features. It is the responsibility of the manufacturer to ensure complete compatibility for this set of features if EA16 support is claimed. All Class EA16 boards will use the STD 32 edge finger specifications defined in Chapter 6. Class EA16 I/O Slaves do not support XA16-XA23 or XA24*-XA31* and may be designed without the two outermost portions of the E connector. EA16 I/O Slaves must include AENx* in their decode logic for slot-specific configuration only. EA16 boards must support both 8-bit and 16-bit EA transfers as defined in Chapter 3.

The following features are implicit requirements of Class EA16 boards:

Permanent Masters	
XD16	8- and 16-bit EA data transfer capability
XA16	16-bit EA I/O address generation for I/O transfers
XA32	32-bit EA memory address generation by masters (XA24*-XA31* may be pulled up inactive and not dynamically driven)
Temporary Masters	
XD16	8- and 16-bit EA data transfer capability
XA16	16-bit EA I/O address generation for I/O transfers
XA32	32-bit EA memory address generation by masters (XA24*-XA31* may be pulled up inactive and not dynamically driven)
I/O Slaves	
XD16	8- and 16-bit EA data transfer capability
XA16	16-bit EA I/O address fully decoded with AENx*
Memory Slaves	
XD16	8- and 16-bit EA data transfer capability
XA32	32-bit EA memory address fully decoded

EA16 Options

Boards manufactured for Class EA16 systems can have the following options. These options can also be defined in the product descriptor. These options are in addition to any options that may be supported for Class SA8 or SA16 transfers.

I	STD-80 interrupt generation by slaves, servicing by masters of NMIRQ*, INTRQ*, INTRQ1* and optionally CNTRL* (INTRQ2*)
ICA	Cascadable interrupt address support, masters / slaves
IXP	Slot-specific interrupt support, Permanent Masters / slaves, positive edge-triggered Slaves drive and Slot X capable Permanent Masters receive this signal
IXL	Slot-specific interrupt support, Permanent Masters / slaves, low-level asserted Slaves drive and Slot X capable Permanent Masters receive this signal
GAX	Geographic Addressing (slot-specific) support masters / slaves
MB	Permanent Master with BUSRQ* / BUSAK* support
MX	Permanent Master that requests the bus via MREQx* and receives MAKx* Temporary Master that requests the bus via MREQx* and receives MAKx*
NOWS	No wait state support, masters / slaves
EBURST	Extended Architecture burst transfers supported
EDMAA	Extended Architecture type A DMA transfers supported
EDMAB	Extended Architecture type B DMA transfers supported
EDMAC	Extended Architecture type C DMA transfers supported

CHAPTER 4

EA16 Product Descriptors

Class EA16 boards have the following product descriptor possibilities:

Permanent Master:	EA16, [SA16 or SA16(I/O)], [SA8 or SA8(I/O)]-[I], [ICA], [IXP], [IXL], [MB], [MX], [NOWS], [GAX], [EBURST], [EDMAA], [EDMAB], [EDMAC]
Temporary Master:	EA16, [SA16 or SA16(I/O)], [SA8 or SA8(I/O)]-[D8, A24], [I], [ICA], [MX], [NOWS], [GAX], [EBURST]
I/O Slave:	EA16, [SA16], [SA8]-[D8, A16], [I], [ICA], [IXP], [IXL], [NOWS], [GAX], [EBURST], [EDMAA], [EDMAB], [EDMAC]
Memory Slave:	EA16, [SA16], [SA8]-[D8, A24], [I], [ICA], [IXP], [IXL], [NOWS], [GAX], [EBURST]

Designators within brackets are optional.

EA16 Product Descriptor Examples

A Class EA16 Permanent Master that supports 16- and 8-bit STD-80 transfers, interrupts, MREQx*/MAKx*, and frontplane 8-bit DMA would have the following product descriptor:

Permanent Master: EA16, SA16, SA8-MX, SDMA8

A similar board that supports only STD-80 I/O [SA8(I/O)], along with the remaining features, would have the following product descriptor:

Permanent Master: EA16, SA8(I/O)-MX, SDMA8

4.3.5. CLASS EA32 BOARDS

Class EA32 boards have 32-, 16-, and 8-bit EA data transfer capabilities as specified within Chapter 3 of this specification. Class EA32 boards may also support Class SA8 and Class SA16 transfers and would be so designated in their product descriptor. If support is provided for Class SA8 or SA16 I/O only, then the product descriptor will indicate this by specifying SA8(I/O) and SA16(I/O), respectively.

Addresses A16-A23 are no longer multiplexed with D0-D7 as in Class SA8 and Class SA16. Class EA32 address signals XA16-XA23 replace A16-A23. They are located on E pins E3-E0 in order to support high performance pipelined transfers. All Class EA32 boards will use the STD 32 edge finger specifications as defined in Chapter 6.

SA8 and SA16 require memory and I/O support. Due to the multiplexing data schemes of SA8 and SA16 memory, memory support for these classes adds complexity for which there is little payback.

In order to allow Extended Architecture masters to pick up the large marketplace of simple STD-80 I/O boards and forego supporting STD-80 memory transfers, a master will specify this within the product descriptor. EA16 and EA32 masters can optionally specify SA8(I/O) as a class in lieu of SA8, and SA16(I/O) in lieu of SA16, if they choose to support SA I/O only.

EA master systems generally will have local memory or use STD 32 EA16 or EA32 memory because of its speed. An EA master that executes out of SA8 or SA16 memory will be considerably slower than if EA16 or EA32 transfers were being used for the same application. In all cases, the product descriptor allows the manufacturer to fully indicate the capabilities of the master.

EA Masters use MREQx* and MAKx* to gain control of the bus. BUSRQ* and DREQx* are for SA Master use only. However, an EA Permanent Master may service SA Temporary Masters by acknowledging BUSRQ* with BUSAK*.

EA32 Requirements

Class EA32 boards have several implicit features that are not specified in the product descriptor. Class EA32 masters and slaves must support this core set of features. It is the responsibility of the manufacturer to ensure complete compatibility for this set of features if EA32 support is claimed. All Class EA32 boards will use the STD 32 edge finger specifications defined in Chapter 6. Class EA32 I/O Slaves do not support XA16-XA23 or XA24*-XA31* and may be designed without the two outermost portions of the E connector. EA32 I/O Slaves must include AENx* in their decode logic for slot-specific configuration.

CHAPTER 4

The following features are implicit requirements of Class EA32 boards:

Permanent Masters	
XD32	32-bit EA data transfer capability
XD16	8- and 16-bit EA data transfer capability
XA16	16-bit EA I/O address generation for I/O transfers
XA32	32-bit EA memory address generation by masters (XA24*-XA31* may be pulled up inactive and not dynamically driven)
Temporary Masters	
XD32	32-bit EA data transfer capability
XD16	8- and 16-bit EA data transfer capability
XA16	16-bit EA I/O address generation for I/O transfers
XA32	32-bit EA memory address generation by masters (XA24*-XA31* may be pulled up inactive and not dynamically driven)
I/O Slaves	
XD32	32-bit EA data transfer capability
XD16	8- and 16-bit EA data transfer capability
XA16	16-bit EA I/O address fully decoded with AENx*
Memory Slaves	
XD32	32-bit EA data transfer capability
XD16	8- and 16-bit EA data transfer capability
XA32	32-bit EA memory address fully decoded

EA32 Options

Boards manufactured for Class EA32 systems can have the following options. These options can also be defined in the product descriptor. These options are in addition to any options that may be supported for Class SA8 or SA16 transfers.

I	STD-80 interrupt generation by slaves, servicing by masters of NMIRQ*, INTRQ*, INTRQ1* and optionally CNTRL* (INTRQ2*)
ICA	Cascadable interrupt address support, masters / slaves
IXP	Slot-specific interrupt support, Permanent Masters / slaves, positive edge-triggered
IXL	Slot-specific interrupt support, Permanent Masters / slaves, low-level asserted
GAX	Geographic Addressing (slot-specific) support masters / slaves
MB	Permanent Master with BUSRQ* / BUSAK* support
MX	Permanent Master that requests the bus via MREQx* and receives MAKx* Temporary Master that requests the bus via MREQx* and receives MAKx*
NOWS	No wait-state support, masters / slaves
EBURST	Extended Architecture burst transfers supported
EDMAA	Extended Architecture type A DMA transfers supported
EDMAB	Extended Architecture type B DMA transfers supported
EDMAC	Extended Architecture type C DMA transfers supported

CHAPTER 4

EA32 Product Descriptors

Class EA32 boards have the following product descriptor possibilities:

Permanent Master:	EA32, EA16, [SA16 or SA16(I/O)], [SA8 or SA8(I/O)]-[I], [ICA], [IXP], [IXL], [MB], [MX], [NOWS], [GAX], [EBURST], [EDMAA], [EDMAB], [EDMAC]
Temporary Master:	EA32, EA16, [SA16 or SA16(I/O)], [SA8 or SA8(I/O)]-[I], [ICA], [MX], [NOWS], [GAX], [EBURST]
I/O Slave:	EA32, EA16, [SA16], [SA8]-[I], [ICA], [IXP], [IXL], [NOWS], [GAX], [EBURST], [EDMAA], [EDMAB], [EDMAC]
Memory Slave:	EA32, EA16, [SA16], [SA8]-[I], [ICA], [IXP], [IXL], [NOWS], [GAX], [EBURST]

Designators within brackets are optional.

EA32 Product Descriptor Examples

A Class EA32 Permanent Master that supports both 16- and 8-bit STD-80 transfers, interrupts,MREQx*/MAKx*, and 8-bit frontplane DMA would have the following product descriptor:

Permanent Master: EA32, EA16, SA16,SA8-MX, SDMA8

A similar board that supports only STD 80 I/O [SA8(I/O)], along with the remaining features, would have the following product descriptor:

Permanent Master: EA32, EA16, SA8(I/O)-MX, SDMA8

4.4. CLASS SIGNAL REQUIREMENT

Certain signals must be managed in each class. This section presents signals that must be supported by each class if support for that class is claimed. Items in brackets for slaves are optional depending on the application. Section 5 details additional signal support for optional features.

4.4.1. CLASS SA8 SIGNAL REQUIREMENTS

Permanent Master
A0-A24, D0-D7, IORQ*, MEMRQ*, ALE*, RD*, WR*, IOEXP, STATUS0*, STATUS1*, WAITRQ*, CLOCK*, SYSRESET*, PBRESET*, DCPDN*
Temporary Master
A0-A24, D0-D7, IORQ*, MEMRQ*, ALE*, RD*, WR*, IOEXP, STATUS0*, STATUS1*, WAITRQ*, BUSRQ* or DREQx*, SYSRESET*, [PBRESET*]
I/O Slave
A0-15, D0-D7, IORQ*, RD* and / or WR*, [IOEXP], [STATUS0*], [STATUS1*], [SYSRESET*]
Memory Slave
A0-23, D0-D7, MEMRQ*, RD* and / or WR*, [STATUS0*], [STATUS1*], [SYSRESET*]
Arbiter (Mn, Dn)
MREQ0*-MREQ (Mn)*, MAK0*-MAK(Mn)* and / or DREQ0*-DREQ (Dn)*, DAK0*-DAK (Dn)*

4.4.2. CLASS SA16 SIGNAL REQUIREMENTS

Permanent Master
A0-A24, D0-D15, IORQ*, MEMRQ*, ALE*, RD*, WR*, IOEXP, STATUS0*, STATUS1*, WAITRQ*, CLOCK*, BHE*, IO16*, MEM16*, SYSRESET*, PBRESET*, DCPDN*
Temporary Master
A0-A24, D0-D15, IORQ*, MEMRQ*, ALE*, RD*, WR*, IOEXP, STATUS0*, STATUS1*, WAITRQ*, BUSRQ* or DREQx*, BHE*, IO16*, MEM16*, SYSRESET*, [PBRESET*]
I/O Slave
A0-15, D0-D15, IORQ*, [RD*], [WR*], [IOEXP], [STATUS*], [STATUS1*], BHE*, IO16*, MEM16*, SYSRESET*
Memory Slave
A0-23, D0-D15, MEMRQ*, [RD*], [WR*], [STATUS*], [STATUS1*], BHE*, IO16*, MEM16*, [SYSRESET*]
Arbiter (Mn, Dn)
MREQ0*-MREQ(Mn)*, MAK0*-MAK(Mn)* and / or DREQ0*-DREQ(Dn)*, DAK0*-DAK(Dn)*

4.4.3. CLASS EA8 SIGNAL REQUIREMENTS

I/O Slave
D0-D7, A2-A15, EX16*, BE0*-BE3*, CMD*, START*, W-R, M-IO, AENx*, [EXRDY]
Memory Slave
A2-A15, XA16-XA23, XA24*-XA31*, EX16*, BE0*-BE3*, CMD*, START*, W-R, M-IO, [EXRDY]

4.4.4. CLASS EA16 SIGNAL REQUIREMENTS

Permanent Master
D0-D15, A2-A15, XA16-XA23, XA24*-XA31*, EX16*, EX32*, BE0*-BE3*, LOCK*, CMD*, START*, W-R, M-IO, CLOCK*, SYSRESET*, PBRESET*, DCPDN*, EXRDY
Temporary Master
D0-D15, A2-A15, XA16-XA23, XA24*-XA31*, EX16*, EX32*, BE0*-BE3*, LOCK*, START*, W-R, M-IO, MREQx*, EXRDY
I/O Slave
D0-D15, A2-A15, EX16*, BE0*-BE3*, CMD*, START*, W-R, M-IO, AENx*, [EXRDY]
Memory Slave
D0-D15, A2-A15, XA16-XA23, XA24*-XA31*, EX16*, BE0*-BE3*, CMD*, START*, W-R, M-IO, [EXRDY]
Arbiter (Mn, Dn)
MREQ0*-MREQ(Mn)*, MAK0*-MAK(Mn)* and / or DREQ0*-DREQ(Dn)*, DAK0*-DAK(Dn)*

4.4.5. CLASS EA32 SIGNAL REQUIREMENTS

Permanent Master
D0-D32, A2-A15, XA16-XA23, XA24*-XA31*, EX16*, EX32*, BE0*-BE3*, LOCK*, CMD*, START*, W-R, M-IO, CLOCK*, SYSRESET*, PBRESET*, DCPDN*, EXRDY
Temporary Master
D0-D32, A2-A15, XA16-XA23, XA24*-XA31*, EX16*, EX32*, BE0*-BE3*, LOCK*, START*, W-R, M-IO, MREQx*, EXRDY
I/O Slave
D0-D32, A2-A15, EX16*, EX32*, BE0*-BE3*, CMD*, START*, W-R, M-IO, [EXRDY]
Memory Slave
D0-D15, A2-A15, XA16-XA23, XA24*-XA31*, EX16*, EX32*, BE0*-BE3*, CMD*, START*, W-R, M-IO, [EXRDY]
Arbiter (Mn, Dn)
MREQ0*-MREQ(Mn)*, MAK0*-MAK(Mn)* and / or DREQ0*-DREQ(Dn)*, DAK0*-DAK(Dn)*

4.5. PRODUCT DESCRIPTOR REFERENCE

4.5.1. FEATURE LISTING

Table 4-1 "Feature Listing" describes the various features that may be found in an STD 32 product descriptor.

SA = Standard Architecture
EA = Extended Architecture

Table 4-1. Feature Listing.

Feature	Description	Additional Signal Support
A16	SA 16-bit I/O address decode (I/O Slave)/ generation (master)	A0-A15
A24	SA 24-bit memory address decode (Memory Slave)/ generation (master)	A0-A24
D8	SA 8-bit data transfer, masters/slaves	D0-D7
D16	SA 16-bit transfer, masters/slaves	D0-D15, MEM16*, BHE*, IO16*
EBURST	EA Burst transfer capability, masters/slaves	SLBURST*, MSBURST*
EDMAA	EA type A DMA transfers supported, masters/slaves	DMAIOW*, DMAIOR*, DREQx*, DAKx*, T-C
EDMAB	EA type B DMA transfers supported, masters/slaves	DMAIOW*, DMAIOR*, DREQx*, DAKx*, T-C
EDMAC	EA type C DMA transfers supported, masters/slaves	DMAIOW*, DMAIOR*, DREQx*, DAKx*, T-C, SLBURST*, MSBURST*
GAX	EA geographical address support masters/slaves	AENx* (masters)
I	SA (STD-80) interrupt generation (slaves)/ servicing (masters) on INTRQ*, INTRQ1*, NMIRQ*, CNTRL* (INTRQ2*), or INTRQ3*	INTRQ*, INTRQ1*, NMIRQ* [CNTRL* (INTRQ2*)], INTRQ3*
ICA	Cascadable interrupt address support, masters/slaves	A8-A10 during INTA

Table 4-1. Feature Listing (Continued).

Feature	Description	Additional Signal Support
IXP	Slot-specific interrupt servicing (Permanent Masters) / generation (slaves), positive edge-triggered	IRQx
IXL	Slot-specific interrupt servicing (Permanent Masters) / generation (slaves), low-level asserted	IRQx
MB	Bus arbitration via BUSRQ* / BUSAK* - SA Masters only	BUSRQ*, BUSAK*
MD	Bus Arbitration via DREQx* / DAKx* - SA Masters only	DREQx*, DAKx*
MX	Bus Arbitration via MREQx* / MAKx* - Eax Masters only	MREQx*, MAKx*
NOWS	No wait-state (NOWS*) support	NOWS*
SDMA8	8-bit SA frontplane DMA as specified in Chapter 2, masters / slaves	
SDMA16	16-bit SA frontplane DMA as specified in Chapter 2, masters / slaves	
SDMABP	8- or 16-bit backplane DMA supported via BUSRQ* / BUSAK* and DMAIOR*, DMAIOW*, T-C	BUSRQ*, BUSAK*, DMAIOR*, DMAIOW*, T-C
XA16	EA 16-bit address decode, I / O transfers	A2-A15, BE0*-BE3* as per Chapter 3
XA32	Full 32-bit EA address space driven, masters may pull XA24*-XA31* passively high. Full 32-bit EA address decoded by Memory Slaves	A2-A15, XA16-XA23, XA24*-XA31*, BE0*-BE3*
XD8	8-bit EA data transfer, masters / slaves	D0-D7
XD16	16-bit EA data transfer, masters / slaves	D0-D15, EX16*, BE0*-BE3*
XD32	32-bit EA data transfer, masters / slaves	D0-D31, EX16*, EX32*, EX32* BE0*-BE3*

4.5.2. FEATURE SUPPORT SIGNAL REQUIREMENTS

Table 4-2 "Signal Requirements" indicates required features, optional features, features that are not allowed, and features that are not applicable for all of the masters, I/O Slaves and Memory Slaves in each class.

Table 4-2. Signal Requirements.

		Class Requirements									Options															
Board	Class	D8	D16	XD8	XD16	XD32	A16(I/O)	A24 (MEM)	XA16 (I/O)	XA32 (MEM)	I	ICA	IXP	IXL	SDMA8	SDMA16	SDMABP	MB	MD	MX	EBURST	EDMAA	EDMAB	EDMAC	NOWS	GAX	Notes
Permanent Masters	SA8	R	-	-	-	-	-	R	-	-	O	O	O	O	O	-	O	O	O	-	-	-	-	-	-	-	
	SA16	R	R	-	-	-	R	R	-	-	O	O	O	O	O	O	O	O	O	-	-	-	-	-	-	-	
	EA16	-	-	R	R	-	-	-	R	R	O	O	O	O	-	-	-	-	-	R	O	O	O	O	O	O	1
	EA32	-	-	R	R	R	-	-	R	R	O	O	O	O	-	-	-	-	-	R	O	O	O	O	O	O	1
Temporary Masters	SA8	R	-	-	-	-	R	R	-	-	O	O	-	-	O	-	O	O	-	-	-	-	-	-	-	-	
	SA16	R	R	-	-	-	R	R	-	-	O	O	-	-	O	O	O	O	O	-	-	-	-	-	-	-	
	EA16	-	-	R	R	-	-	-	R	R	O	O	-	-	-	-	-	-	-	R	O	-	-	-	O	O	1
	EA32	-	-	R	R	R	-	-	R	R	O	O	-	-	-	-	-	-	-	R	O	-	-	-	O	O	1
I/O Slaves	SA8	R	-	-	-	-	R	-	-	-	O	O	O	O	O	-	O	-	-	-	-	-	-	-	-	-	2
	SA16	O	R	-	-	-	R	-	-	-	O	O	O	O	O	O	O	-	-	-	-	-	-	-	-	-	
	EA8	-	-	R	-	-	-	-	R	-	O	O	O	O	-	-	-	-	-	-	O	O	O	O	O	O	
	EA16	-	-	R	R	-	-	-	R	-	O	O	O	O	-	-	-	-	-	-	O	O	O	O	O	O	
	EA32	-	-	R	R	R	-	-	R	-	O	O	O	O	-	-	-	-	-	-	O	O	O	O	O	O	
Memory Slaves	SA8	R	-	-	-	-	-	R	-	-	O	O	O	O	-	-	-	-	-	-	-	-	-	-	-	-	
	SA16	O	R	-	-	-	-	R	-	-	O	O	O	O	-	-	-	-	-	-	-	-	-	-	-	-	
	EA8	-	-	R	-	-	-	-	-	R	O	O	O	O	-	-	-	-	-	-	O	-	-	-	O	O	
	EA16	-	-	R	R	-	-	-	-	R	O	O	O	O	-	-	-	-	-	-	O	-	-	-	O	O	
	EA32	-	-	R	R	R	-	-	-	R	O	O	O	O	-	-	-	-	-	-	O	-	-	-	O	O	
Arbiter(n)	SA8	-	-	-	-	-	-	-	-	-	-	-	O	O	-	-	-	O	O	O	-	-	-	-	-	-	
	SA16	-	-	-	-	-	-	-	-	-	-	-	O	O	-	-	-	O	O	O	-	-	-	-	-	-	
	EA16	-	-	-	-	-	-	-	-	-	-	-	O	O	-	-	-	O	O	O	-	-	-	-	-	-	
	EA32	-	-	-	-	-	-	-	-	-	-	-	O	O	-	-	-	O	O	O	-	-	-	-	-	-	

R: Required O: Optional -: Not Applicable

Notes for Table 16:

1 EA16 and EA32 masters may optionally choose support for SA8 and SA16 I/O only and claim this by specifying SA8(I/O) and SA16(I/O), respectively, in their product descriptors.

2 8-bit address decoding STD 80 I/O boards may claim Class A compatibility if IOEXP is decoded low as part of the board select.

- 10-bit address decoding STD I/O boards may claim Class A compatibility if addressed in range X100h-X3FFh.
- Boards falling into this category must state clearly this limitation by specifying the number of address bits decoded.

STD COMPLIANCE LEVELS

CHAPTER 4

C H A P T E R 5

ELECTRICAL SPECIFICATIONS

ELECTRICAL SPECIFICATIONS

CHAPTER 5

5.1. INTRODUCTION

The STD 32 bus is composed of 136 signals as follows:

- Five are +5V
- Nine are ground
- One is battery voltage
- One is +12V
- One is -12V
- Two are AUX GND (auxiliary ground)

The remaining 117 signals comprise the control, interrupt, arbitration, address and data signals described in Chapters 2 and 3.

These signals are TTL level signals driven by either the Permanent Master, Temporary Master, or I/O slaves. The active level of the signal is given in the signal name. Active low signals are followed by an asterisk (*).

5.2. POWER DISTRIBUTION

Power is supplied to STD 32 bus cards via the backplane. All power is available as DC regulated voltage. Switching power supplies are recommended.

5.2.1. D.C. VOLTAGE SPECIFICATIONS

Table 5-1. Voltage Specification.

Source	Description	Nominal Value (Volts)	Allowed Variance from nominal	Max Ripple
+5	+5VDC	5.0	±5%	50mV
+12	+12VDC	12.0	±5%	50mV
-12	-12VDC	-12.0	±5%	50mV
GND	Ground	0.0	-	-
AUX GND	Auxiliary Ground	0.0	-	-
VBAT	Battery Voltage	2.0 - 5.0	-	-

It is suggested that peripheral cards bypass all voltages with sufficient capacitance (22µF recommended) at the backplane to minimize the effect of power transients. It is also suggested that power at individual ICs be bypassed with high frequency filter capacitors (0.1 to 0.33µF), and that the power pins are also bypassed with the high frequency capacitors near the backplane. The high frequency capacitors help dampen ground bounce problems and smooth out any high frequency ripple on the local power grids. Boards designed for STD 32 should have solid ground and power planes. They should utilize connections to all ground and power pins available on the backplane.

5.2.2 CONNECTOR ELECTRICAL RATINGS

The connector will be comprised of pins with the following capabilities:

Table 5-2. Connector Specifications.

Current Carrying Capacity	1.0 Amps / pin
Voltage Rating	600VDC
Contact Resistance	30-40mΩ

5.3. A.C. SIGNAL SPECIFICATIONS

The characteristics of a signal on the backplane are determined by three items:

1. Backplane geometry (layout, termination)
2. The characteristics of the device driving the signal
3. The type and number of receivers on the signal

Capacitive loading on the backplane should be kept to a minimum by reducing the number of loads on each signal and keeping the trace lengths from the backplane to a minimum. Backplane signals being used by a board should be less than two inches in length to minimize capacitive loading.

5.3.1. CHARACTERISTIC IMPEDANCE

Backplane geometry determines the characteristic impedance (Zo) seen by a signal. This impedance drops as connectors and terminations are added to the backplane. The "effective impedance" (Zo') seen by a driver is the characteristic impedance lowered by connectors, peripheral cards, and backplane terminations. The lower this impedance becomes, the harder it is to dynamically drive a signal above TTL threshold levels. Zo' must be greater than 20Ω for a fully loaded backplane.

The goal in designing a backplane is to maintain the highest characteristic impedance as is practically possible. This typically will range between 50 and 100Ω and is directly related to the signal trace width, height, and distance from the nearest Vcc or GND plane. Zo can be increased by minimizing trace width and height (depending upon your current requirements) and maximizing its distance from Vcc and GND planes.

5.3.2. SIGNAL TERMINATIONS

Diode Termination

To minimize signal reflections, a diode termination can be placed at both ends of the backplane signal trace to absorb undershoots and overshoots. Figure 5-1 "Diode Termination" shows an example of diode termination. Several IC vendors (for example, Texas Instruments) offer diode termination arrays explicitly for this purpose. Diode termination is the preferred backplane termination mechanism for STD 32.

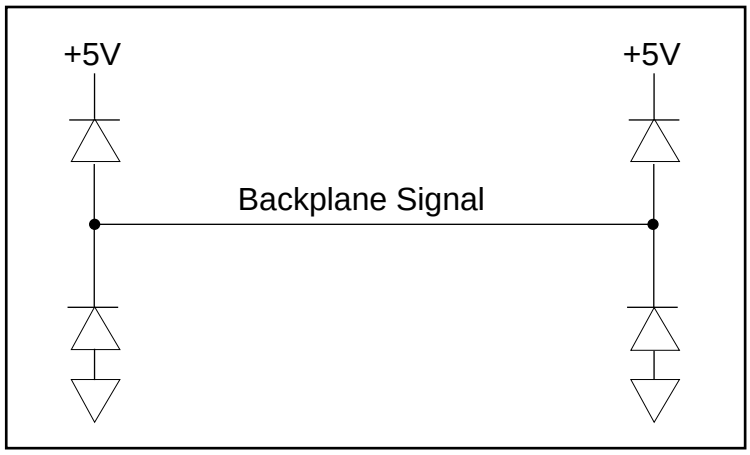


Figure 5-1. Diode Termination.

RC Termination

To minimize signal reflections, an RC termination network that matches the backplane impedance can be used. Unfortunately, the effective impedance of the backplane varies as peripheral cards are added or removed from the system. This means that the terminations should be calculated for typical loading. It is suggested that RC terminations be chosen to match the characteristic impedance as closely as possible. This is especially important for large backplanes. Shorter backplanes may not require any termination at all. Figure 5-2 "RC Signal Termination" shows an example of RC termination.

Open collector signals (e.g., WAITRQ* and MEM16*) should NOT be RC terminated since the corresponding increase in rise time may cause incompatible timings. These signals should be terminated by the Permanent Master with pullup resistors as specified in Table 5-5 "Signal Drivers, Receivers, and Terminations".

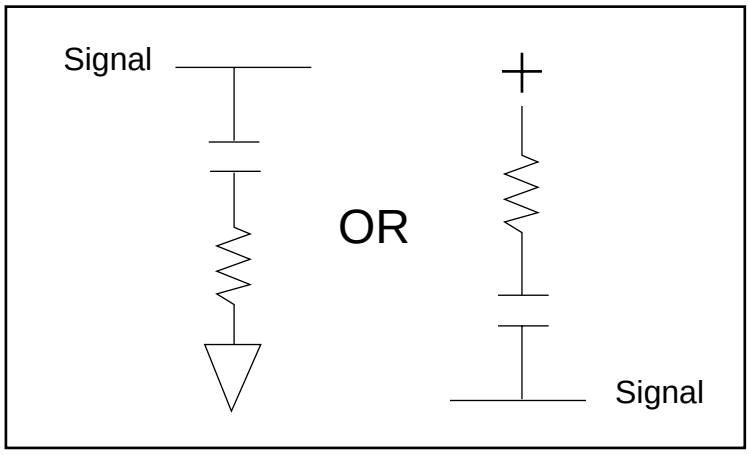


Figure 5-2. RC Signal Terminations.

Other Considerations

Slot-specific signals should NOT be terminated. These signals consist of AENx*, DAKx*, MAKx*, MREQx*, DREQx*, and IRQx. It is suggested that series termination be used on these signals at the source (driver) to minimize undershoot and overshoot at the receiving device.

5.4. D.C. SIGNAL SPECIFICATIONS

There are three types of drivers used in STD 32 systems: three-state, totem pole, and open-collector.

Three-state: Three-state drivers can be driven to either a high level, low level or a high-impedance state so that other drivers can drive the signal. Only one of these drivers on each signal can be active at any one time, or contention may result, with possible damage to the driver.

Totem pole: Totem pole drivers can drive a signal to either a high steady state level or a low steady state level. They are different than three-state drivers in that they cannot go to a high impedance state to allow other drivers to be active on the same signal. For this reason, signals driven by a totem pole driver can have only one driver.

Open-collector: Open-collector drivers are capable only of driving a signal low and sinking current. When not driving, this type of driver does not drive any appreciable current and the level of the signal is determined by an external device. Typically a pullup resistor is used which acts to pull the signal high with a time constant (RC) equal to its value (R) multiplied by the load capacitance (C). These drivers are used when multiple sources, such as an interrupt line or wait request line, might need to drive the signal active.

5.4.1. DRIVER AND RECEIVER CHARACTERISTICS

The characteristics of the drivers and receivers differ depending upon the type of system being designed. The scope of this section includes the design of TTL (typical) systems and CMOS (low power) systems, based upon the use of TTL compatible drivers.

CHAPTER 5

TTL Systems

STD 32 boards must use drivers and receivers that guarantee the following characteristics:

Table 5-3. TTL Driver Characteristics.

V_{OL}	Steady State Driver	low output level	≤ 0.5V
V_{OH}	Steady State Driver	high output level	≥ 2.4V
I_{OL}	Steady State Sink Current	(all drivers)	24mA Min[1]
I_{OH}	Steady State Source Current	(totem pole,three-state)	-2mA Min
Z_{OL}[2]	Output Impedance	Low State	5Ω
Z_{OH}[2]	Output Impedance	High State	≤ 13Ω

1 Slot-specific signal (MREQx*,MAKx*, AENx*, IRQ*, DREQx*, DAKx*) steady state sink current (I_{OL}) requirements are relaxed to 5mA min.

2 Output impedance for CLOCK*, RD*, WR*, INTAK*, ALE*, DMAIOR*, DMAIOW*, START*, and CMD*, assuming $3.5V_{OH}$.

Negative currents indicate flow out of a node; positive currents indicate flow into a node.

Table 5-4. TTL Receiver Characteristics.

V_{IL}	Steady State Receiver	low input level	0.8V
V_{IH}	Steady State Receiver	high input level	≥ 2.0V
I_{IL}	Steady State Source Current	(all receivers)	-0.8mA Max[1]
I_{IH}	Steady State Sink Current	(all receivers)	60µA Max[1]

1 Open-collector pullups are to be located at the Permanent Master, with the exception of DCPDN*, which will have the pullups located at each source. Signals must not be driven to a steady state level higher than the voltage available at the +5V power pin, or to a lower steady state voltage than the ground pin.

Capacitive Loading

It is recommended that each board within the STD 32 system be limited in capacitance such that all drivers, receivers, transceivers, and trace capacitance do not exceed 20pF.

CMOS Systems

It is recommended that systems designed for low power usage be based on technology that provides TTL-compatible drivers and receivers. Backplanes designed for CMOS systems will have pullups so that CMOS inputs will not be allowed to float.

5.4.2. BUS DRIVER AND RECEIVER LOCATIONS

Table 5-5. Signal Drivers, Receivers, and Terminations.

Bus Signals	Type[1]	Driver Locations	Receiver Locations	Resistance Termination[2, 5, 6]	
				Location	Value
D0-D31 (32 Lines)	TRI	Masters and Slaves	Masters and Slaves	All Masters	1.8kΩ ±10%
A0-A23	TRI	Masters	Slaves	All Masters	1.8kΩ[7] ±10%
XA24* - XA31*	TRI	Masters	Slaves	All Masters	1.8kΩ ±10%
WR*, RD*, MEMRQ*, IOEXP, BHE*, ALE*, STATUS0*, STATUS1*, IORQ*, INTAK*, W-R, BE0*-BE3*, M-IO, DMAIOW*, DMAIOR*	TRI	Masters	Masters and Slaves	All Masters	1.8kΩ ±10%
BUSRQ*, NMIRQ*, PBRESET*,	O.C.	Masters and Slaves	Permanent Master	Permanent Master	316Ω ±7%
SLBURST*, NOWS*, WAITRQ*, MEM16*, IO16*, EXRDY	O.C.	Slaves	Masters	Permanent Master	316Ω ±7%
DCPDN*[3]	O.C. N.G.	Masters and Slaves	Masters and Slaves	Source	10kΩ ±10%
BUSAK*	TPL	Permanent Master	Temporary Master		
MAKx* < 50µA leakage	TPL	Slot X	Temporary Master	Temporary Master	10kΩ[4] ±10%
EX8*, EX16*, EX32*	O.C.	Masters and Slaves	Masters	Permanent Master	316Ω ±7%

Table 5-5. Signal Drivers, Receivers, and Terminations (Continued).

Bus Signals	Type[1]	Driver Locations	Receiver Locations	Resistance Termination[2]	
				Location	**Value**
MSBURST*, START*	TRI	Masters	Masters and Slaves	All Masters	1.8kΩ ±10%
MREQx*	TPL	Temporary Master	Slot X	Slot X	10kΩ ±10%
MASTER16*	O.C.	Masters	Permanent Master	Permanent Master	316Ω ±7%
IRQx	TPL	Masters and Slaves	Slot X	Slot X	10kΩ ±10%
CMD*, SYSRESET*, CLOCK*	TPL	Permanent Master	Masters and Slaves		
DAKx*	TPL	Slot X	Masters and Slaves	Masters and Slaves	10kΩ ±10%
AENx* ≤ 50µA leakage	TPL	Slot X	Masters and Slaves	Masters and Slaves	10kΩ[4] ±10%
DREQx* ≤ 50µA leakage	TPL	Slaves	Slot X	Slot X	10kΩ ±10%
T-C	TRI	Masters and Slaves	Masters and Slaves		
LOCK*	TRI	Masters	Masters	All Masters	1.8kΩ ±10%
INTRQ*, INTRQ1*, CNTRL (INTRQ2*), INTRQ3*	O.C.	Masters and Slaves	Masters and Slaves	Permanent Master	316Ω ±7%

1 TRI = Three-State Drive , O.C. = Open Collector, N.G. = Non-Glitching, TPL = Totem Pole

2 Termination resistors of 100kΩ or greater may be used on any signal for test purposes.

3 Not meant to be driven or used by TTL gates that glitch on power-up or power-down. Use discrete transistors or battery-backed CMOS gates to prevent glitches.

4 Pulldown to ground.

5 The Permanent Master must resistively terminate all signals that could be managed by a Temporary Master in response to a BUSRQ* or MREQ*.

6 316Ω terminations can be implemented with 300Ω±1% or 330Ω±2% resistors.

CHAPTER 6

MECHANICAL SPECIFICATIONS

6.1. INTRODUCTION

This chapter lists and defines the mechanical specifications that must be followed to ensure that an STD 32 system is dimensionally correct and compatible with STD-80 series boards. This chapter provides specifications for board dimensions, backplanes and connectors.

6.2. BOARD DIMENSIONS

The printed circuit board size and outline is defined in Figure 6-1 "STD-80 Series PCB Outline" for STD-80 series cards. The figures "STD 32 PCB Outline (with Extensions)" (Fig.6-2) and "STD 32 PCB Outline (without Extensions)" (Fig.6-3) define the mechanical dimensions for STD 32 series cards. STD 32 cards that do not support extended signals appear similar to STD-80 series cards but have tighter tolerances. All EA and SA I/O boards, and SA masters should use Figure 6-2 "STD 32 PCB Outline (with Extensions)". EA masters and memory boards use Figure 6-3 "STD 32 PCB Outline (without Extensions)". The extended signals are necessary only for EA memory transfers. These three figures are provided for comparison purposes.

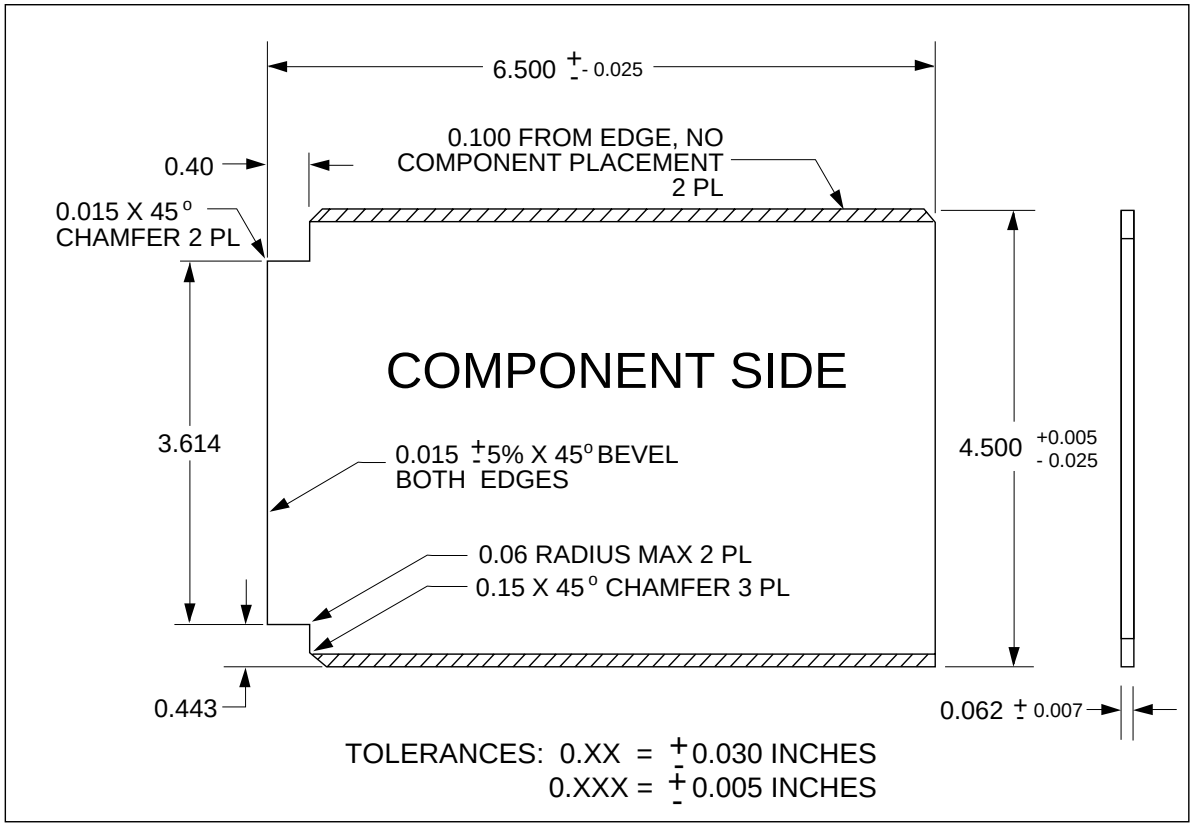


Figure 6-1. STD-80 Series PCB Outline.

6.2.1. PCB LAYOUT RESTRICTIONS

STD boards are intended to be supported along both sides of the card by card guides. The card guides can be a maximum of 0.1 inch deep. Therefore, no components are allowed within 0.1 inch of the card edge, as illustrated in Figures 6-1, 6-2 and 6-3.

Section 2.3 defines a tooling hole that could be used to mount a card edge bracket. If used, components must not be placed within 0.1 inch of the tooling hole for mounting tabs.

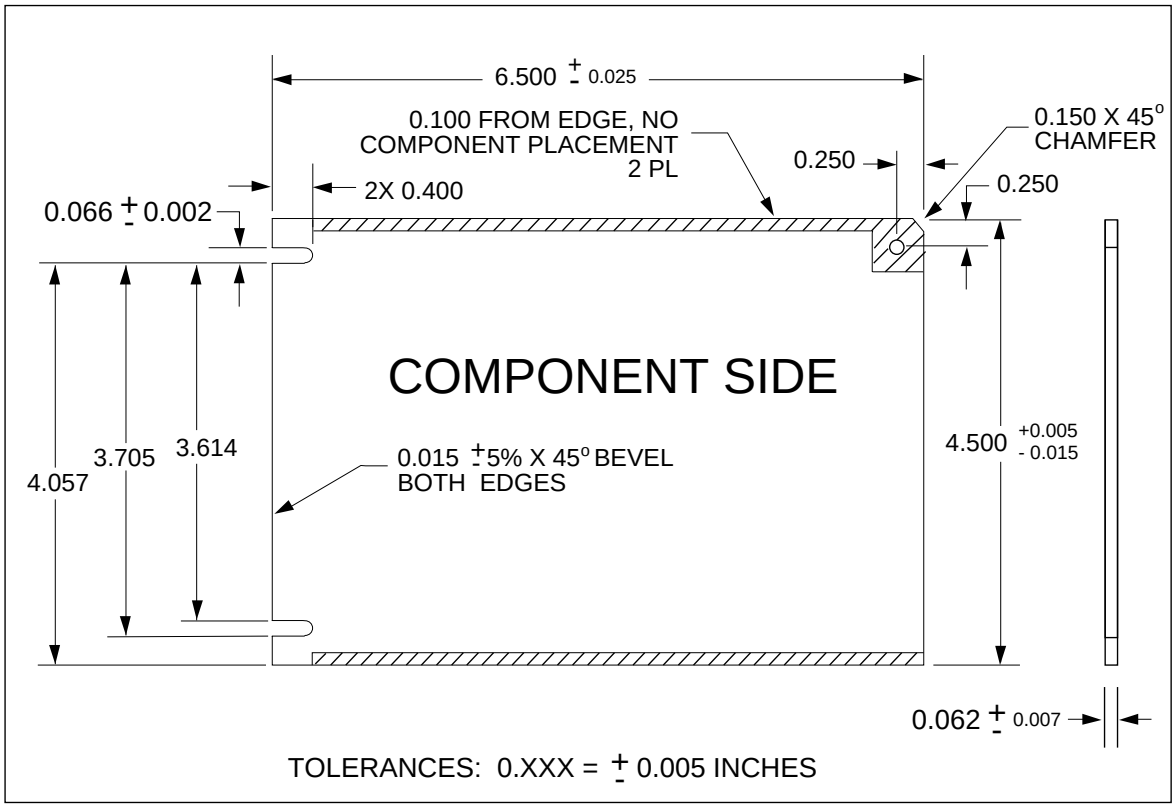


Figure 6-2. STD 32 PCB Outline (with Extensions).

6.2.2. PCB LENGTH RESTRICTIONS

The length of an STD card is 6.5 inches maximum. Extensions or porches that extend beyond 6.5 inches are potentially incompatible with future STD bus enclosures, mounting brackets and card cages.

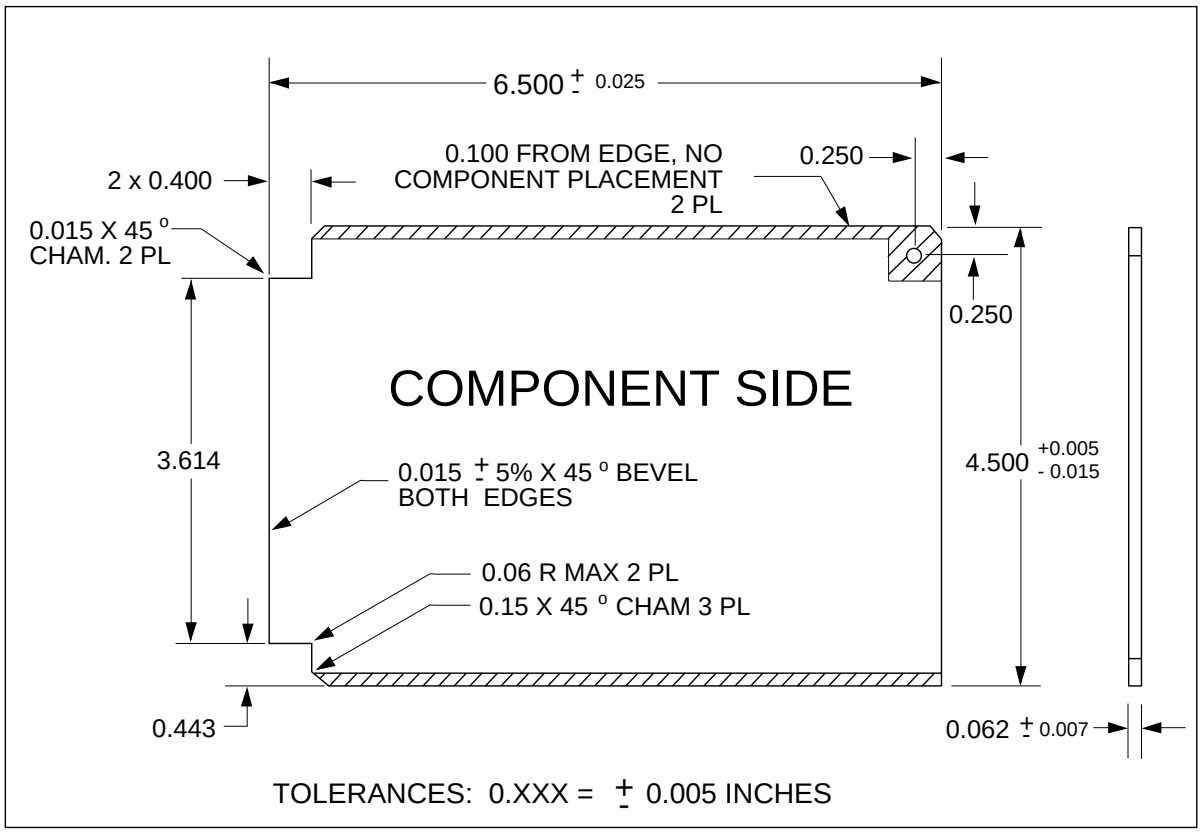


Figure 6-3. STD 32 PCB Outline (without Extensions).

6.2.3. PCB TOOLING HOLE

Each STD 32 board should include an extractor that can extend beyond the 6.5 inches maximum length. The tooling hole location and size for mounting the extractor are specified in Figure 6-4 "STD 32 PCB Tooling Hole Locations". These dimensions must be followed for STD 32 compatibility. Additional tooling holes are suggested for manufacturing purposes. Typically, tooling holes are provided at diagonal locations of a PCB to aid board fabrication procedures. The location of any additional tooling holes is at the discretion of the board designer.

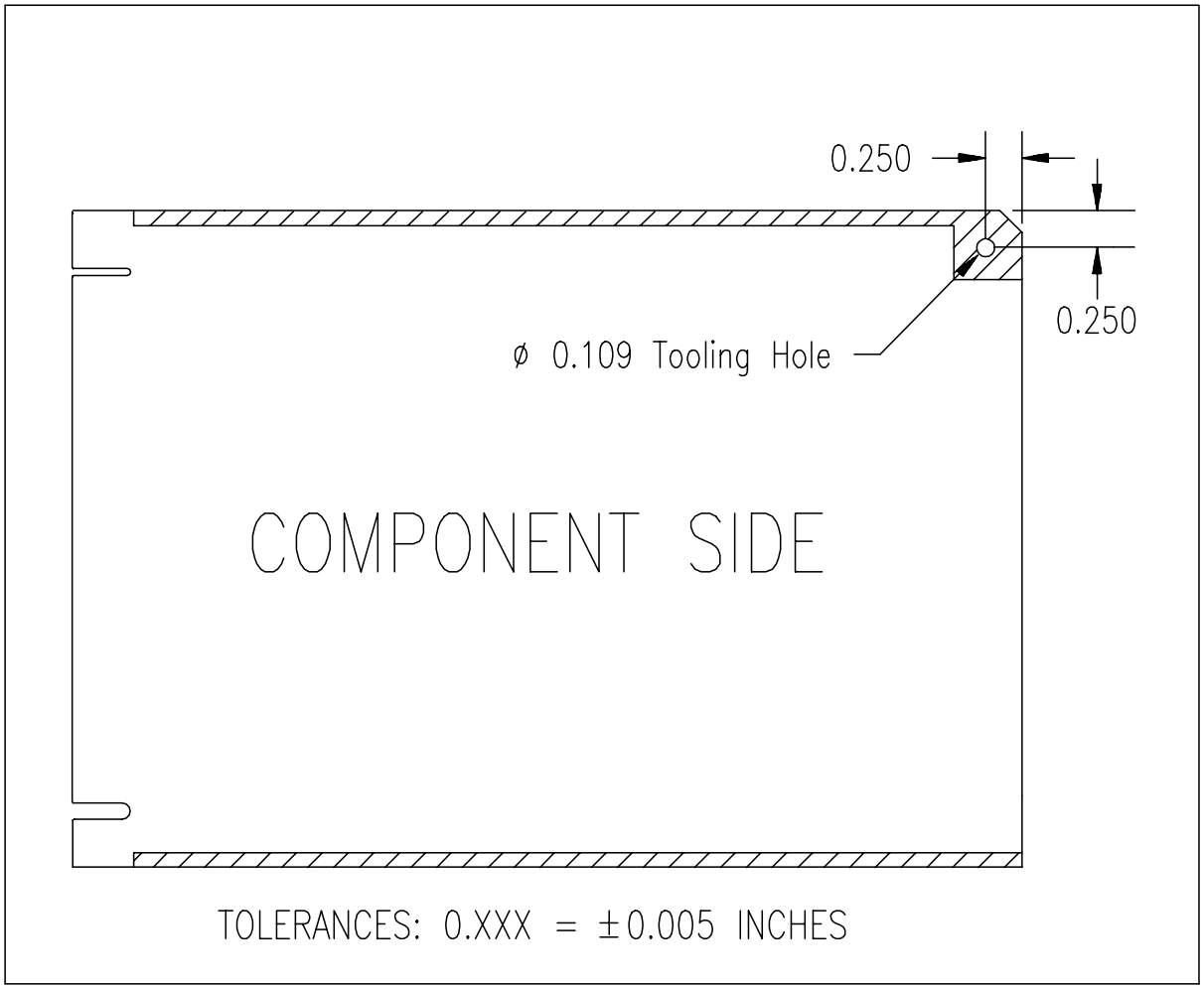


Figure 6-4. STD 32 PCB Tooling Hole Locations.

6.3. BACKPLANE

The STD backplane is an important component in the system. The backplane will connect most of the STD slot signals in parallel for STD-80 series systems utilizing the P interface signals. Figure 6-5 "STD-80 Series Backplane Block Diagram" illustrates the block diagram for an STD-80 series backplane. Most of the P interface signals for STD 32 series systems are also connected in parallel. The additional E interface signals supported by STD 32 series systems contain both slot-specific signals and parallel signals connected to each slot.

The E slot-specific interface signals are connected to the end slot (Slot X) on the backplane for use by the Permanent Master. The end slot is defined as the left-most slot when viewing the backplane from the side through which cards are inserted with the extractor oriented up. Figure 6-6 "STD 32 Backplane Block Diagram" is a block diagram of an STD 32 backplane.

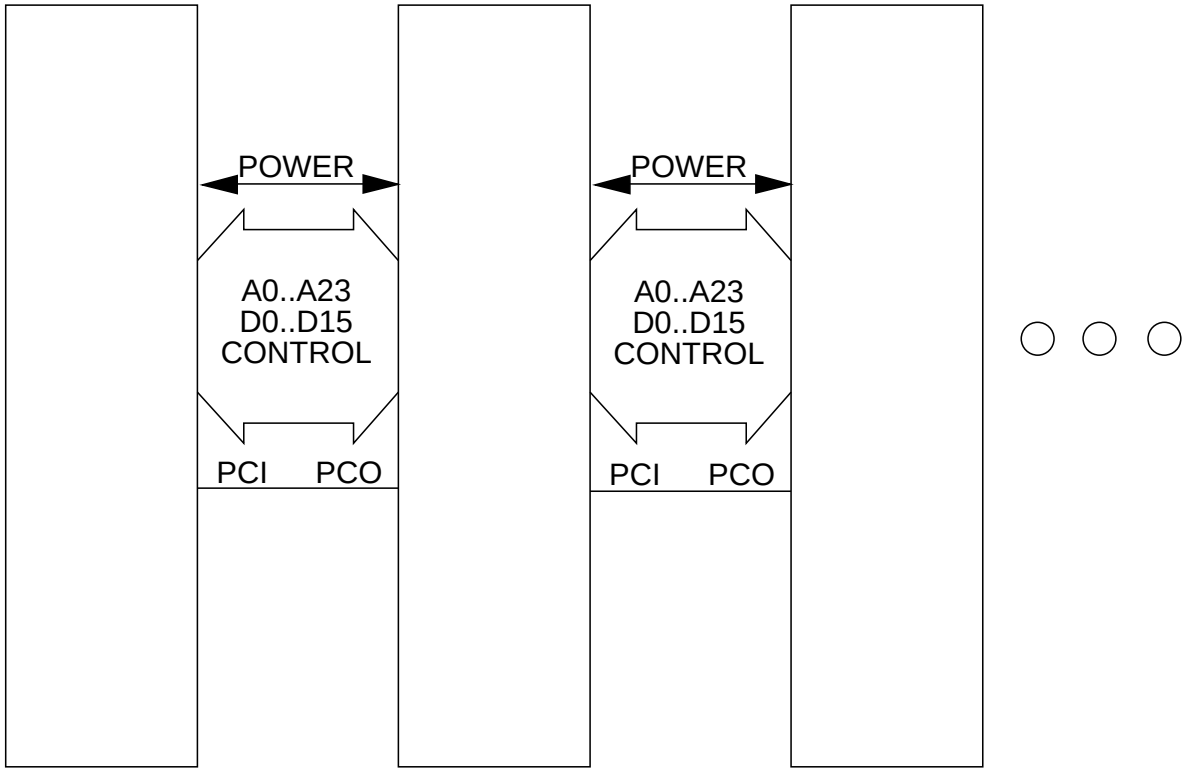


Figure 6-5. STD-80 Series Backplane Block Diagram.

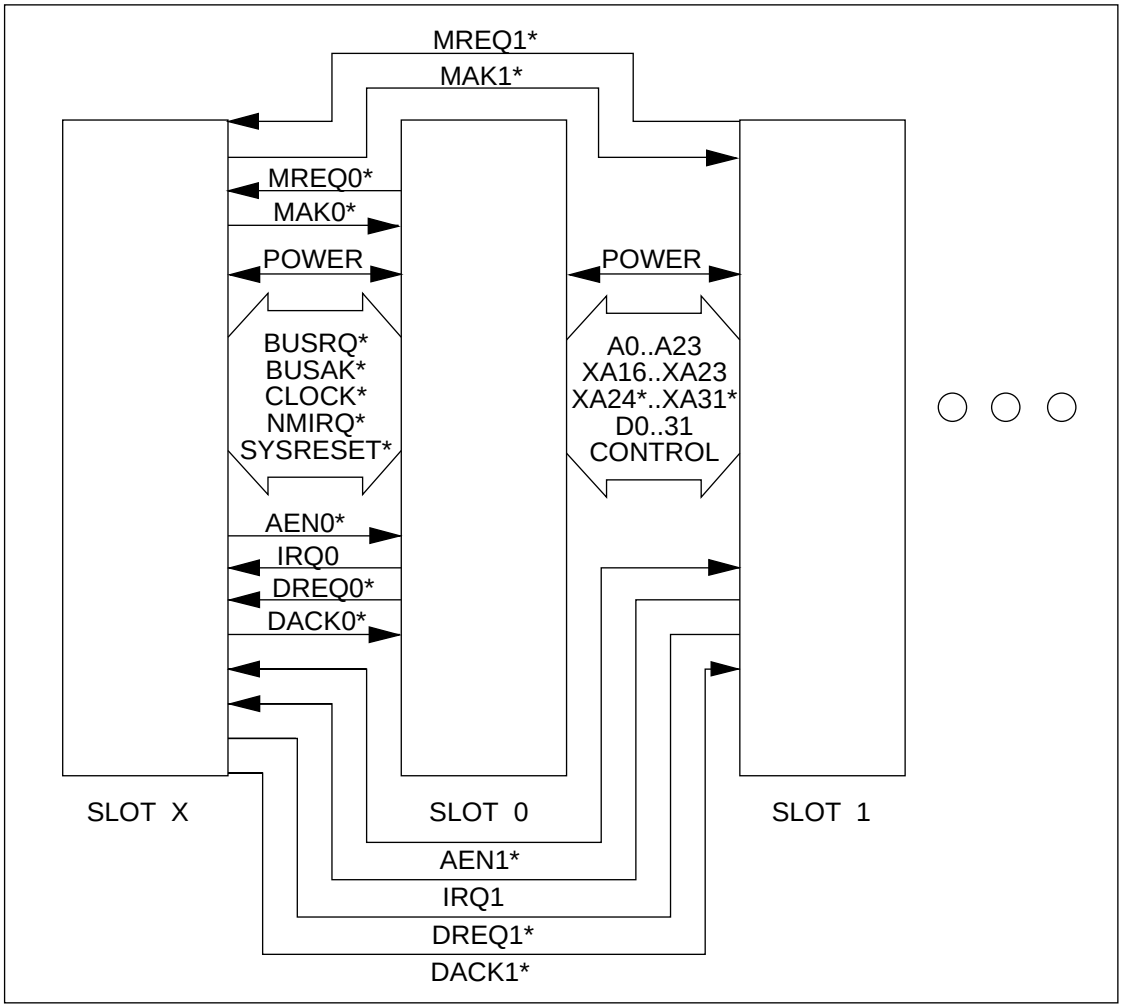


Figure 6-6. STD 32 Backplane Block Diagram.

CHAPTER 6

Table 6-1 lists the STD 32 backplane characteristics.

Table 6-1. STD 32 Backplane Characteristics.

Minimum center-to-center trace spacing on backplane	16 mil minimum
Minimum trace width	8 mil
Number of planes	5 max
Unloaded Impedance (with connectors)	55Ω min
Termination (non-CMOS systems)	R-C to power or ground
Termination (CMOS systems)	100kΩ pullup
Copper Thickness	2 oz
Backplane Thickness	0.093 inch

Figure 6-7 "STD 32 Backplane Layer Thickness" indicates backplane layer stacking and spacing requirements. Note that 2 oz. copper is used for all layers.

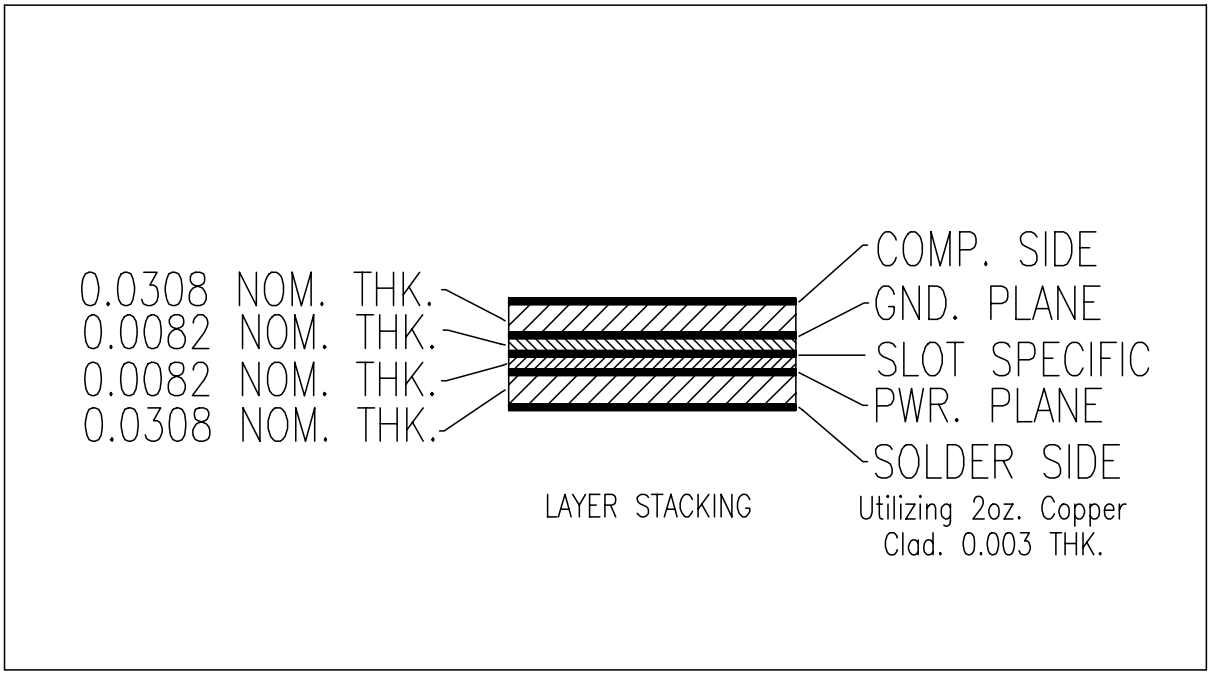


Figure 6-7. STD 32 Backplane Layer Thickness.

6.3.1. POWER CONNECTION

Power connections to STD-80 series backplanes can be made many different ways, as long as good power distribution is maintained across the backplane. Connections may be made by soldering, insulation displacement headers, or screw terminals.

For STD 32 series backplanes, screw terminal locations for power distribution will be provided across the backplane at a maximum of every eight slots. As an option, backplanes could be provided without the screw terminals installed if the power connections are to be soldered.

If screw terminals are installed, an extra 1.5 inches should be allocated for connectors and power wiring between the backplane and any sheet metal or enclosure.

6.3.2. BOARD-TO-BOARD SPACING

The following printed circuit board specifications must be adhered to when designing STD-80 series or STD 32 compatible boards. A minimum board-to-board spacing of 0.5 inch should be assumed when selecting components for complete backward compatibility with 8-bit STD-80 series backplanes and card cages. A minimum board-to-board spacing of 0.625 inch is required for STD 32, allowing for higher components and greater air flow over the cards.

Board-to-board Spacing (LC) - center of boards when plugged into backplane ±.01 inch.

Board Thickness (LT) - the typical board thickness ±0.007 inch.

Component Lead Length (LL) - the length of the component leads below the printed circuit board cannot exceed 0.040 inch maximum.

Component Height (LH) - the following equation is used to determine the maximum height of the component above the printed circuit board:

$$LH < LC - LT - LL$$

Figure 6-8 "STD Board-To-Board Separation" illustrates the parameters used in Table 6-2 "Component Height And Board-To-Board Spacing" for both STD-80 series and STD 32 board-to-board spacing.

Table 6-2. Component Height And Board-To-Board Spacing.

	LC	LT	LL	LH
STD-80	0.50-0.01	0.062+0.007	0.040	<0.381
STD 32	0.625-0.01	0.062+0.007	0.040	<0.506

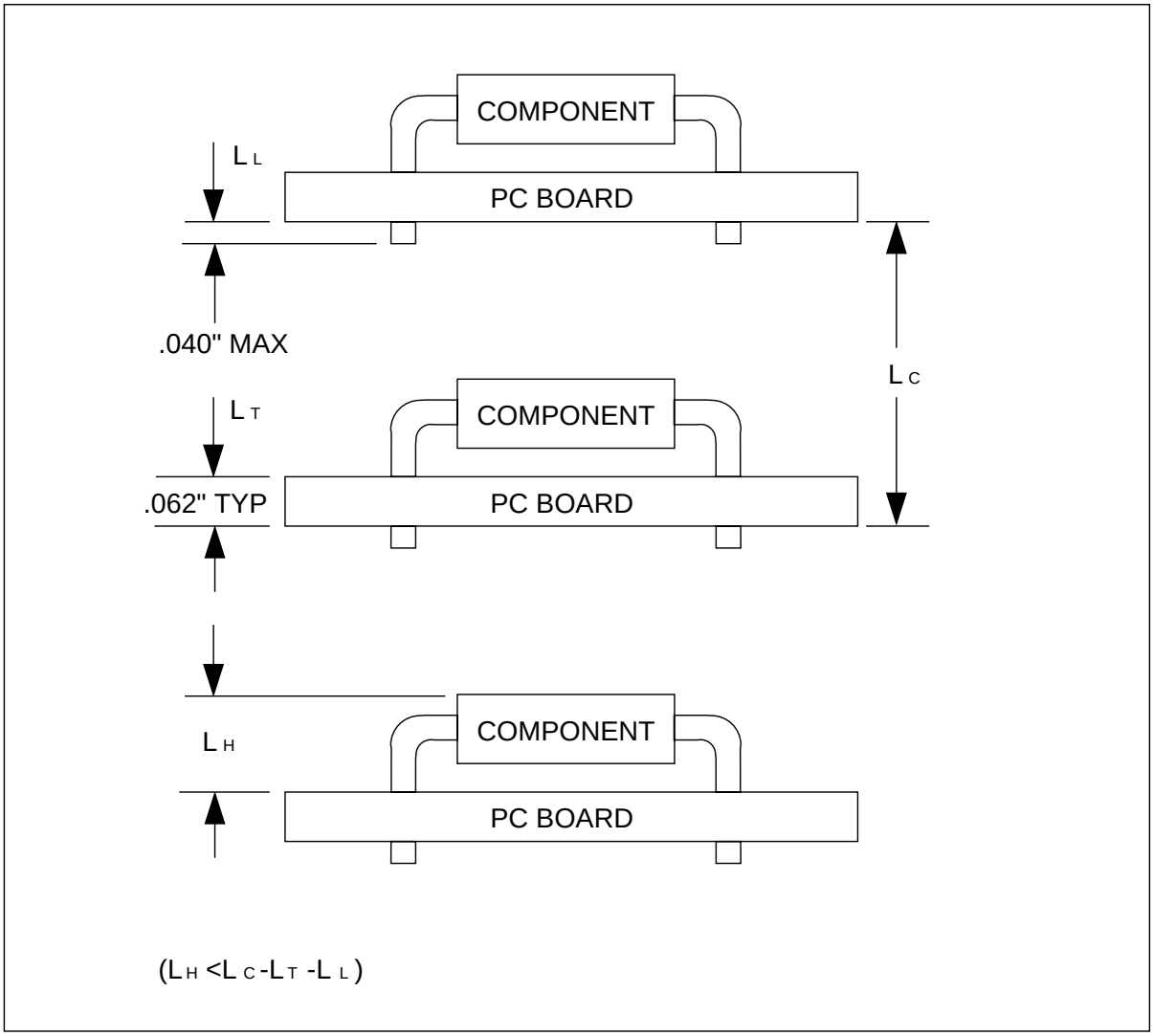


Figure 6-8. STD Board-To-Board Separation.

6.4. CONNECTORS

STD-80 series boards utilize card edge fingers designed to mate with a 56 pin (dual 28 pin) 0.125 inch spacing card edge connector. The card edge fingers are called the P connector. STD 32 boards also utilize card edge fingers designed to mate with a 136 pin (dual 68 pin) 0.0625 inch spacing card edge connector to implement the P and extended (E) signals. Both the P and E mechanical, electrical, and pin assignments are given in the following sections.

The STD-80 series card edge connectors located on the backplane are designated J1, J2, J3, etc. These connectors are dual 28 pin (0.125 inch spacing) card edge connectors. The STD 32 series card edge connectors located on the backplane are designated JE1, JE2, JE3, etc. These connectors are dual 68 (0.0625 inch spacing) card edge connectors.

Odd numbered pins should be located on the component side of the board in ascending order when going from top to bottom as illustrated in Figure 6-15 "STD 32 Pin Assignments (Component Side)" and Figure 6-17 "STD 32 Slot X Pin Assignments (Component Side)". Even numbered pins are on the solder side of the board.

For use in STD 32 backplanes, it is recommended that card edge fingers have a minimum of 30µ inches of gold plating over 200µ inches of nickel.

6.4.1. P STD CONNECTOR

Connectors and printed circuit boards designed for STD-80 series compatibility should meet the following mechanical, electrical, and pinout specifications.

Table 6-3. STD-80 Series J1 Connector Specifications.

Current per Contact	3A Max
Acceptable PCB Thickness	0.062 inch ±0.007 inch
Voltage Drop	30mV max at 3A
Operating Voltage	600 VDC at sea level
Operating Temperature	0°-65° C

Mechanical And Electrical Specifications

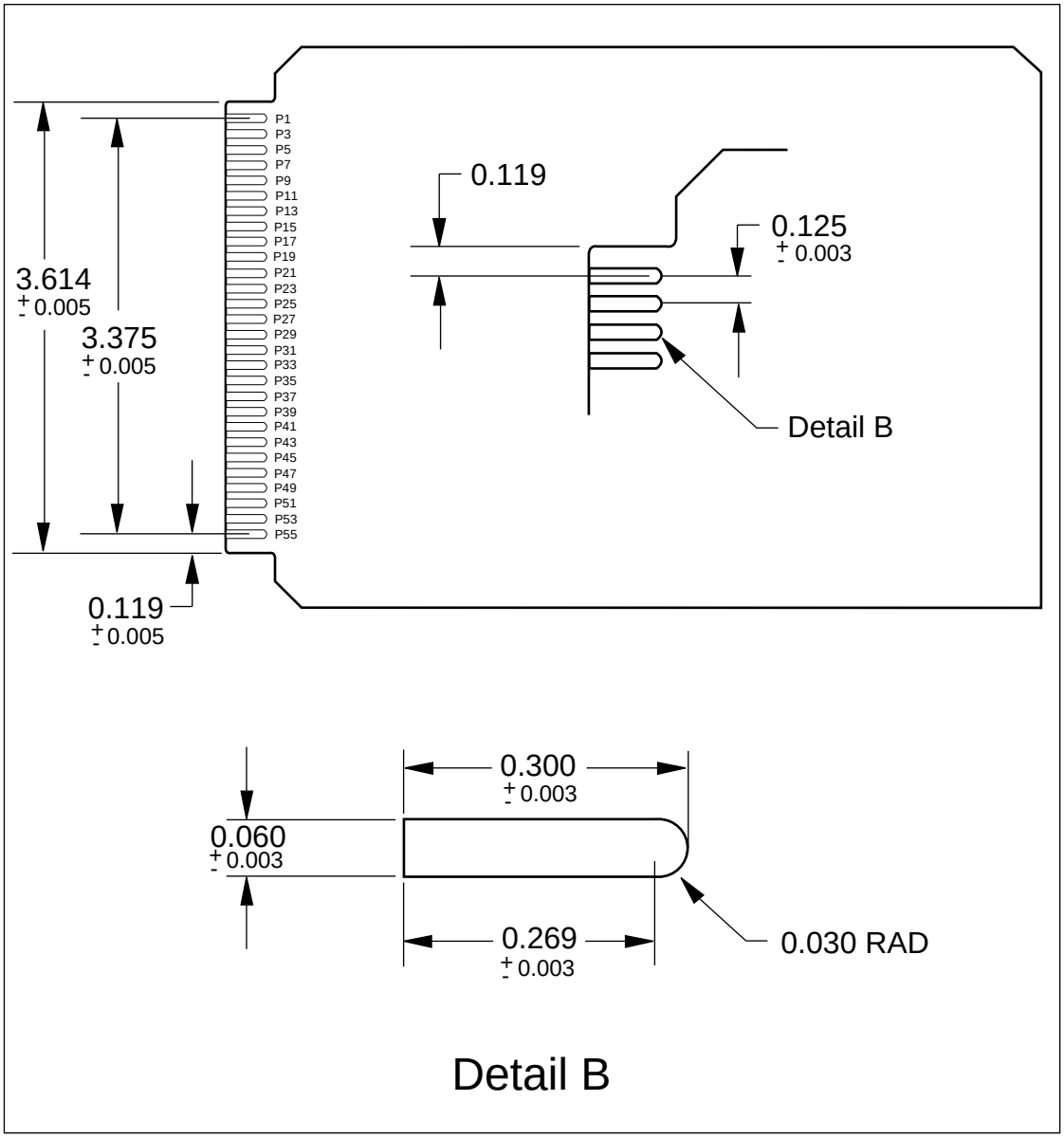


Figure 6-9. STD-80 Series P Finger Design.

CHAPTER 6

Pin Assignments

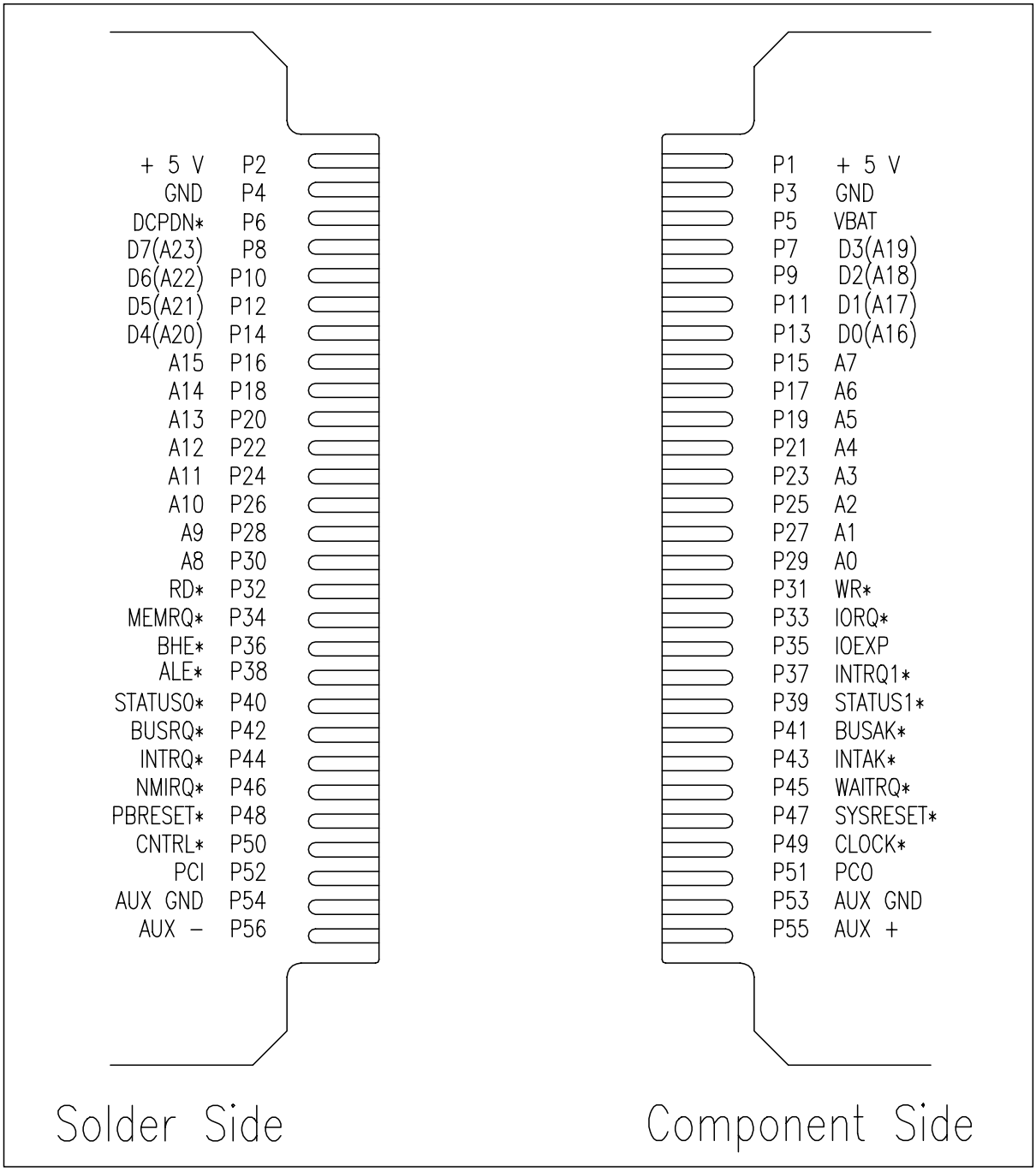


Figure 6-10. STD-80 Series Pin Assignments.

6.4.2. E STD 32 CONNECTOR

Connectors and printed circuit boards designed for STD 32 series compatibility should meet the following mechanical, electrical, and pinout specifications.

Card edge fingers must have a minimum of 30µ inches of gold plating over 200µ inches of nickel.

Mechanical And Electrical Specifications

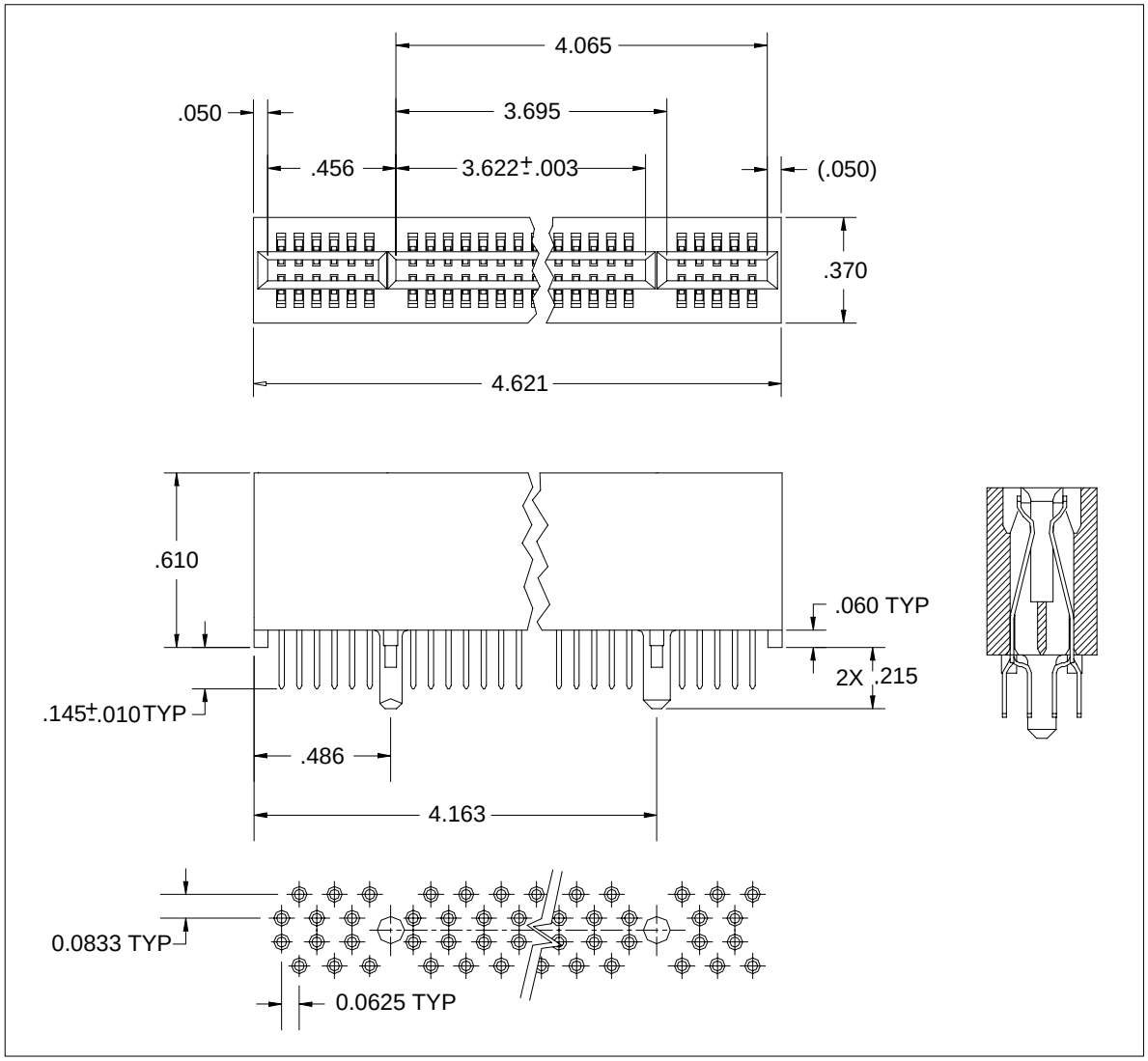


Figure 6-11. STD 32 Motherboard Connector Design.

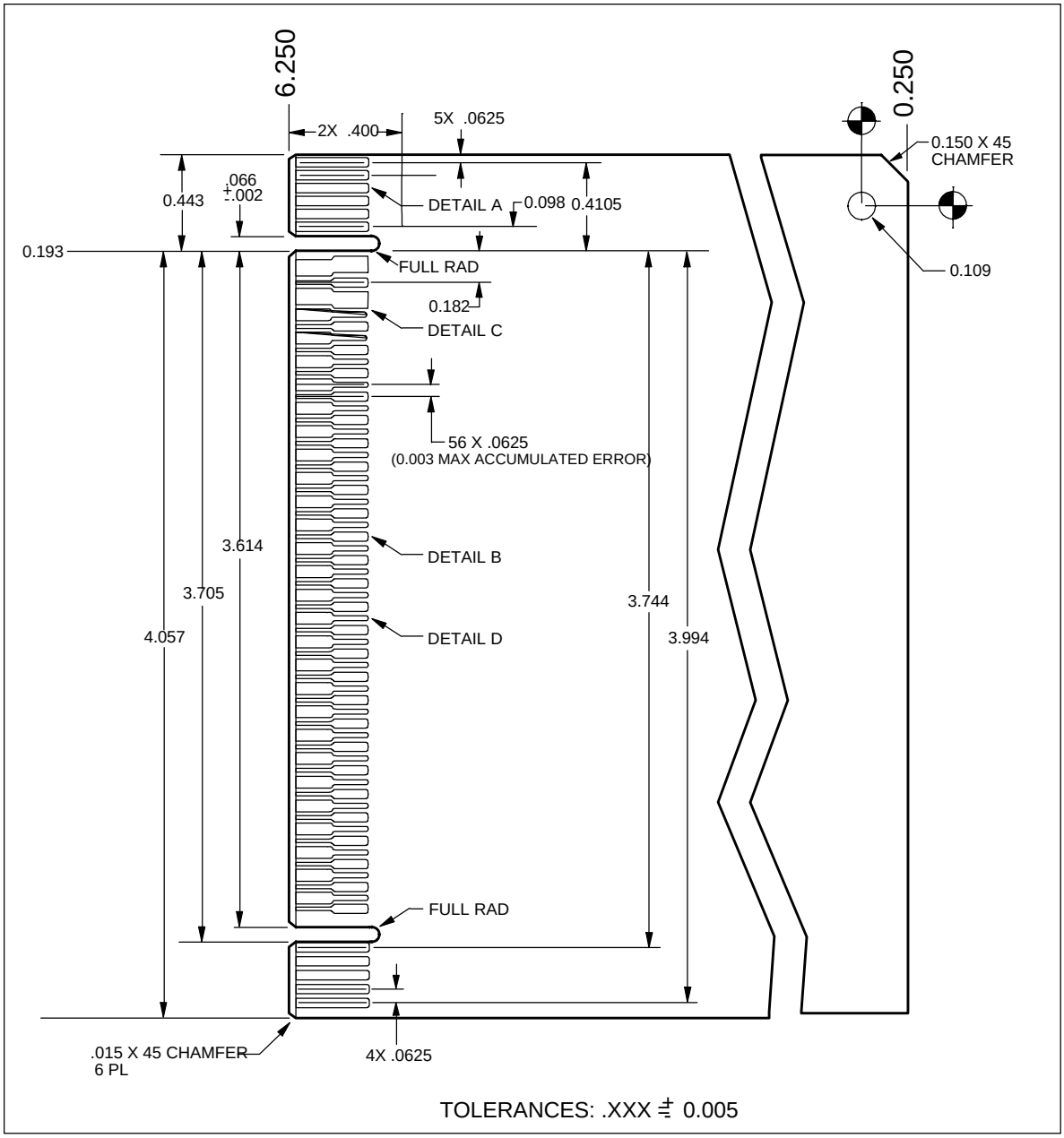


Figure 6-12. STD 32 P/E Board Fabrication (with Extensions). **US Patent 5,061,190**

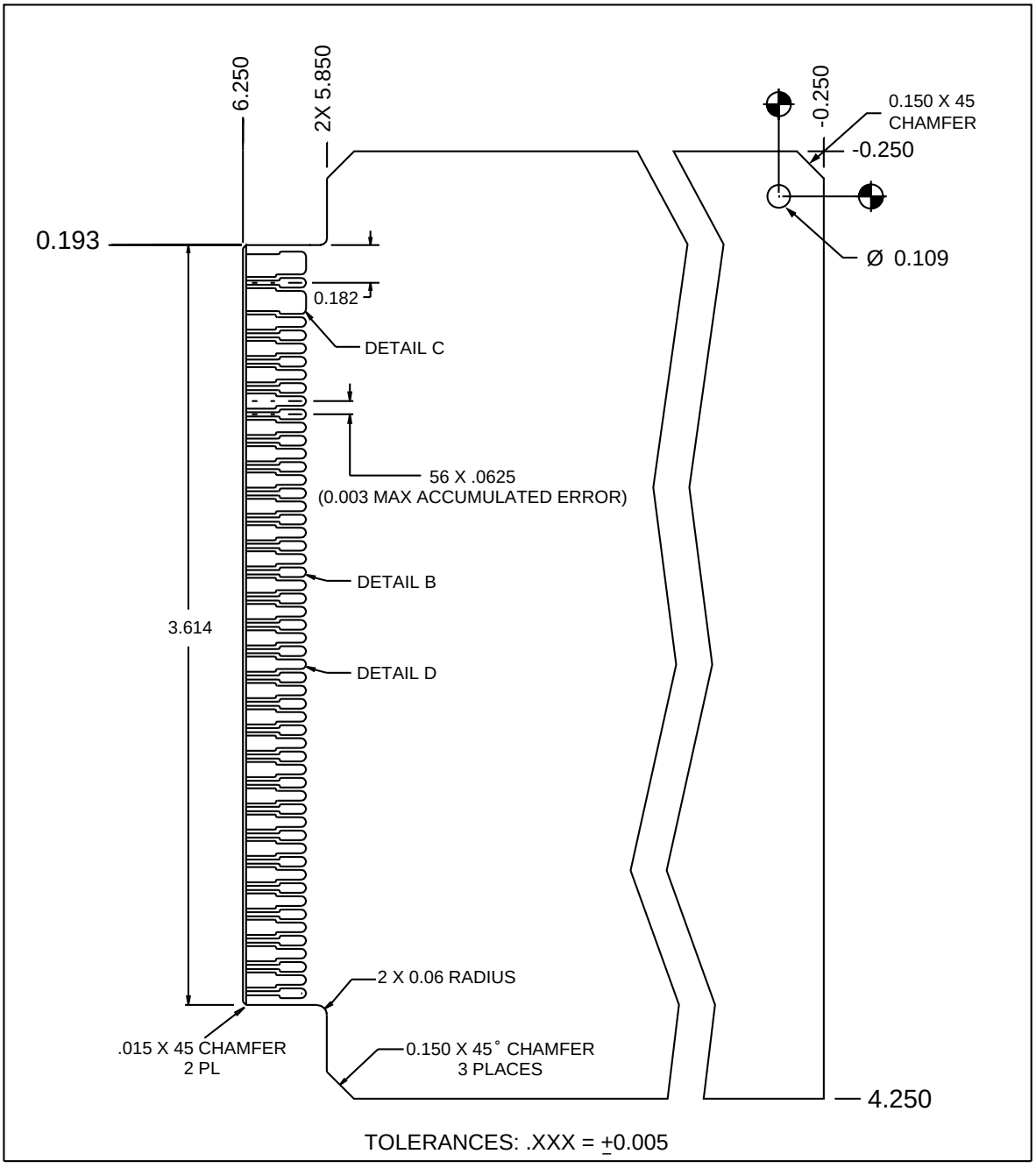


Figure 6-13. STD 32 P/E Board Fabrication (without Extensions). **US Patent 5,061,190**

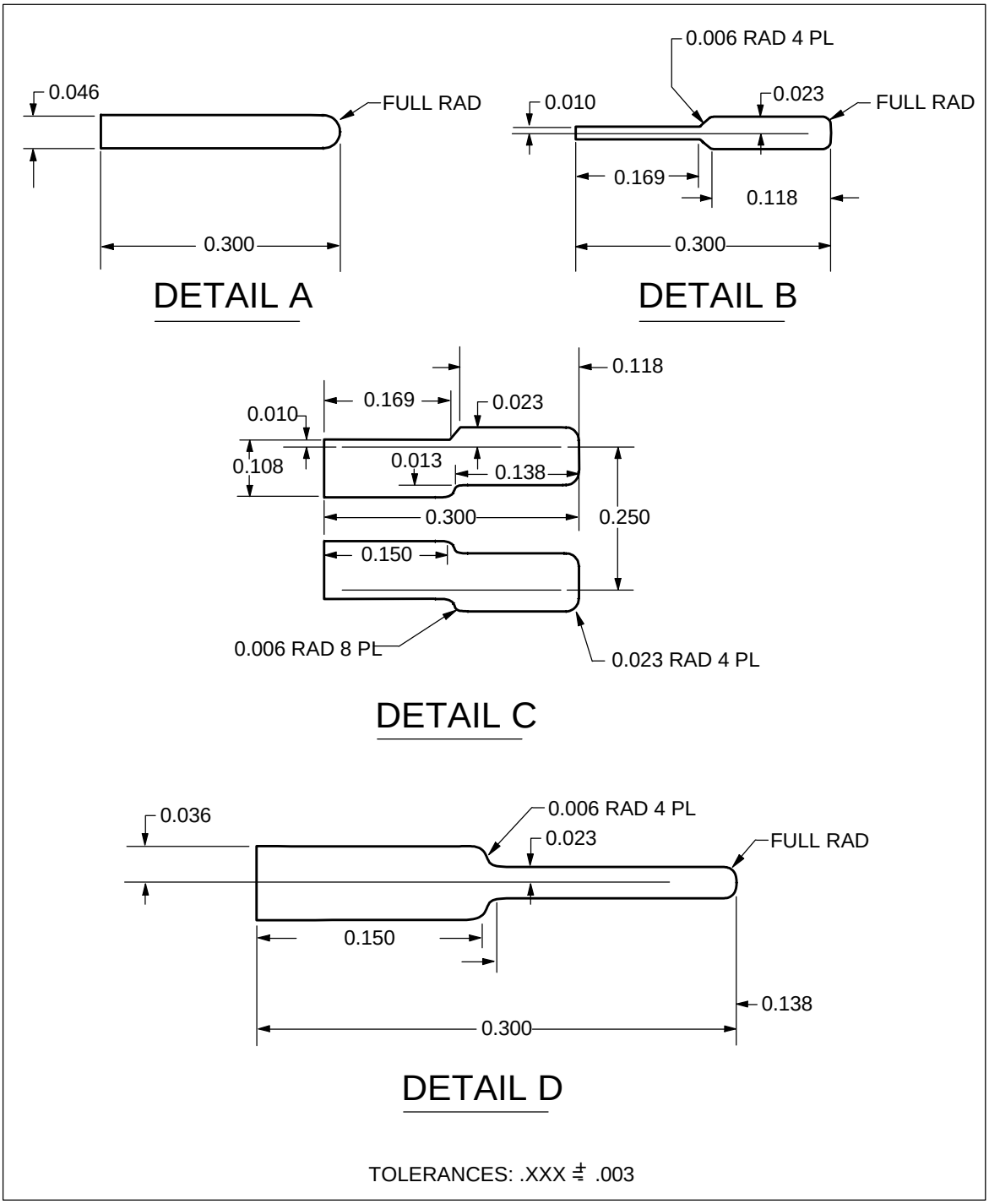


Figure 6-14. STD 32 P/E Finger Dimensions. **US Patent 5,061,190**

Table 6-4. STD 32 P/E Connector Specifications.

Current per Contact	1.0A min
Acceptable PCB Thickness	0.062 inch ±0.007 inch
Operating Temperature	-40 to +85° C
Mating Cycles	500 minimum
Insertion Force	6 ounces per pair
Vibration	10 to 2kHz @ 15Gs / 0.06" displacement
Contact Normal Force	135 grams per contact
Contact Resistance	30mΩ, increasing to 40mΩ through use
Connector Body	Glass filled polyphenylene sulfide, UL 94V-0
Insulation Resistance	50,000MΩ min
Environmental Humidity	0 to 95% with no condensation

Pin Assignments

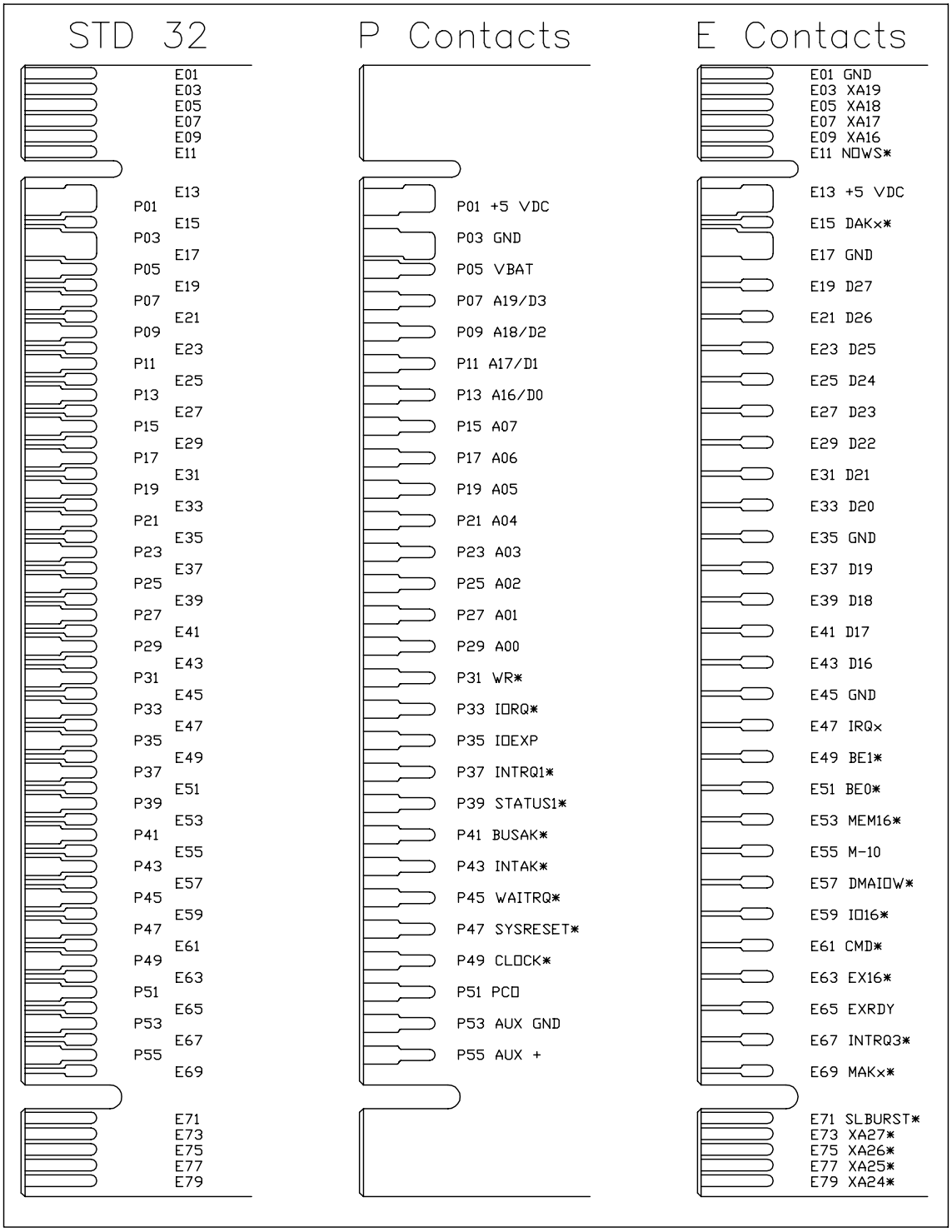


Figure 6-15. STD 32 Pin Assignments (Component Side). **US Patent 5,061,190**

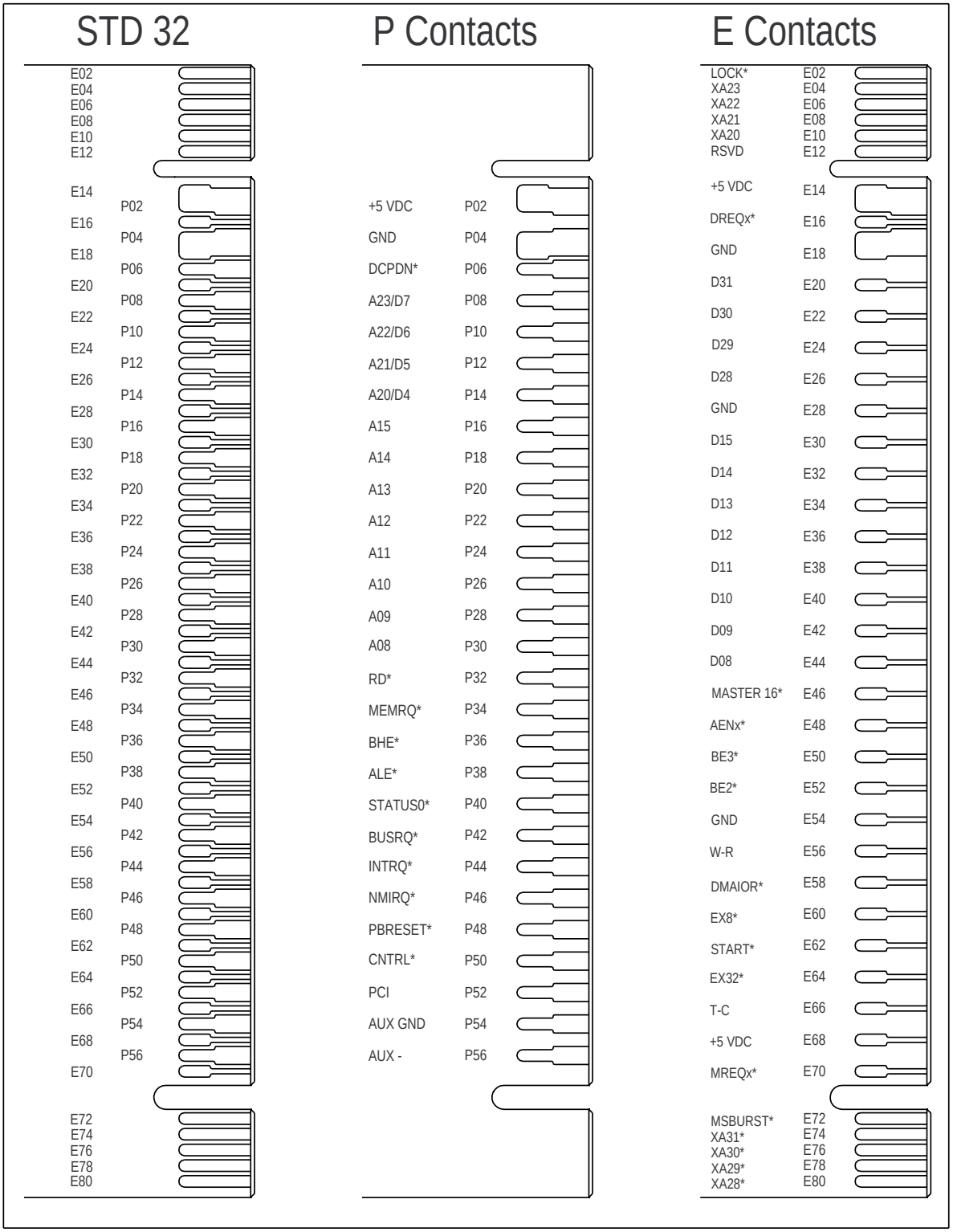


Figure 6-16. STD 32 Pin Assignments (Solder Side). **US Patent 5,061,190**

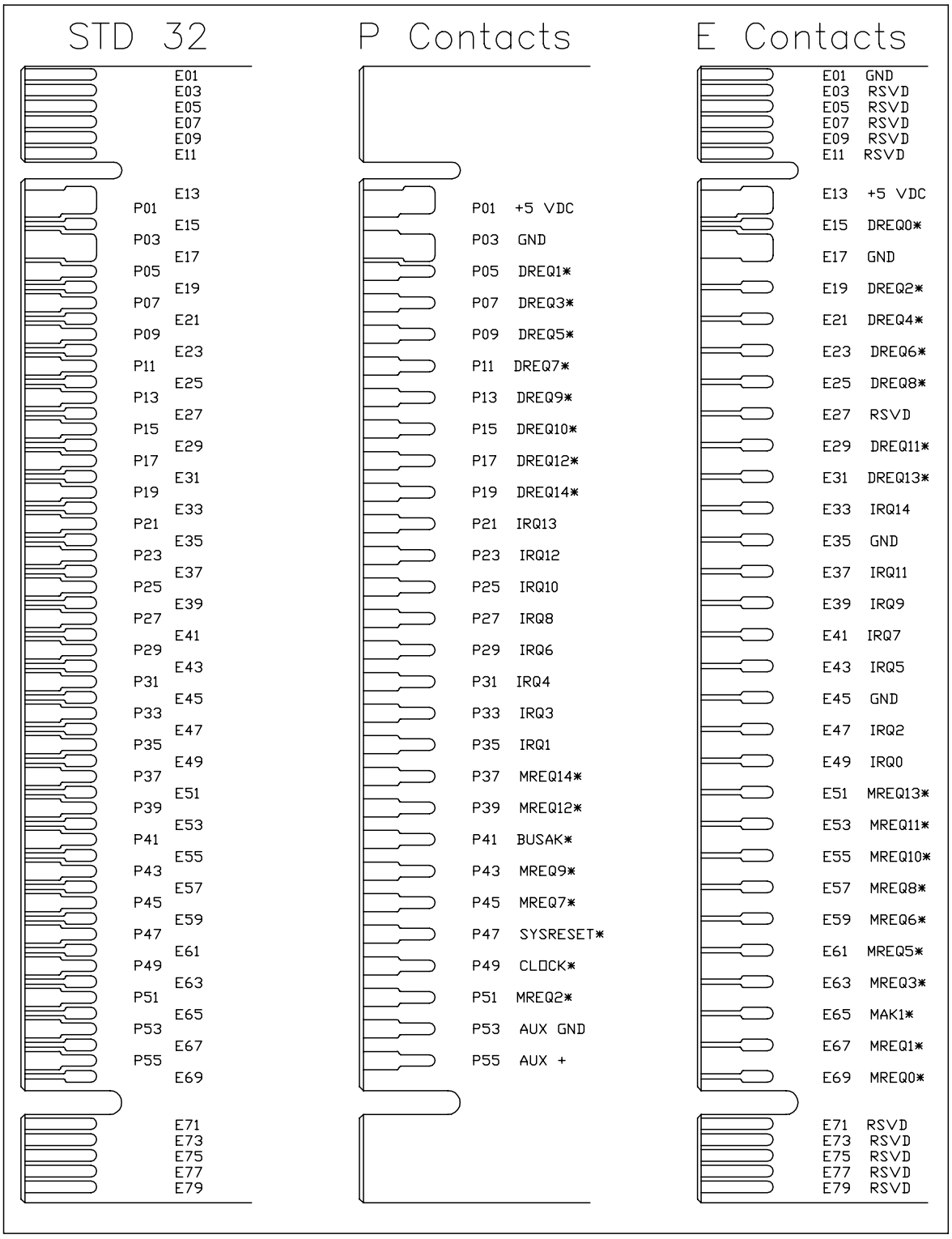


Figure 6-17. STD 32 Slot X Pin Assignments (Component Side). **US Patent 5,061,190**

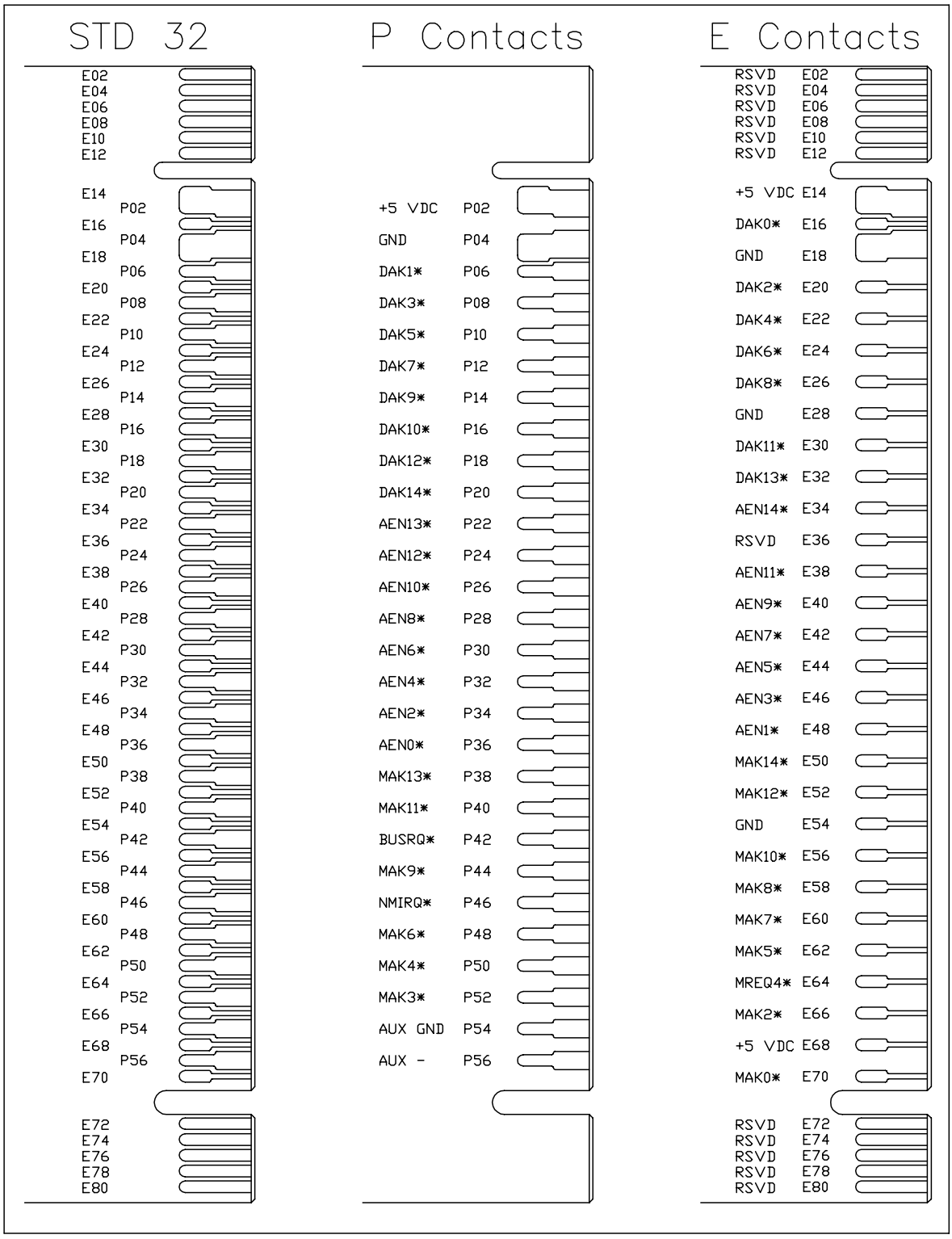


Figure 6-18. STD 32 Slot X Pin Assignments (Solder Side). **US Patent 5,061,190**

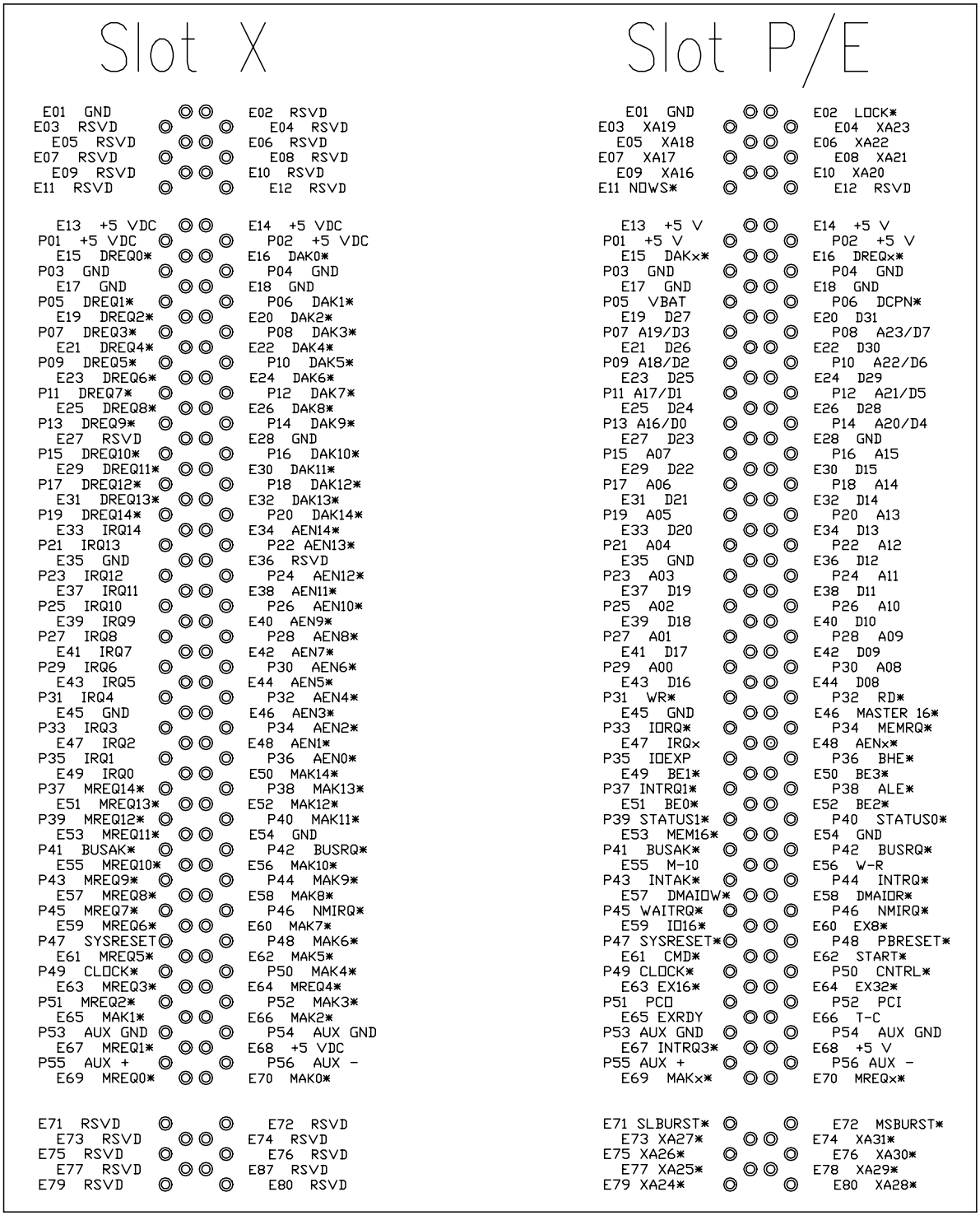


Figure 6-19. STD 32 Backplane Pin Assignments. **US Patent 5,061,190**

APPENDIX A

STD-80 SERIES COMPATIBILITY

STD-80 SERIES COMPATIBILITY

APPENDIX A

A.1. INTRODUCTION

Every effort was made to ensure that STD-80 series boards operate under the 8-bit STD 32 bus specification. However, some differences should be noted. These differences are listed in this appendix.

A.2. SIGNALS

- **Substitution of BHE* for MEMEX*.** Pin 36 of the STD-80 series bus specification is defined to facilitate memory boards decoding only 16 address lines without redundant mapping. Pin 36 of the STD 32 bus specification signals a data transfer across the E bus data lines D8-D15.
- **MCSYNC* becomes ALE*.** Pin 38 of the STD-80 series bus specification is defined as MCSYNC*. The same pin is defined by the STD 32 bus specification as ALE*. The functional difference is that MCSYNC* occurs at the start of every machine cycle while ALE* occurs at the start of machine cycles in which any address line above A7 is different from the previous machine cycle.
- **Timing parameters.** The timing for the STD-80 series bus specification is generated relative to the CLOCK* signal. The timing for the STD 32 bus specification is not entirely clock referenced. In addition to being specified differently, there are some differences in the values themselves. Refer to the timing diagrams in Chapter 2 for more information.

A.3. MECHANICAL

- **Board dimensions.** STD 32 uses the same overall board dimensions as STD-80 series boards except that board tolerances must be held to ±0.005 inch to ensure proper mating with the STD 32 connector.

  STD-80 series boards with the specified ±0.010 inch tolerances will work without any problems in STD 32 backplanes. Only new STD 32 cards using the extended bus signals must hold to ±0.005 inch tolerances.

- **Card edge connectors.** STD 32 uses card edge connectors having 136 contacts spaced on a .0625 inch pitch, mating with matching land patterns on the printed circuit board. Not all boards require the extended land patterns to function, but it is advisable to use the full 114 pattern on the main part to minimize wear on the contacts of unused pins.

  The STD 32 card edge connectors are thoroughly specified for number of insertions, contact force, vibration, shock, and operating temperature. Plating requirements for edge fingers have been added. They are 30µ inches (minimum) of gold over 200µ inches of nickel.

- **Tooling hole definition.** A tooling hole is defined for size and location along the top card edge for future mounting brackets. The tooling hole can be used for an extractor tab when mounting brackets are not used.

- **STD 32 backplane specifications.** The backplane electrical characteristics are now specified. Adherence to the backplane electrical characteristics will help assure reliable operation when using products from more than one backplane manufacturer.

  The board-to-board spacing has been increased from 0.5 inch minimum specified in the STD-80 series specification to a minimum of 0.625 inch. This will allow for higher components and space for more air flow over cards.

A.4. ELECTRICAL

- **Characteristic impedance (Zo').** STD 32 specifies that backplanes should have an unloaded characteristic impedance greater than 55ohm.

- **Termination.** STD 32 recommends diode termination networks for non-Slot X signals to absorb overshoot and undershoot energy and minimize incident wave switching noise.

- **Voltage Specifications.** STD 32 specifies a maximum ripple voltage on power supply lines of 50mV. All other specifications remain the same. Four additional ground connections have been added to increase ground integrity. One additional +5V connection has been added for more uniform voltage feed on both sides of the board.

- **Capacitive Loading.** STD 32 cards should not load the backplane beyond 20pF per signal. This specification will limit the loading of the backplane per signal to no more than 20pF located within two inches of the backplane. Transceivers, such as those used to drive data signals, are typically in the 9 through 12pF range. Therefore, only one transceiver per signal is recommended. The logic family used will be the deciding factor.

A P P E N D I X B

STD 32 DESIGNER'S GUIDE

APPENDIX B

B.1 INTRODUCTION

This designer's guide is a "hands-on" example of designing an I/O card that is compatible with both the simpler STD-80 and the higher performance STD 32 specification. An example design is undertaken that expands upon a generic core of logic that is adaptable to many applications. An I/O design was chosen for the example because many users of the STD bus design a simple I/O card to perform a specific function not available in the general marketplace. The following paragraphs address some specific questions to help define what designing an STD 32 I/O card is all about.

B.1.1. COMMONLY ASKED QUESTIONS

Q. Is my favorite STD I/O card compatible with the STD 32 specification?

A. Yes, as long as the card was designed to meet the mechanical and timing requirements of the STD-80 specification. STD 32 compliance can be met at different pre-defined levels. Since all STD-80 I/O boards, defined as Standard Architecture (SA) within the STD 32 specification, are 8-bit designs, they generally will be classified as:

STD 32 Compliance
I/O Slave: SA8-[I],[ICA],[SDMA]

Where [ ] defines the following options:

I: Interrupt request capability on INTRQ*, INTRQ1*, NMIRQ*, CNTRL (INTRQ2*), or INTRQ3*

ICA: Cascade interrupt controller

SDMA8: 8-bit SA frontplane DMA capability as specified within the STD 32 specification

The simplest SA I/O (STD-80) board would be specified as:

STD 32 Compliance
I/O Slave: SA8

See Chapter 4 of the STD 32 specification for more information on compliance levels.

STD 32 allows for 8-bit address decoders (the general requirement is that boards fully decode all 16 bits of I/O address) by requiring software to address these boards at locations FC00h-FFFFh, where IOEXP will be driven low. These boards must use IOEXP in their decode logic.

Manufacturers who have designed boards to be STD-80 compatible will immediately be able to claim STD 32 compliance!

Q. How can I improve the capabilities of my existing I/O product line by supporting STD 32 features?

A. New products or new revisions of existing products can use STD 32 connector land patterns to improve reliability and reduce connector wear.

The most direct improvement to many boards is the ability to drive IRQx. The STD bus has always been limited in its interrupt channels. STD 32 greatly increases the capabilities of the STD bus by providing a dedicated interrupt for each STD 32 slot. This typically requires the addition of a trace to drive backplane pin E47 with a positive true signal (for positive edge-triggered interrupt controllers) or a low-true signal for level-sensing controllers. In addition, INTRQ3* is a bussed interrupt that has been added to relieve backplane interrupt congestion.

Another capability introduced with STD 32 is the concept of defined DMA transfers on the backplane. Previously, designers tended to customize each DMA implementation that did not have the backbone of a specification to guarantee compatibility. STD 32 now gives a reference point from which to add DMA capabilities to designs.

Q. What are the benefits of using the Extended Architecture transfers of STD 32?

A. Performance. The systems of the 1990s demand fast, simple I/O and faster DMA types to meet their full potential. Designing for tomorrow's technology is becoming increasingly necessary. STD 32 gives the option of tapping into this performance. The STD-32 systems of the year 2000 will be mature 32-bit systems that still support the 8-bit board of 1980.

The move toward chipsets to reduce part density allows Extended Architecture to be implemented with off-the-shelf solutions from the PC marketplace. This results in greater compatibility at both the software and the hardware level.

APPENDIX B

B.2 DESIGNING FOR STD 32

Designing a card for STD 32 compatibility can mean different things, depending on the capabilities of the board being designed. To allow for this, the STD 32 bus specification defines different levels of compliance. An STD 32 compatible card can be as simple as an 8-bit card designed for STD-80 or as complex as a complete i486 master processor. This guide is intended to help designers of new I/O cards support both STD-80 Standard Architecture (SA) and Extended Architecture (EA) transfer types. Designers will see the benefit of modifying older designs to incorporate the new features such as slot-specific interrupts, DMA, and the additional ground and power pins that allow better power distribution.

Although this guide cannot be comprehensive, fundamental building blocks will be presented to help you implement your specific design. Although we will target I/O peripheral designs, memory designs are an extension of the principles presented here.

Here is a brief review of some STD 32 terminology and transfer types.

B.2.1. STANDARD ARCHITECTURE (SA) TRANSFERS

Standard Architecture (SA) transfers reflect older style STD bus transfer types, commonly defined by the STD-80 specification. STD-80 defines only 8-bit transfers using simple control lines and multiplexed address and data. I/O and memory are differentiated by the MEMRQ* and IORQ* signals. The direction of the transfer is defined by status signals STATUS0* and STATUS1*, and by RD* and WR*. The maximum transfer rate is 4 clock cycles at 8 MHz, or 2 Mbytes/second, if unrestricted access to the bus is given during a memory move. The arbitration scheme is limited because of its simplicity. Generally, bus topologies are limited to one system master and one temporary master, such as a DMA controller, and as many slaves as will fit into the backplane.

Multiplexing of data and address allows the STD-80 bus to have up to 24 bits of address instead of the original 16, but it also introduces a set of problems related to contention and ground bounce, if individual boards are not designed properly. These limitations led to the creation of STD 32, which defines a growth path for high performance designs, yet remains compatible with the core of STD: 8-bit I/O designs.

B.2.2. EXTENDED ARCHITECTURE (EA) TRANSFERS

Placing high performance processors on the congested STD-80 bus can seriously cripple their capabilities. STD 32 introduces an alternative transfer style to supplement STD-80 transfers in the form of Extended Architecture (EA) transfers. STD 32 still maintains compatibility with STD-80 cards.

Extended Architecture transfers are derived from the Extended Industry Standard Architecture (EISA) transfer types. EISA defines new transfer types to supplement ISA systems (such as the IBM AT). STD 32 adds these advanced transfer types to the STD bus. In this way, de facto silicon support for the EA transfers is available for STD 32. The important EISA features, such as speed, geographic addressing, and 32-bit address and data are maintained. Through the use of compliance level specification, STD 32 allows future designs to conform completely to the EA architecture.

EA transfers use non-multiplexed address and data busses to transfer data, which allows pipelining an address for the next cycle with the current cycle. The transfer is defined as follows:

M-IO specifies memory or I/O
W-R specifies direction
BE0*-BE3*, A2-A15, XA16-XA23, and XA24*-XA31* specify location
D0-D31 transfers the data

The size of the transfer is controlled by the signals EX32*, EX16* and EX8*, which are returned by the slave. START* initiates the transfer; CMD* defines the data transfer window and is extended by the master until the slave returns the signal EXRDY asserted. Boards that support EA transfers drive the signals EX8*, EX16*, or EX32* to the master in order to define the type and size of access that the master must perform. EX8*, EX16*, and EX32* driven by the slave indicate that the address is not latched and that the slave can handle the EA transfer speed. The timings for EX8* are consistent with EX16* and EX32*.

The fundamental cycle is a two cycle transfer, but BURST DMA style transfers (1 cycle) are available for a 32 Mbyte/second high end rate (at 8 MHz). Normal two cycle 32- bit data transfers allow 16 Mbytes/second. As indicated, the transfer rates attainable with STD 32 significantly increase the viability of the STD bus as a solution for speed-intensive applications. From an I/O application standpoint, interfacing with EA transfers is very simple. In fact, the most demanding part of EA transfer interfacing is managing 32-bit, 16-bit, and 8-bit cycles dynamically.

EA and SA Transfer Support

It is possible to implement a design that can handle both EA and SA transfer cycles. As systems that support EA transfers become readily available, these systems will need faster and wider (16- or 32-bit) transfers to avoid I/O bottlenecks at the system level. You will be able to roll I/O designs into these higher performance systems by following the guidelines of this designer's guide. Not every design requires this dual transfer capability. The design concepts presented here will allow you to make that decision from a more informed perspective.

B.2.3. INTERRUPT GENERATION

Devices that need interrupt support can drive several different interrupt signals. Each board can drive one of three bussed backplane signals existing on the STD-80 backplane and the bussed signal INTRQ3* defined by STD 32. STD 32 defines a slot-specific interrupt for each board. This signal allows the I/O device to have its own interrupt level. This signal (IRQx) is not bussed; the level is defined by the permanent master. In this example, the I/O device drives its interrupt request to any of the backplane interrupts via a jumper selection.

Bussed interrupt capability is signified with the compliance descriptor I. For example:

STD 32 Compliance
I/O Slave: SA8-I

Slot-specific interrupts are designated with the compliance descriptor IX. For example, if the above board also supported slot-specific (positive edge-triggered) interrupts, it would have the following product descriptor:

STD 32 Compliance
I/O Slave: SA8-I, IXP

B.2.4. DIRECT MEMORY ACCESS INTERFACING

EA Direct Memory Access (DMA) is supported within STD 32 by the backplane signals DREQx*, DAKx*, DMAIOW*, DMAIOR*, and T-C. Additional data bus management is needed during DMA transfers. Typically, DMAIOR* and DMAIOW* would be qualified by DAKx* to control direction and enabling of the buffer(s). SA DMA transfers are supported via frontplane cabling and are defined in Chapter 2 of the STD 32 specification.

SA DMA Compliance Designation

I/O boards that meet STD 32 definitions for frontplane DMA transfers may state that this capability exists with the SDMA8 or SDMA16 designator within the product descriptor. For example, if the board described in the previous example, which supports slot-specific and bussed interrupts, also supports 8-bit SA DMA, it would have the following product descriptor:

STD 32 Compliance
I/O Slave: SA8-I, IX, SDMA8

EA DMA Compliance Designation

Boards that support the different EA DMA modes have the additional designators EBURST, EDMAA, EDMAB, and/or EDMAC. These specify varying levels of performance and would be included in the product descriptor if supported.

B.2.5. MEMORY DESIGN DIFFERENCES

Memory design is more detailed than I/O design because a larger address is needed to define a specific cell within the 32-bit range. EA transfers no longer have address multiplexed with data, which allows pipelined transfers at the expense of extra buffers. Burst mode DMA transfers therefore allow 32 Mbyte/sec transfers at 8 MHz.

EA memory design implementation is similar to EA I/O design implementation. The main difference is that 32 bits of address are needed instead of 16. Buffer management for SA and EA transfer implementation becomes more detailed because SA transfers get A16-A23 from different pins than do EA transfers. I/O designs do not have this complication since SA I/O transfers and EA I/O transfers share the lower 16 address bits.

APPENDIX B

B.3 EXAMPLE IMPLEMENTATION

This example shows how a single I/O board interface can be used for the STD-80 bus and the STD 32 bus (SA or EA). This design approach guarantees the board will be useful with all future STD 32 systems while maintaining current STD-80 compatibility.

B.3.1. DESIGN FEATURES

The example design is partitioned into two logical sections: the front end logic and the application. The design builds on the front end logic features, which are shown in schematic 1 at the end of the STD 32 specification. An example application using this logic is illustrated in schematic 2 at the end of the specification.

Front End Logic

The front end logic presented for the example assumes the application supports STD-80, SA16, SA8, EA16, and EA8 I/O transfers. There are three parts to the front end logic:

1. Control Logic - Buffer management, wait state insertion, local read/write generation
2. Decode Logic - The two 74ALS520s and jumpers
3. Data Buffers/Address Latches - System bus and swap buffers, local address latch

The front end logic interfaces with the internal logic, which defines the board's application. This designer's guide spotlights an I/O application, but memory designs are a variation on this theme.

An I/O device typically needs a chip select, read and write, latched address, and data. Relevant control signals are available to the user from the control logic.

DMA Logic

The front end logic in the example releases buffer management and command control when the signal LOCAL_DACK* is asserted. The following signals are three-stated when LOCAL_DACK* is asserted:

WR_EVEN*
RD_EVEN*
WR_ODD*
RD_ODD*
DATA_DIR
HIDATAOE*
LODATAOE*
SWAPOE*

This allows the application to manage DMA transfers.

Application Example

The example given here has an 8-bit I/O device and a 16-bit read/write register that also drives a set of LEDs. The 8-bit I/O device is fully decoded on even and odd bytes so that it occupies a contiguous range of locations in the I/O map. For simplicity, the function of the device is not discussed. Fundamentally, it is an eight register I/O device that generates interrupts and requires DMA support. A 16-bit register can be read and written as 16 bits or as even and odd 8-bit values. The register is a pair of '996s. The LEDs on the outputs of the '996s help with proof of concept. This circuit is shown in schematic 2 at the end of the STD 32 specification.

B.3.2. SIGNAL DISCUSSION

The interface signals for the example design front end are discussed in detail in this section. Actual equations appear at the end of the designer's guide.

Board Select Decoding

(BD_SEL*, EQ*, AENx*, IORQ*, M-IO)

BD_SEL* is decoded in different ways, depending on the configuration.

STD-80 or SA transfers — For SA mode transfers, A0-A15 should be decoded (A0-A7 optionally) to define an I/O map region of up to 256 bytes for the board. This example assumes 16 internal registers (eight for the device and four redundant locations for the 16-bit register). The 74F520 output is qualified by IORQ* to produce BD_SEL*.

EA transfers — EA transfers are decoded by sampling the address bus (A0-A15), M-IO, START*, CMD*, and W-R. I/O designs need only decode A0-A15, whereas memory designs must decode A0-A15, XA16-XA23, and XA24*-XA31*. In the design example, BD_SEL* is latched by START* based on EQ*, M-IO, and AENx*. The internal control signals are generated from W-R based on BD_SEL* being true during CMD*.

Slot-Specific Consideration — STD 32 allows slot-specific addressing by driving the AENx* signal separately for each slot; each slot gets its own AEN* signal. The Permanent Master drives this signal in the I/O address ranges 0z000h-0z0FFh, 0z400h-0z4FFh, 0z800h-0z8FFh, and 0zC00h-0zCFFh. Peripheral boards use the AENx* signal to allow configuration by the Permanent Master. A typical board that allows slot-specific addressing might decode A0-A9=000000000b, and AENx* true for the slot-specific feature. Slot-specific addressing support is specified in the STD 32 compliance by the GAX designator. Both the master and peripheral must be GAX compliant for the above mechanism to be in place.

For support by masters without AENx* capability, all 16 bits of address must be decoded and the board should be mapped in one of the ranges 0y100h-0y3FFh, 0y500h-0y7FFh, 0y900h-0yBFFh, where y is any address. This is done by removing W38 in the application example.

In conclusion, the designer has three choices

1. For SA transfers, BD_SEL* should be determined by decoding down from A15 to the total number of ports that the I/O application needs. The example uses 16 locations, so A4-A15 are used to select BD_SEL*, as well as IORQ*.

2. For slot-specific EA transfers, A12-A15 are ignored, A8 and A9 must be decoded low, and the rest of the address bits are decoded appropriately. These designs can be mapped only at 0z000h-0z0FFh, 0z400h-0z4FFh, 0z800h-0z8FFh, and 0zC00h-0zCFFh, with "z" defining the slot. The driver software has to determine the location of the board by selectively looking at each slot until the device is found. The software then appends A12-A15 to the rest of the address for further accesses, and the master hardware drives AENx* appropriately. AENx* may be ignored for masters that do not have AENx* support.

3. For PC style peripherals that are not slot-specific (i.e., the example), all address bits must be decoded down to the size of the board. PC peripherals are generally located between 100h-3FFh or an alias of that address.

Internal Address Bus and Control

(IAB) and (ADR_LE)

The Internal Address Bus (IAB) is buffered from the backplane by a 74ALS573 latch. A0-A7 is latched in EA mode. This latch is transparent in SA mode. The IAB should not be used to derive the signals EX8*, EX16*, EX32*, IO16*, or DEVICE16*. IAB0 and IAB1 are decoded from BE0-3* when in EA mode.

Internal Data Bus and Control

(IDB) and (LODATAOE*, HIDATAOE*, SWAPOE*, DATA_DIR, DEVICE16*,BP16*)

The Internal Data Bus (IDB) consists of the even (D0-D7) and odd (D8-D15) busses. For 8-bit accesses to the board, all data is driven on the backplane D0-D7 signals. 16-bit accesses have even and odd data driven on their respective data busses. In an 8-bit backplane, a 16-bit device must have its odd address data gated to the D0-D7 bus for proper operation. The signal DEVICE16* is an input from the application that defines how the buffers are to be utilized. Between DEVICE16* and BP16*, the control logic has enough information to coordinate the correct byte-laning. BP16* has meaning only in SA transfers. BP16* must be high (W16 not installed) for STD-80 backplanes. The signal DEVICE16* must be dynamically driven if both 8- and 16-bit devices are present on the board. DEVICE16* can be statically pulled high if the application devices are all 8-bit or statically pulled low for 16-bit devices.

APPENDIX B

8-bit devices

SA MODE — 8-bit I/O devices should reside on the D0-D7 bus and not drive the signal IO16*. This will tell the host CPU to break 16-bit accesses into two 8-bit accesses. Older STD-80 boards (which are 8-bit data line devices by definition) do not drive this signal, allowing these boards to work with newer STD 32 host CPUs. These host CPUs have a pullup on IO16* and will access all cards as 8-bit devices unless IO16* is driven.

EA MODE — EA 8-bit transfers are similar to SA 8-bit devices except that the EX8* signal must be driven back to the master by the rising edge of the clock generating START*. This informs the master that the cycle is to be completed as an EA transfer.

16-bit devices

SA MODE — 16-bit devices will attach to both D0-D7 and D8-D15. When decoded, the application logic will drive IO16* low, telling the host CPU that a 16-bit transfer is legitimate. The CPU then has the option of driving all 16 bits of data on the bus, along with the signal BHE*, which is used by the I/O board to enable the data internally. The CPU may also access the device in byte-wide fashion. In this case, D0-D7 of the IDB is enabled to the backplane signals D0-D7 during even accesses. D8-D15 of the IDB is likewise enabled to D8-D15 during odd 8-bit accesses. If IO16* is not returned by the slave to a 16-bit master, it is assumed that BHE* is not significant to the I/O Slave. Some chip sets drive BHE* low during both halves of such a downshift transfer.

In 8-bit backplanes or with 8-bit CPUs, the signal BP16* should be unstrapped (pulled high) to allow 16-bit devices to have their data swapped down to D0-D7 on odd byte accesses.

EA MODE — 16-bit accesses under EA mode are very similar to EA8 transfers, except that the signals EX16* (instead of EX8*) and DEVICE16* are asserted. DEVICE16* is used by the local buffer control logic to define buffer management during odd byte accesses. EX16* and EX8* are used by the master logic (typically the CPU) to define the type and size of transfer to be performed.

Command Signals

(WR_EVEN*, RD_EVEN*, WR_ODD*, RD_ODD*)

The front end logic supplies individual read and write signals based on address for 16-bit devices. 16-bit application devices that have common read and write pins for both addresses will have to OR the even with the odd command signals. Depending on the application, common read and write signals may be best.

SA MODE — The command signals are derived from RD*, WR*, IORQ*, address, and BHE*, and follow standard STD bus conventions.

EA MODE — The command signals are derived from CMD*, W_RL, and BD_SEL*. CMD* is the final qualifier that defines the duration of the command.

WAITRQ* and EXRDY

The option is given to insert one wait state per transfer in either SA or EA mode. The application may insert additional wait states by negating the signal READY prior to the rising edge of CLOCK*. Ready must be steady state with enough setup for the speed device being used.

SA MODE — WAITRQ* will be asserted during T2 if jumper W25 is inserted. This will give one wait state unless READY is negated prior to T3. The application may then insert as many wait states as necessary by keeping READY negated. WAITRQ* will be driven high one clock after READY is driven high.

EA MODE — EXRDY will be negated during the first half of the first cycle of CMD* if W26 is inserted. This will insure one wait state. The application may insert additional wait states by negating READY prior to the rising edge of the next cycle and keeping READY negated for the required number of cycles. The control logic will gate EXRDY with the first half of every CLOCK* cycle that READY is asserted. READY should only transition during the last half of a CLOCK* cycle.

B.3.3. APPLICATION OPERATION

The example illustrates buffer management and transfer size management, which are necessary in dynamic bus sizing systems such as STD 32. The master (typically the CPU) and the slave I/O device have several responsibilities. The CPU initiates a cycle by driving an address and SA and/or EA command signals on the bus. The slave responds with the type of transfer that is needed. The following demonstrates how the front end logic reacts to an on-board access.

EA Transfer

(EAMODE Jumper = In)

1. The CPU presents the I/O address and control signals.
2. The slave responds with EX8* or EX16*, when appropriate. EX16* is driven only for accesses to the 16-bit register. DEVICE16* is driven to buffer control during 16-bit accesses.
3. The IAB is latched with START* for cycle usage.
4. The local command signals are asserted based on CMD* and W_R, etc.
5. EXRDY, if enabled, is negated for one wait state.
6. Additional wait states may be asserted by the application by negating the READY signal.
7. Data is gated according to the transfer size and device size.
8. The cycle terminates when CMD* is removed.

SA Transfer

(EAMODE Jumper = Out), STD-80 Transfer (BP16 Jumper = Out)

1. The slave responds with IO16* when appropriate. IO16* is driven only for accesses to the 16-bit register. DEVICE16* is driven to buffer control during 16-bit accesses.
2. The IAB is transparent to the STD 32 bus throughout the cycle.
3. The local command signals are asserted based on RD* or WR*.
4. WAITRQ*, if enabled, is asserted for one wait state.
5. Additional wait states may be asserted by the application by negating the READY signal.
6. Data is gated according to transfer size and device size.
7. The cycle terminates when RD* or WR* are removed.

B.3.4. APPLICATION SIGNAL DISCUSSION

Chip Select Decoding

(CSDEVICE*, LEDEN*)

Chip selects for a particular I/O device can usually be derived from the board select signal and the Internal Address Bus (IAB). Depending on the I/O device, the number of address lines looked at will vary. In this example, the I/O device has eight internal read/write registers. The board select (BD_SEL) signal is derived from A4-A15.

CSDEVICE* is decoded by qualifying BD_SEL with A3 of the IAB equal to 0. This will drive CSDEVICE whenever the I/O address 0500h-0507h is presented by the CPU. The 16-bit register is enabled with LEDEN* in a similar fashion when 0508h-050Fh is accessed by the host CPU. Note that the register will be redundantly mapped within this range. When the 16-bit register is accessed, the signal EX16* is asserted for the master, and DEVICE16* is asserted for the local control logic. DEVICE16* is driven throughout the cycle. EX16* and EX8* are driven only while EQ*, AENx*, A3, and MIO are asserted.

DMA Command Signals

(DRD*, DWR*)

DRD* and DWR* are driven when the LOCAL_DACK* signal is active. The signals WR_EVEN* and RD_EVEN* will be three-stated at this time, allowing the application to control them.

DMA Buffer Control Signals

(DATA_DIR, LODATAOE*)

The backplane interface logic does not drive the buffer control signals during DMA cycles. The application logic drives these signals for the DMA transfer when LOCAL_DACK* is active.

Data Bus Sizing Signals

(EX8, EX16*, IO16*, DEVICE16*)

Accesses to the board can be 8-bit or 16-bit. The application logic returns the proper signal to the host processor by the rising clock edge that takes away START* in EA mode transfers. For SA mode transfers, IO16* is returned throughout the cycle. DEVICE16* is driven to the buffer manager (U7) to indicate the type of transfer for which to gate data.

B.4 CONCLUSIONS

This example is not meant to solve all design considerations for all modes of operation, but rather to provide a stepping stone for a specific application. Some applications will be more complicated, and some will be less complicated. With the implementation shown above and with the equations given on the following pages, designers should be able to implement systems that support SA and EA I/O transfer types.

The prime benefit of the EA transfers is speed. 32-bit I/O transfers are an extension of the themes presented in this designer's guide. Other important considerations for EA capabilities include the availability of much higher performance DMA transfers, expanded interrupt capabilities, and a clearly mapped growth path for the future.

APPENDIX B

```
                          module std32a13
        TITLE           'STD 32 APPLICATION IODEVICE Logic
        ENGINEER
        COMPANY         STD 32 SIG
        PROJECT         STD 32 Designers Guide
        REVISION        1.3
        COMMENT         IODEVICE Implementation for STD 32 APPX B'
@onset
std32a13 DEVICE 'P22v10';
VCC             MACRO {1};
ACTIVE          MACRO {1};
INACTIVE        MACRO {0};
ON              MACRO {1};
OFF             MACRO {0};
A               MACRO {1};
I               MACRO {0};
TRANSPARENT     MACRO {1};
LATCHED         MACRO {0};
Z               = .Z.;
X               = .X.;

!bd_sel          PIN 1;
!eq              PIN 2;
!eamode          PIN 3;
 a3              PIN 4;
 iab3            PIN 5;
 mio             PIN 6;
!laenx           PIN 7;
!local_dack      PIN 8;
!dmaior          PIN 9;
!dmaiow          PIN 10;
!dmar            PIN 11;
!dmaw            PIN 13;

!device16        PIN 14 ISTYPE 'com';
!ex16            PIN 15 ISTYPE 'com';
!io16            PIN 16 ISTYPE 'com';
!leden           PIN 17 ISTYPE 'com';
!csdevice        PIN 18 ISTYPE 'com';
!lo_data_oe      PIN 19 ISTYPE 'com';
 data_dir        PIN 20 ISTYPE 'com';
!dwr             PIN 21 ISTYPE 'com';
!drd             PIN 22 ISTYPE 'com';
!ex8             PIN 23 ISTYPE 'com';

EQUATIONS

ex16.oe = eamode & !mio & eq & laenx & a3;
ex8.oe = eamode & !mio & eq & laenx & !a3;
io16.oe = bd_sel & iab3;


data_dir.oe = local_dack;

ex8        = ON;
ex16       = ON;
io16       = ON;
device16   =    eamode & !mio & eq & a3
             #  eamode & device16 & bd_sel     "latch until end of cycle
             # !eamode & bd_sel & a3;
leden      =    bd_sel & iab3;                 "xxx8h - xxxfh

csdevice   =    bd_sel & !iab3 & !local_dack;  "xxx0h - xxx7h

" dma equations

dwr        =    eamode & dmaiow
             # !eamode & dmaw;
drd        =    eamode & dmaior
             # !eamode & dmar;
lo_data_oe =    eamode & (dmaiow # dmaior)
             # !eamode & (dmaw # dmar)
```

APPENDIX B

```
             #  lo_data_oe.pin & local_dack;   "extra hold time
data_dir   =    eamode & !dmaior
             # !eamode & !dmar;

                           END std32a13
```

APPENDIX B

```
module std32b13
        TITLE           'STD 32 Designers Guide Front End Logic
        ENGINEER
        COMPANY         STD 32 SIG
        PROJECT         STD 32 Designers Guiide
        REVISION        1.3
        COMMENT         STD 32 Backplane Interface'
@onset

std32b13 DEVICE 'MACH120A';

VCC             MACRO {1};
ACTIVE          MACRO {1};
INACTIVE        MACRO {0};
ON              MACRO {1};
OFF             MACRO {0};
A               MACRO {1};
I               MACRO {0};
TRANSPARENT     MACRO {1};
LATCHED         MACRO {0};
Z               = .Z.;
X               = .X.;

!start          PIN 49;
!aenx           PIN 21;
!eq             PIN 50;
!cmd            PIN 10;
!sysreset       PIN 20;
!rd             PIN  9;
!wr             PIN  6;
!iorq           PIN 17;
!eamode         PIN 54;
!bp16           PIN  5;
!bhe            PIN  3;
!intak          PIN 37;
!local_dack     PIN 14;
!device16       PIN 13;
 be3            PIN 28;
 be2            PIN 30;
 be1            PIN 31;
 be0            PIN 33;
 a1             PIN 26;
 a0             PIN 51;
 mio            PIN 24;
 w_r            PIN 40;
 bclk           PIN 15;
 ready          PIN 16;

!bd_sel         PIN 36 ISTYPE 'com';
!exoe           PIN 38 ISTYPE 'com';
!ea_brd_add     NODE ISTYPE 'com';
 w_rl           PIN 39 ISTYPE 'com';
 adr_le         PIN 41 ISTYPE 'com';
 data_dir       PIN 22 ISTYPE 'com';
!lo_data_oe     PIN  7 ISTYPE 'com';
!hi_data_oe     PIN  4 ISTYPE 'com';
!swap_oe        PIN 23 ISTYPE 'com';
!bd_rd_even     PIN 25 ISTYPE 'com';
!bd_rd_odd      PIN 32 ISTYPE 'com';
!bd_wr_even     PIN 29 ISTYPE 'com';
!bd_wr_odd      PIN 11 ISTYPE 'com';
```

APPENDIX B

```
!exrdy          PIN 44 ISTYPE 'com';
!exrdy2         PIN 43 ISTYPE 'com';
 iab1           PIN 66 ISTYPE 'com';
 iab0           PIN 67 ISTYPE 'com';
!waitrq         PIN  2 ISTYPE 'reg_d,buffer';
!waitrq2        PIN 12 ISTYPE 'reg_d,buffer';
 lbe3           NODE istype 'com';
 lbe2           NODE istype 'com';
 lbe1           NODE istype 'com';
 lbe0           NODE istype 'com';
 byte_0         NODE istype 'com';
 byte_1         NODE istype 'com';

EQUATIONS
iab1 = (!eamode & a1
     # eamode & !lbe3 & lbe1 & lbe0
     # eamode & !lbe2 & lbe1 & lbe0);

iab0 = (eamode & !lbe3 & lbe2 & lbe1 & lbe0
     # eamode & lbe3 & !lbe1 & lbe0
     # eamode & !lbe2 & !lbe1 & lbe0
     # !eamode & a0);

lbe3 = be3 & start
     # lbe3 & !start
     # lbe3 & be3;

lbe2 = be2 & start
     # lbe2 & !start
     # lbe2 & be2;

lbe1 = be1 & start
     # lbe1 & !start
     # lbe1 & be1;

lbe0 = be0 & start
     # lbe0 & !start
     # lbe0 & be0;

byte_0 = (!lbe2 & lbe1 & lbe0
     # lbe3 & lbe2 & !lbe0
     # !lbe2 & !lbe1 & !lbe0);

byte_1 = (lbe3 & !lbe1
     # !lbe2 & !lbe1
     # !lbe3 & lbe1 & lbe0);

data_dir.oe = !local_dack;
lo_data_oe.oe = !local_dack;
hi_data_oe.oe = !local_dack;
swap_oe.oe = !local_dack;
bd_rd_even.oe = !local_dack;
bd_rd_odd.oe = !local_dack;
bd_wr_even.oe = !local_dack;
bd_wr_odd.oe = !local_dack;
exrdy.oe = exoe;
exrdy2.oe = exoe;
waitrq.clk = bclk;
waitrq2.clk = bclk;
waitrq.oe = !eamode & eq & iorq & waitrq & !sysreset & !intak;
waitrq2.oe = !eamode & eq & iorq & waitrq & !sysreset & !intak;
```

```
w_rl    = w_r & start
        # w_rl & !start
        # w_rl & w_rl;

adr_le    =  eamode & start
          # !eamode;

ea_brd_add = aenx & !mio & eq;

bd_sel    =  eamode & ea_brd_add & start & !sysreset
          #  eamode & bd_sel & !start & !sysreset
          #  eamode & bd_sel & ea_brd_add & !sysreset
          # !eamode & eq & iorq & !sysreset;

bd_rd_even=  eamode &  device16 & bd_sel & !w_rl & cmd & byte_0 & !sysreset
          #  eamode & !device16 & bd_sel & !w_rl & cmd & !sysreset
          # !eamode &  device16 & eq & iorq & rd & !a0 & !sysreset
          # !eamode & !device16 & eq & iorq & rd & !sysreset;

bd_wr_even=  eamode &  device16 & bd_sel & w_rl & cmd & byte_0 & !sysreset
          #  eamode & !device16 & bd_sel & w_rl & cmd & !sysreset
          # !eamode &  device16 & eq & iorq & wr & !a0 & !sysreset
          # !eamode & !device16 & eq & iorq & wr & !sysreset;

bd_wr_odd =  eamode &  bd_sel & w_rl & cmd & byte_1 & !sysreset
          # !eamode &  bp16 & eq & iorq & wr & bhe & !sysreset
          # !eamode & !bp16 & eq & iorq & wr & a0 & !sysreset;

bd_rd_odd =  eamode &  bd_sel & !w_rl & cmd & byte_1 & !sysreset
          # !eamode &  bp16 & eq & iorq & rd & bhe & !sysreset
          # !eamode & !bp16 & eq & iorq & rd & a0 & !sysreset;

data_dir  =  eamode & w_rl
          # !eamode & !rd & iorq;

lo_data_oe=
             (
                 eamode &  device16 & bd_sel & (w_rl # cmd) & byte_0
              #  eamode & !device16 & bd_sel & (w_rl # cmd)
              # !eamode &  bp16 & device16 & eq & iorq & (rd # wr) & !a0
              # !eamode & !device16 & eq & iorq & (rd # wr)
             ) & !sysreset;

hi_data_oe=
             (
                 eamode & device16 & bd_sel & (w_rl # cmd) & byte_1
              # !eamode & bp16 & device16 & eq & iorq &  (rd # wr) & bhe
             ) & !sysreset;

swap_oe   =
             (
               !eamode & !bp16 & device16 & eq & iorq & (rd # wr) & a0 & !local_dack
             # !eamode & !bp16 & device16 & (rd # wr) & a0 & local_dack
             ) & !sysreset;
             " the swap buffer is used for local 16-bit devices in 8-bit
             " only backplanes (STD-80 backplanes)

exrdy     =  ON;        " double up for increased current capability
exrdy2    =  ON;
```

APPENDIX B

```
exoe      =  (
              eamode & ea_brd_add & !bclk & start " at end of START cycle
           #  eamode & ea_brd_add & bclk & exoe   " 1 wait at 1st CMD clk
           #  eamode & ea_brd_add & bclk & !ready " more if ready asserted
              ) & !sysreset;


waitrq.d  =  (
             !eamode & !rd & !wr & !intak
           # !ready
              ) & !sysreset;
                 " gives 1 wait state # more if ready
                 " returned with setup to rising edge of bclk

waitrq2.d =  (
             !eamode & !rd & !wr & !intak
           # !ready
              ) & !sysreset;
                 " gives 1 wait state # more if ready
                 " returned with setup to rising edge of bclk
                 " waitrq2 gets paralleled with waitrq for 24 ma

                              END std32b13
```

A P P E N D I X C

HOT SWAP IMPLEMENTATION

APPENDIX C

C.1. INTRODUCTION

Hot swapping is the ability to insert and remove boards from a powered backplane without disturbing the system. This appendix describes in detail the implementation of Hot Swap on the STD 32 bus. Both hardware and software implementation are discussed.

C.2. HARDWARE IMPLEMENTATION

Several hardware issues must be considered when dealing with Hot Swapping. The first issue is a method to insert and remove a board without disrupting the signal or power quality on the backplane. The second issue is a method to notify the application software that a particular board is being swapped. The last issue is a method to reset and bring up the board after it is brought into the system. The sections to follow detail the implementation requirements to develop Hot Swap capability on an STD 32 system.

C.2.1 COMPATIBILITY

Hot Swap is a new feature added to the STD 32 product line. Any I/O or CPU card developed with Hot Swap capability will work only in the STD 32 backplane. It is not compatible with STD-80 backplanes. However, any number of standard STD-80 or STD 32 cards may reside in an STD 32 backplane without affecting the capability of inserting or removing a Hot Swap STD 32 card.

Unique software drivers must be developed to maximize the use of Hot Swap without user intervention.

C.2.2. LOGIC FAMILIES

The basic problem with inserting or removing a board into a live backplane is that most logic families, such as CMOS, ACL, ACT, and HCMOS, do not provide isolation when power is removed. These logic families contain P-channel transistors or diodes between the I/O pin and VCC. These transistors or diodes are forward biased when VCC is removed, causing the bus to be pulled to one diode drop above ground. The net result is that the bus is disturbed when the device is powered up or down. Some logic families, such as ABT and BCT, do not contain clamping diodes or transistors and guarantee that the input and output will be off when VCC is applied or removed.

Every signal that connects to the backplane must be received with an ABT-type buffer from National Semiconductor or equivalent to prevent glitching of the backplane logic upon insertion. The output enables must remain above the rising VCC level until VCC has ramped up to 4.75 V.

C.2.3. POWER SEQUENCING

To ensure that the ABT buffers do not enable during insertion or removal, the buffer output enables must reach a high level of >2.0 V before VCC is applied or maintain a >50% VCC level during VCC ramping. To ensure this condition a staggered finger pattern has been specified to provide power sequencing. The new finger pattern applies power so the ground pins connect first, followed by the signal pins, and last, the VCC pins. Power conditioning circuitry is also needed on the board for controlling current surges when the board is initially inserted.

A five level staggered finger pattern has been defined for this specification to provide for proper signal and power sequencing. The STD 32 extensions must also be used to guarantee proper alignment. Table C-1 provides descriptions and distances from the edge of the board for each signal group.

Table C-1. Backplane Finger Recess Definitions (Five Levels of Fingers).

Level	Card Edge Distance (Inches)	Description
Ground	0.000	Ground is the first level connected to the STD backplane.
VCC_BIAS	0.020	VCC_BIAS (pin E68) provides power to the internal control logic. It keeps the backplane buffers from glitching on power up. Logic using VCC_BIAS should be kept at a minimum to avoid glitching the power supply.
Signals	0.060	Backplane interface signals.
VCC_IN	0.080	VCC_IN to the PCB board. Input to power conditioning MOSFETs to provide onboard power.
BD_SEL#	0.160	Board Select (pin E54) enables power-up circuitry on the PCB board. The backplane is held in tri-state until this pin is connected and VCC reaches 4.75 V.

Detailed mechanical drawings and implementation details for the new finger pattern are given in Section C.2.4 , "Backplane Mechanical Specification."

C.2.4. BACKPLANE MECHANICAL SPECIFICATION

The card edge fingers for the STD 32 interface have been changed to accommodate the requirements for Hot Swap. Figures C-1 through C-6 show the actual dimensions of the new five-level finger layout.

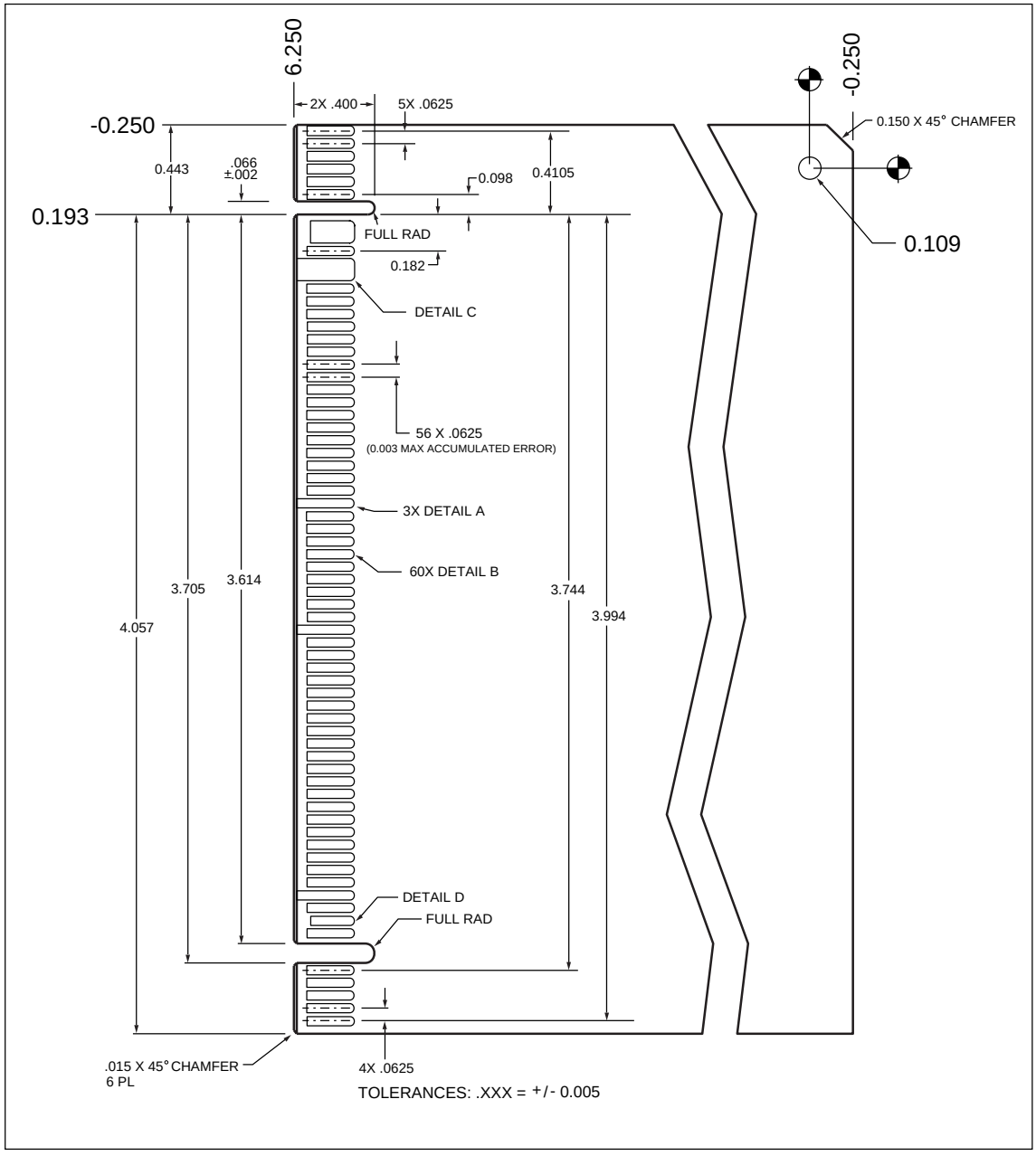


Figure C-1. Live Insertion STD 32 P/E Board Fab Drawing (Component Side).

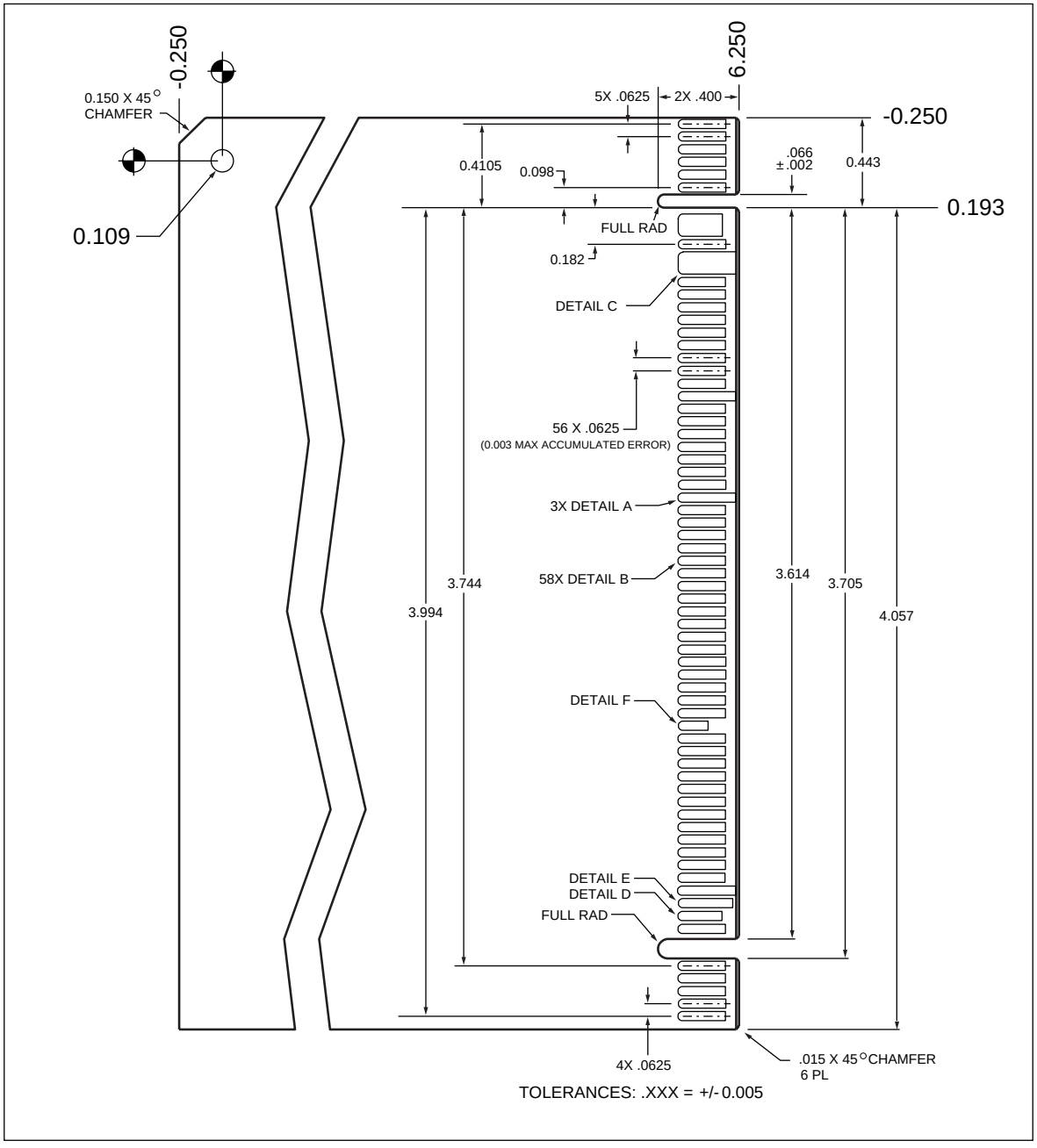


Figure C-2. Live Insertion STD 32 P/E Board Fab Drawing (Solder Side).

APPENDIX C

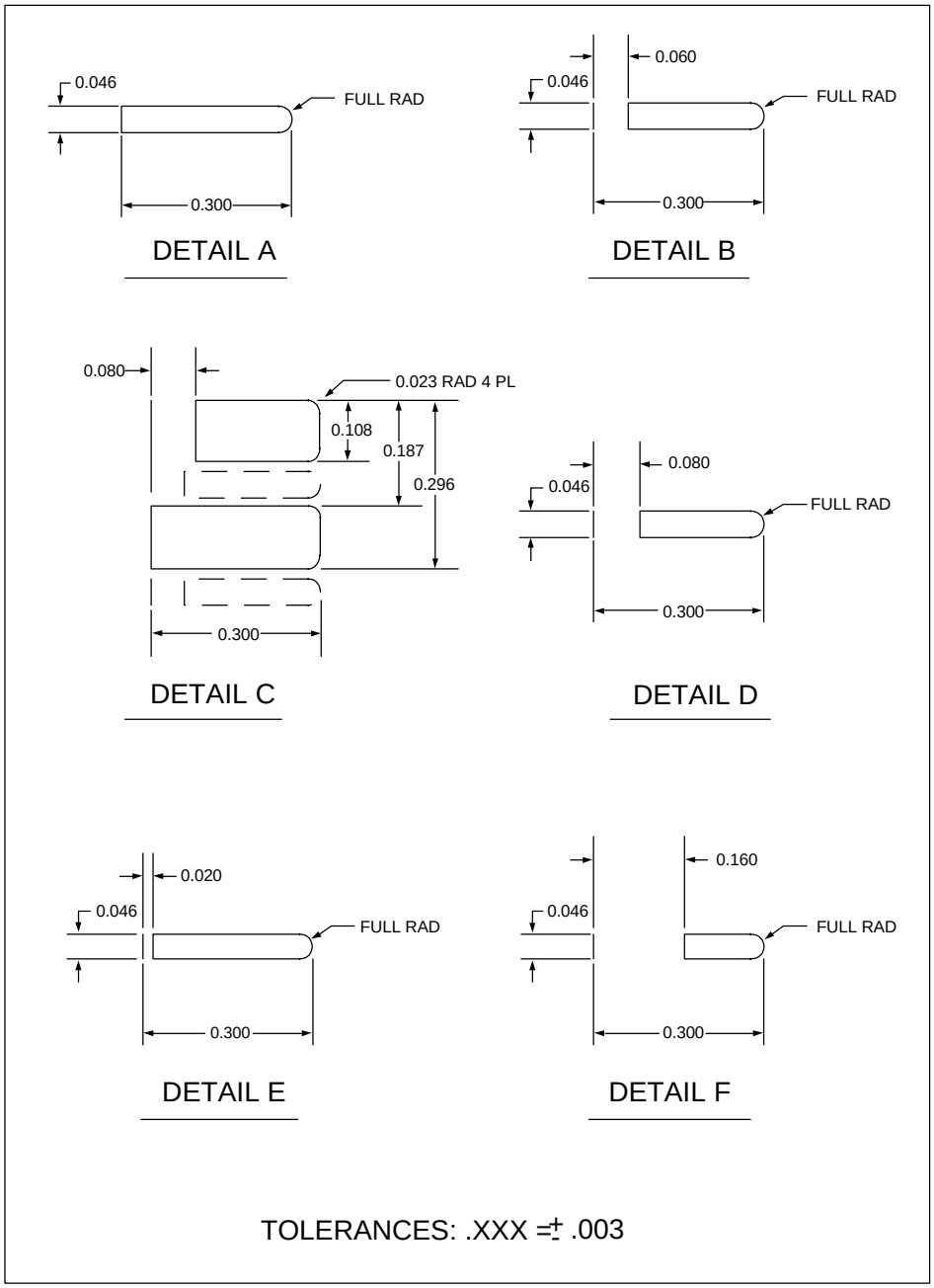


Figure C-3. STD 32 Live Insertion P/E Finger Dimensions.

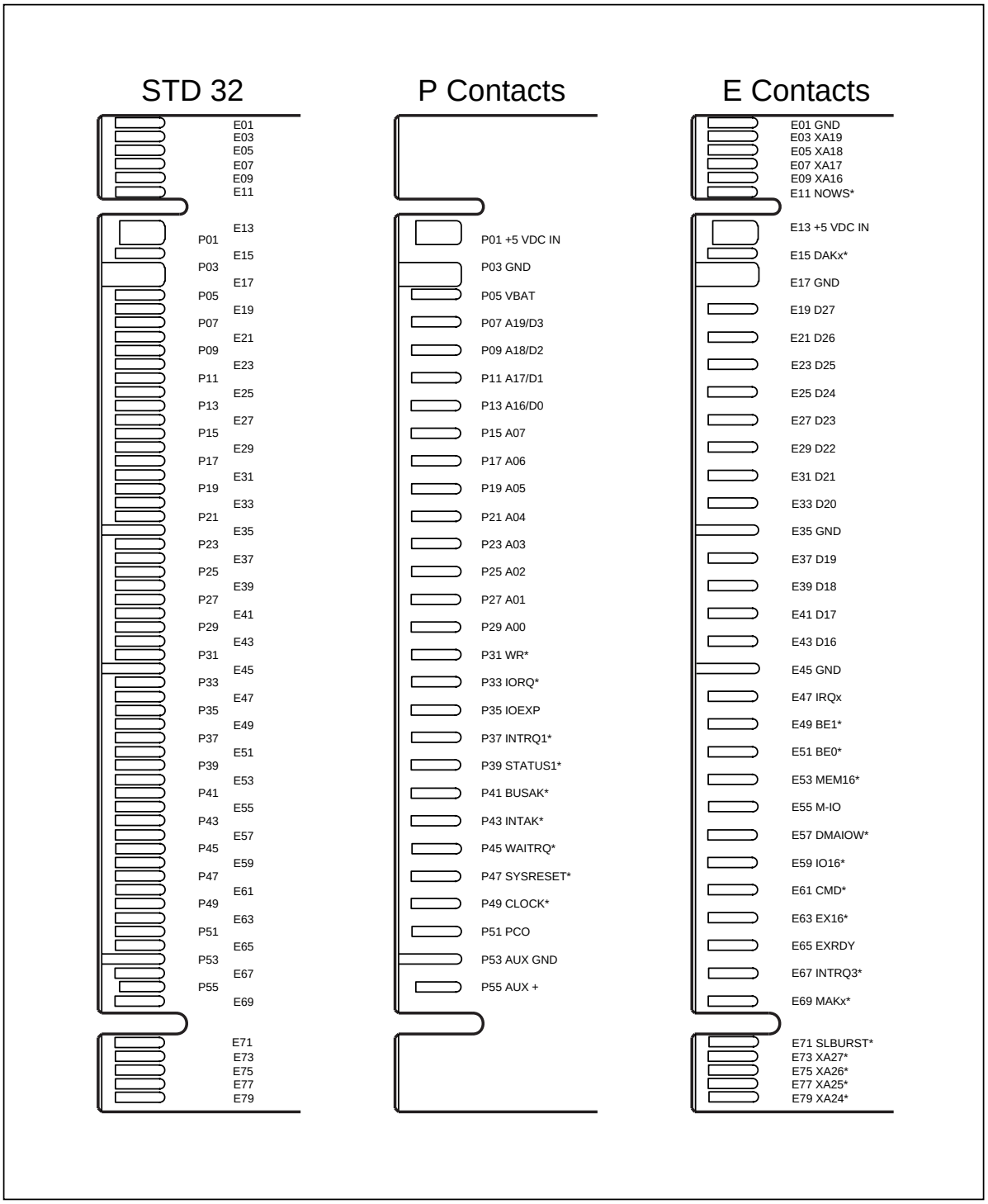


Figure C-4. STD 32 Live Insertion Pin Assignments (Component Side).

APPENDIX C

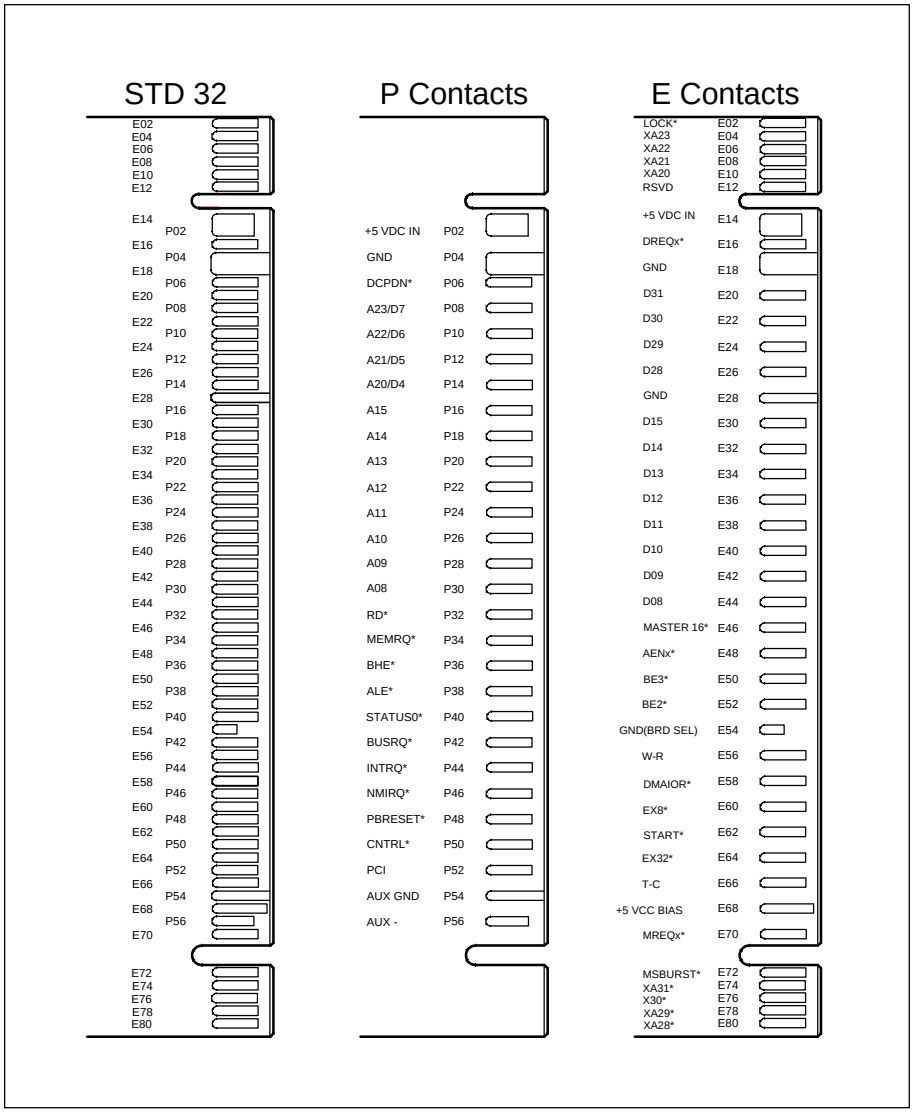


Figure C-5. STD 32 Live Insertion Pin Assignments (Solder Side).

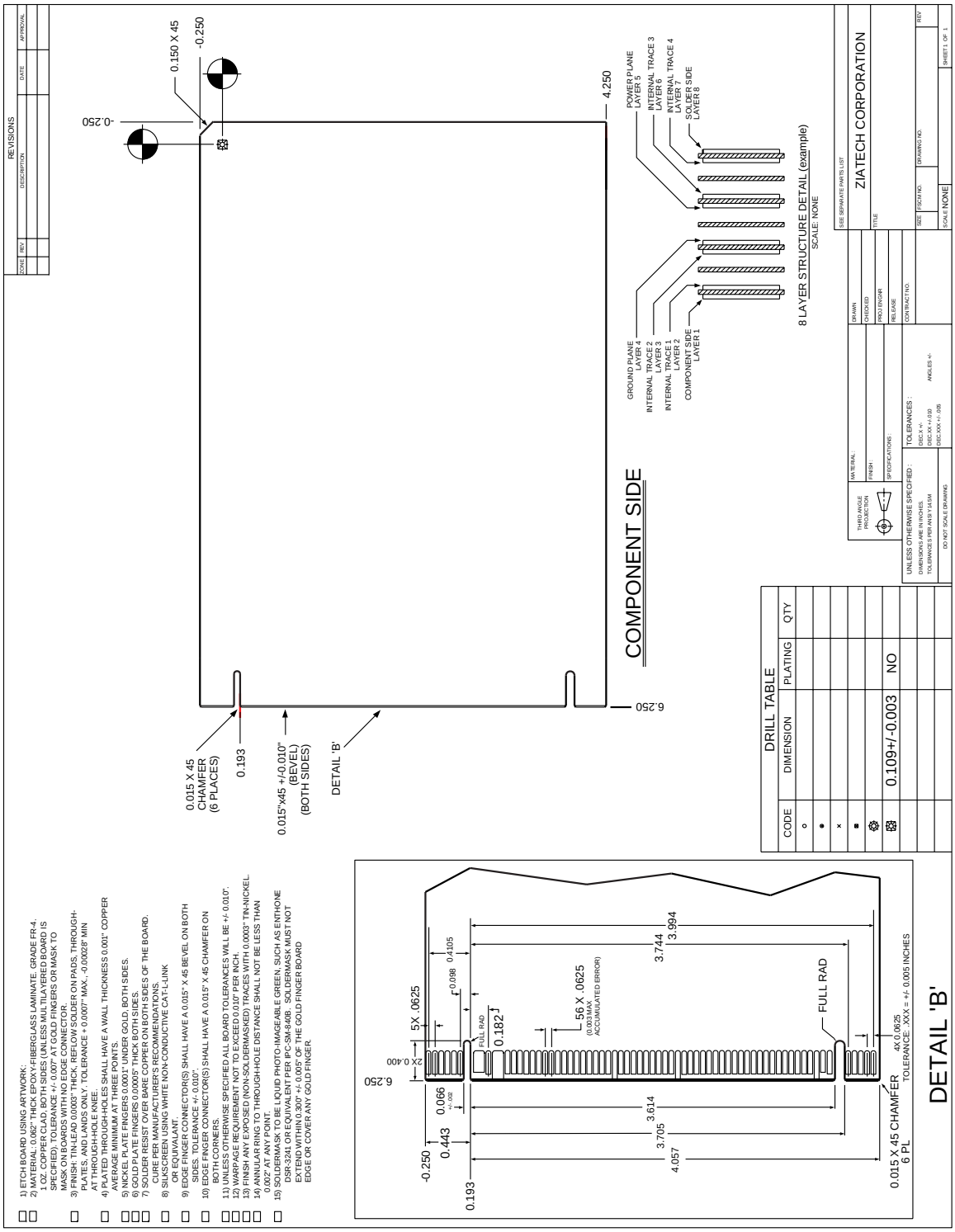


Figure C-6. Sample Hot Swap Fabrication Drawing.

APPENDIX C

C.2.5. POWER PLANE SECTIONING

Any board using the Hot Swap technology will have three power planes: VCC_BIAS, VCC_IN, and VCC. The VCC_BIAS plane gives power to all backplane control logic to prevent power-up glitching. VCC_IN is the input to the low on resistance MOSFET. VCC provides all the power to the remaining circuits on the I/O board.

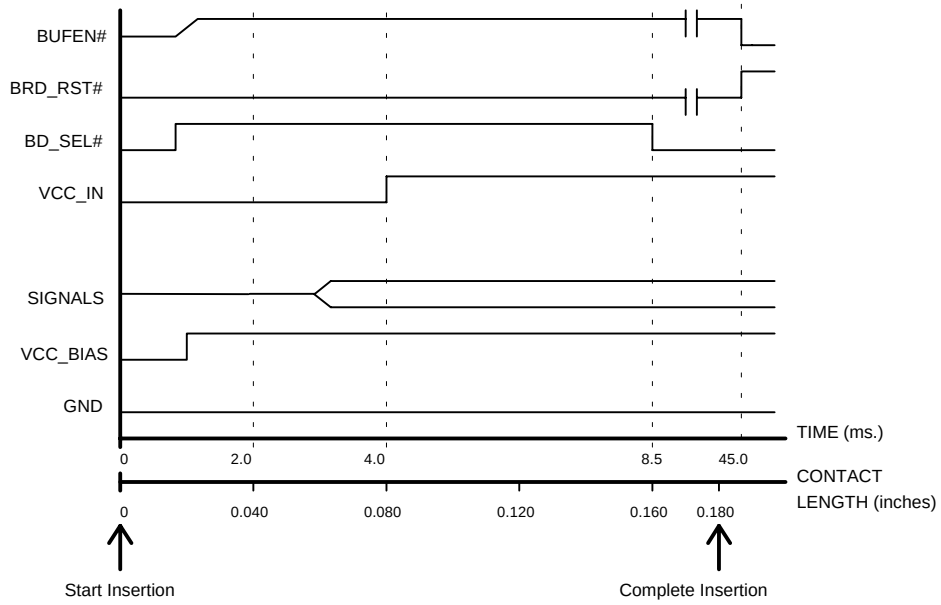
C.2.6. POWER LEAKAGE LIMITING

Power limiting is very important on all circuits powered from VCC_BIAS. Leakage current can cause damage to onboard circuits and potentially glitch the backplane power supply. Power leakage happens when part of the PAL or other logic is used to control the backplane logic and also control other onboard circuitry. To limit this problem all control circuit output pins must be tri-stated while VCC is below 4.75 V. The only signals that should be driven hard are GREEN_LED#, BRD_RST#, and POWER_CTL. All other signals should be in a tri-state condition until VCC has reached 4.75 V.

The output signals BUFEN0#, BUFEN1#, SWAPEN#, IO16#, DATA_DIR#, BUFIN_EN#, INTR#, and PWR_INTR should be pulled high to VCC_BIAS via a 1 k ohm pullup resistor.

C.2.7. CARD INSERTION TIMING

The following timing diagram shows the timing of the main control signals during board insertion.

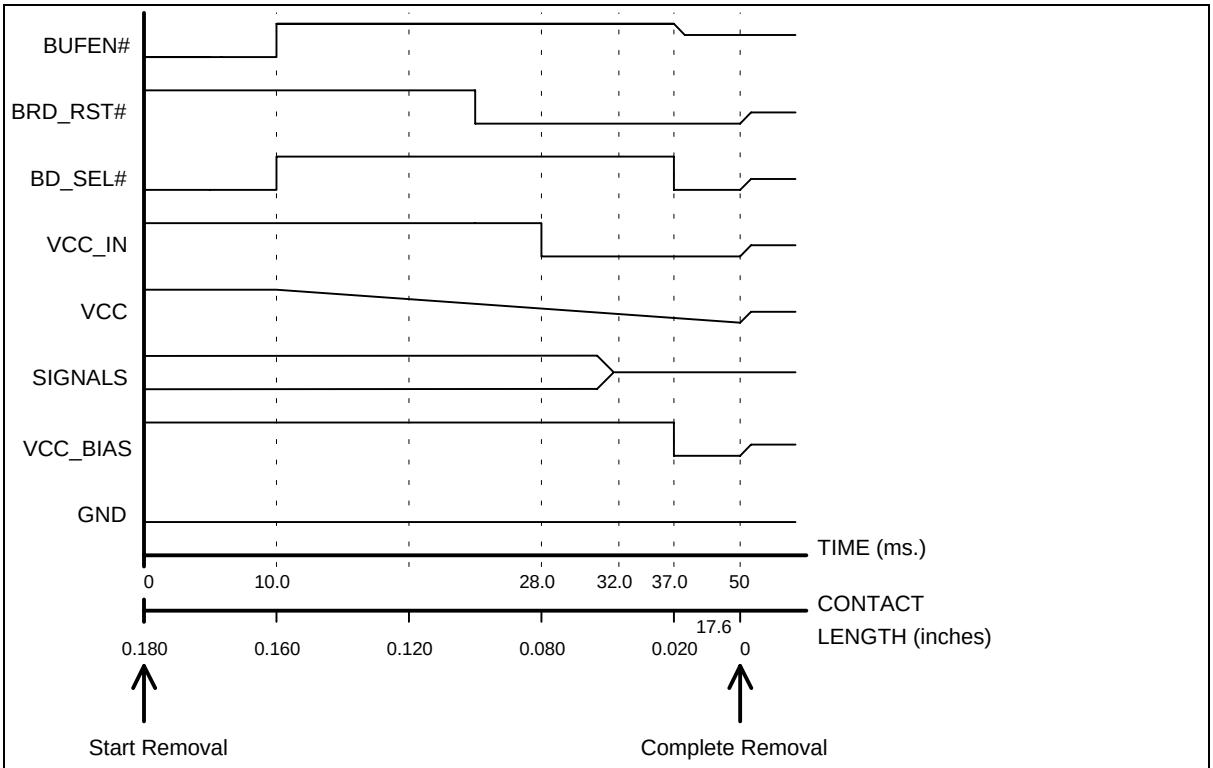


VCC_BIAS = power for board sense and ABT buffer control
SIGNALS = all of the STD backplane signals
VCC_IN = main board power input
BD_SEL# = signal for sensing when board is inserted or removed from backplane
BRD_RST# = board reset--self generating, ANDed with SYSRESET* on backplane
BUFEN# = buffer enable signal; this is tied to VCC_BIAS through an internal pullup on the PAL device
GND = board ground signal

Figure C-7. Card Insertion Diagram.

C.2.8. CARD REMOVAL TIMING

The following timing diagram shows the timing of the main control signals during board removal. This diagram assumes that the SWAP signal is used to initiate a board removal. However, note that the manual switch or software isolate bit can also initiate a removal sequence.



VCC_BIAS = power for board sense and ABTE buffer BIAS pin
SIGNALS = all of the STD backplane signals
VCC_IN = VCC from backplane to the MOSFET
VCC = main board power
BD_SEL# = signal for sensing when board is inserted or removed from backplane
BRD_RST# = board reset - self generating, ANDed with SYSRESET* on backplane
BUFEN# = buffer enable signal; this is tied to VCC_BIAS through an internal pullup on the PAL device
GND = board ground signal

Figure C-8. Card Removal Diagram.

C.2.9. POWER CONTROL LEDS

Power Control LEDs are not necessary to implement a Hot Swap board, but they are defined to maintain consistency across the Hot Swap product line.

There are two Power Control LEDs, as defined below. The LEDs should be adjacent to each other and located near the card ejector.

Green LED - This LED is turned on when the peripheral board has been inserted but the main board has not reached the minimum voltage to power the peripheral board. It turns off when the main power reaches the proper operating level.

Red LED - This LED is turned on only when the peripheral board has been inserted and the main power has reached the proper operating level to power the peripheral board.

Table C-2. LED Control Table.

VCC_BIAS	VCC	Green LED	Red LED
Off	Off	Off	Off
On	Off	On	Off
On	On	Off	On
Off	On	N/A	N/A

C.2.10. POWER CONTROL REGISTER

The Power Control register is used to control the onboard power and determine the status of the peripheral board. This register can be located at either a memory or an I/O location, defined by the designer, but an I/O location is preferred. Because of the Hot Swap implementation, this register can be accessed only when power has been applied to the peripheral board.

Table C-3. Power Control Register.

Bit	Description	Function	Access
Bit 0	Power Control Bit	0 = Power Enabled, 1 = Powered Off	R/W
Bit 1	Power Interrupt Reset	0 = Enable Interrupt, 1 = Reset Power Interrupt	R/W
Bit 2	Power Interrupt	1 = Interrupt Occurred, 0 = No Interrupt	R
Bit 3	Power Down Request	1 = Request Power Down, 0 = No Request	R
Bit 4	Soft Reset	1 = Reset, 0 = Not Reset	R/W

Bit 0 - Power Control Bit (R/W). This bit is used to turn off the power to the peripheral board. Writing a "0" to bit 0 enables the power. Writing a "1" to bit 0 turns off the power, making the Power Control register unreadable until the peripheral board has been removed and reinserted by the user.

Bit 1 - Power Interrupt Reset (R/W). This bit is used to clear the Power Interrupt or Power Down Request bits and the Power Interrupt. Writing a "1" to bit 1 resets both the Power Interrupt and the Power Down Request bits, bit 2 and bit 3 respectively, and resets the Power Interrupt pin if enabled. A "0" must be written to this bit to enable the interrupt status bits to be enabled again after they are cleared. This bit is set to a "0" on power up or when a reset condition occurs.

Bit 2 - Power Interrupt (R only). This bit when set to a "1" indicates that the peripheral board has gone through a power cycle. It is reset by writing a "1" to bit 1 of the Power Control register.

Bit 3 - Power Down Request (R only). This bit is set to a "1" if the front panel peripheral board switch has been depressed by the user and VCC is applied to the peripheral board. This bit is set to a zero upon power up or when reset occurs. If this feature is not used, the user must drive bit 3 to a "0" during a read of the Power Control register.

Bit 4 - Soft Reset (R/W). This bit is used as a software reset to the peripheral board. Writing a "1" to this bit resets the peripheral board. Writing a "0" will undo the peripheral board reset. This bit is set to a zero during a system reset or a power up cycle. If this feature is not used, the user must drive bit 4 to a "0" during a read of the Power Control register.

C.2.11. FRONT PANEL PUSHBUTTON SWITCH

The front panel pushbutton switch is not required to implement a Hot Swap peripheral board, but is defined to maintain consistency across the Hot Swap product line.

Listed below are the functions of the pushbutton switch based on the different values of peripheral board power.

State 0: **VCC = Off, VCC_BIAS = Off**
Push the button, go to State 0.

State 1: **VCC = Off, VCC_BIAS = On**
Push the button, Set bit 0 of the Power Control register to zero to enable peripheral board power, go to State 2.

State 2: **VCC = On, VCC_BIAS = On**
Push the button, Set bit 3 of the Power Control register to a 1. Turn on the Power Interrupt, go to State 2.

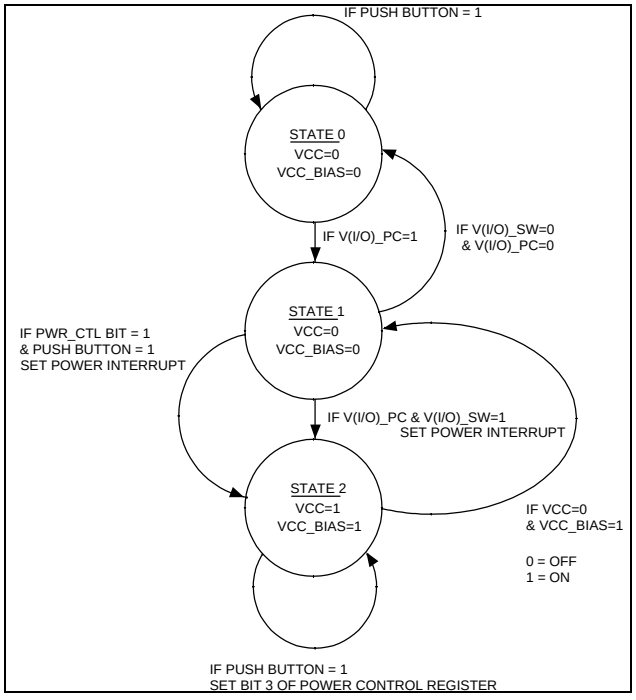


Figure C-9. Pushbutton Switch Operation.

HOT SWAP IMPLEMENTATION

APPENDIX C

A P P E N D I X D

STD 32 BUS SPECIFICATION HISTORY

STD 32 BUS SPECIFICATION HISTORY

APPENDIX D

D.1 INTRODUCTION

This appendix documents the changes made to the STD 32 Bus Specification and Designer's Guide since its initial release.

Note: The current specification uses single-part page numbering instead of the double-part page numbering (chapter number-page number) used in previous versions. Thus, the **Page** column in sections D.4 through D.8 of this appendix shows the current version's section numbers in brackets below the previous version's page number reference. In the current specification, find changes described in the **Change** column by referring to the section number given in the bracket.

Beginning in section D.9 of this appendix page numbers are no longer referenced; instead, section numbers alone are given to indicate where changes to previous versions have been made.

D.2 VERSION 1.0, SEPTEMBER 28, 1990

Version 1.0 was the initial specification release.

D.3 VERSION 1.0 ADDENDUM, NOVEMBER 30, 1990

Pages 5-5 and B-21 were replaced by corresponding pages in the STD 32 addendum. The page replacing 5-5 emphasized the importance of termination for large backplanes and that slot-specific signals should not be terminated. The pages supplementing Appendix B contained equations for the front end example presented in the STD 32 designer's guide.

D.4 VERSION 1.1 RELEASE, DECEMBER 24, 1991

The following is a list of the changes made to the STD 32 Bus Specification for version 1.1

Page	Change
1-3 [2.1]	Section 2.1 title changed from "STD-80 Compatible Bus Cycles" to "Standard Architecture Compatible Bus Cycles." Reference to E pins clarified.
1-4 [1.2.2]	Section 2.2 title changed from "STD 32 Compatible Bus Cycles" to "Extended Architecture Compatible Bus Cycles." References to STD 32 compatible bus cycles changed to Extended Architecture (EA). DREQx* corrected to DREQx. Reference to Permanent Master changed to full-featured Permanent Master. Reference to Slot X bus arbiter clarified.
1-6 [1.3]	Kb changed to Kbyte as abbreviation for Kilobyte; Mb changed to Mbyte as abbreviation for megabyte.
Chapter 2	Chapter title changed from "STD-80 Compatible Bus Cycles" to "Standard Architecture Bus Cycles."
2-2 [2.1]	Reference to Standard Architecture (SA) transfers added.
2-3 [2.2]	Table 2-1 title changed from "STD-80 Functional Pin Assignments" to "Standard Architecture Functional Pin Assignments."
2-7 [2.2.2]	Reference to Permanent and Temporary Masters not driving all 24 address lines removed.
2-8 [2.2.3]	"Permanent and Temporary Masters not driving all 24 address lines must terminate those not driven with a 10kΩ±10% resistor to +5V" changed to "Permanent Masters must terminate all address pins with a 4.7kΩ±10% resistor to +5V." The following text was added: Memory slaves should decode 24 bits of address. I/O slaves should decode 16 bits of address.
2-10 [2.2.4]	IOEXP I/O address range FC00-FCFFh corrected to FC00-FFFFh.
2-12 [2.2.4]	The following note was added to the MEM16* signal description: Note: Master implementations must allow sufficient time for MEM16* to be generated by a memory slave. A14 through A23 must be generated in time for tD2a and tD2b to be satisfied by a memory slave. Meeting only the setup time to ALE* rising (tS1, tS2) is not sufficient for satisfying address generation requirements.

APPENDIX D

2-17 [2.2.4]	The following comments were added to the WAITRQ* description: Masters must be capable of inserting two wait states to guarantee PC channel ready timing (CHRDY) compatibility. Warning: With WAITRQ* active, the Permanent Master may not be able to service interrupt requests, DMA requests, asynchronous data transfers, or dynamic RAM refresh. Masters must be able to complete on-board cycles with WAITRQ* asserted on the STD bus. For these reasons, it is best not to keep wait request asserted for more than 10µs. The Permanent Master must terminate WAITRQ* with a 300Ω±1% resistor to +5V.
2-21 [2.2.4]	Period removed from DRQ*/ DAK*.
2-23 [2.3.3]	References to STD-80 changed to SA.
2-24 [2.4]	Status timing diagram updated; tD2 split into tD2a and tD2b.
2-25 [2.4]	Read timing diagram updated.
2-26 [2.4]	Write timing diagram updated.
2-27 [2.4]	Interrupt timing diagram updated.
2-28 [2.4]	Fly-by DMA timing diagram updated.
Chapter 3	Chapter title changed from "STD 32 Compatible Bus Cycles" to "Extended Architecture Bus Cycles."
3-2 [3.1]	Reference to Extended Architecture (EA) added.
3-3 [3.1]	Table 3-1 title changed "STD 32" to "Extended Architecture."
3-4 [3.2]	Pin 56 description corrected from AUX Positive to AUX Negative.
3-5 [3.2]	Pin 49 Mnemonic corrected for BEI to BE1.
3-6 [3.2.4]	"STD 32 bus cycles" changed to "EA bus cycles."
3-12 [3.2.4]	IRQ* corrected to IRQx in the IRQx description.
3-15 [3.4]	"DMA Request Acknowledge Timing" changed to "Master Request/Acknowledge Timing." The following note was added: Additional timings will be provided in future versions of this specification.

3-16 [3.4]	Figure 3-2 title changed from DMA Request Acknowledge timing to Master Request Acknowledge Timing.
4-7, 4-10 [4.3.1]	A16 address range changed from FC00h-FCFFh to FC00h-FFFFh.
4-22 [4.3.5]	Permanent Master product descriptor [IX] changed to [IXP], [IXL].
5-8 [5.4.2]	Resistance termination value 300Ω±1% corrected to 316Ω±7%. Note 6 added.
6-2 [6.2]	Figure 6-1, STD-80 Series PCB Outline, dimensions updated.
6-3 [6.2.1]	Figure 6-2, STD 32 PCB Outline (with extensions), dimensions updated.
6-4 [6.2.2]	Figure 6-3, STD 32 PCB Outline (without Extensions), dimensions updated.
6-5 [6.2.3]	Figure 6-4, STD 32 PCB Tooling Hole Locations, dimensions updated.
6-11 [6.4.1]	Acceptable PCB Thickness changed from 0.0625 ±0.003 inch to 0.062 ±0.007 inch.
6-15 [6.4.2]	Figure 6-12, STD 32 P/E Design, was replaced with Figures 6-12a, 6-12b, and 6-12c. "Patent pending" changed to specify US Patent 5,061,190.
6-18 [6.4.2]	Table 6-4, STD 32 P/E Connector Specifications, added specifications for voltage drop, vibration, and hertz stress. Acceptable PCB thickness changed from 0.0625 ±0.003 inch to 0.062 ±0.007 inch.
6-19 through 6-23 [6.4.2]	"Patent Pending" updated to specify US Patent 5,061,190.
6-23 [6.4.2]	MREWQ5* corrected to MREQ5*.
A-3 [A.3]	STD 32 board tolerances changed from ±0.003 to ±0.005 inch.
Appendix C	STD 32 Bus Specification History added.
D-2 [glossary]	Modules corrected to modulus in hertzian stress definition.

D.5 VERSION 1.1 ADDENDUM, FEBRUARY 10, 1992

Pages 6-15 and 6-16 [6.4.1] were replaced by corresponding pages in the STD 32 addendum. The dimensions for the tooling hole shown in Figures 6-12a and 6-12b were corrected to read 0.125 DIA.

APPENDIX D

D.6 VERSION 1.2 RELEASE, SEPTEMBER 8, 1992

The following is a list of the changes made to the STD 32 Bus Specification for version 1.2.

Page	Change
1-4 [1.2.2]	DREQx corrected to DREQx*.
2-5 [2.2]	Pin 15 changed from MAKx* (Master Acknowledge) to DAKx* (DMA Acknowledge). Pin 16 changed from MREQx* (Master Request) to DREQx* (DMA Request). Pin 67 changed from LOCK* (Lock), Signal Flow Out, to INTRQ3* (Interrupt Request 3), Signal Flow In. Pin 69 changed from DAKx* (DMA Acknowledge) to MAKx* (Master Acknowledge). Pin 70 changed from DREQx* (DMA Request) to MREQx* (Master Request).
2-12 [2.2.4]	Termination resistor values changed from 300Ω±1% to 316Ω±7% for IO16* and MEM16*.
2-15 through 2-20 [2.2.4]	Termination resistor value changed from 300Ω±1% to 316Ω±7% for BUSRQ*, INTRQ*, INTRQ1*, INTRQ3*, WAITRQ*, NMIRQ*, PBRESET*, and CNTRL*.
2-16 [2.2.4]	INTRQ3* (E Pin 67) added to INTRQ* and INTRQ1* description. Reference to P interrupt request corrected to interrupt request.
2-21 [2.2.4]	All references to DREQ* corrected to DRQ*.
2-22 [2.3.1]	References to MREQx* and MAKx* changed to DREQx* and DAKx*. MX compliance designation changed to DX compliance designation. Reference to timing diagram Figure 2-10 added. Table 2-9 title changed from "Signals that Temporary Masters Optionally Manage" to "Signals that Standard Architecture Temporary Masters Optionally Manage." INTRQ3* added to INTRQ*, INTRQ1*, and CNTRL* (INTRQ2*) as signals Temporary Masters may also manage.
2-23 [2.3.2]	INTRQ3* added to INTRQ*, INTRQ1*, and CNTRL* (INTRQ2*) as signals I/O slaves should implement. Added a reference to the "SA Master Request/Acknowledge Timing" diagram.
2-28 [2.4]	An incorrect reference to tH2 was deleted from the "Fly-By DMA Timing" diagram.
2-31 [2.4]	Figure 2-10, "(SA) Master Request Acknowledge Timing," was added.
3-3, 3-5 [3.1]	LOCK* reassigned to E pin 2. MAKx* and DAKx* pin designations swapped. MREQx* and DREQx* pin designations swapped. DREQx corrected to DREQx*. Pin 67 reassigned to INTRQ3*.
3-8 through 3-13 [3.2.4]	Termination resistor value changed from 300Ω±1% to 316Ω±7% for MASTER16*, MEM16*, IO16*, EX8*, EX16*, EX32*, NOWS*, SLBURST*, EXRDY, T-C.

3-8 [3.2.4]	The description of AENx* was rewritten.
3-10 [3.2.4]	SLBURST* and MSBURST* descriptions were moved from page 3-13. The MSBURST* description changed "The Temporary Master must drive MSBURST*" to read "The Temporary Master must manage MSBURST*."
3-11 [3.2.4]	MAKx* pin number changed from 15 to 69. "A Temporary Master generates a master request" changed to "An EA Master generates a master request." MREQx* pin number changed from 16 to 70. "By a master" changed to "by an EA master". The following text was added to the MREQx* description: SA masters do not use this signal for acquiring the bus." The MREQx* and MAKx* entries in Table 3-4 changed to DREQx* and DAKx*, respectively. The DREQx entry was corrected to DREQx*.
3-12 [3.2.4]	The pin number for LOCK* changed from 67 to 2.
3-13 [3.2.4]	All references to DREQx were corrected to DREQx*. The pin numbers for DAKx* and DREQx* changed from 69 and 70 to 15 and 16, respectively. A reference to the DREQx* description was added to the DAKx* description. The DREQx description changed from active-high to active-low. The following text was added to the DREQx* description: SA bus masters may also use DREQx* to gain access to system resources in a multiple master architecture. In this mode, SA bus masters generate DREQx* synchronous to the system clock. The Slot X Arbiter generates DAKx* synchronous to the system clock after arbitration. DREQx* Permanent Master termination of DREQx* changed from termination to ground to termination to +5V. SLBURST* and MSBURST* descriptions moved to page 3-10.
3-14 [3.2.4]	"DREQx*" was added in the following sentence: "The Permanent Master must resistively terminate all signals that could be managed by a Temporary Master in response to a BUSRQ*, DREQx*, or MREQ*."
3-16 [3.2.4]	"MERQx*" corrected to "MREQx*."
4-4 [4.2.1]	Arbiter mode description changed from "Arbiter (n) - Slot X arbiter that supports n MREQx* requests" to "Arbiter (Mn, Dn) - Slot X arbiter that supports Mn MREQx* requests and / or Dn DREQx* requests for bus ownership".
4-5 [4.2.1]	"Arbiter (n)" changed to "Arbiter (Mn,Dn)".
4-8 [4.3.1]	I option description changed from "STD-80 interrupt generation by slaves, servicing by masters of NMIRQ*, INTRQ*, INTRQ1*, and optionally CNTRL* (INTRQ2*)" to "STD-80 interrupt generation by slaves, servicing by masters of any of the interrupts NMIRQ*, INTRQ*, INTRQ1*, CNTRL* (INTRQ2*), or INTRQ3*." MX option changed to MD. Signal names MREQx* and MAKx* changed to DREQx* and DAKx*, respectively.

APPENDIX D

4-9 [4.3.1]	Permanent Master and Temporary Master product descriptor designator [MX] changed to [MD].
4-11 [4.3.2]	MX option changed to MD. Signal names MREQx* and MAKx* changed to DREQx* and DAKx*, respectively. Permanent Master and Temporary Master product descriptor designator [MX] changed to [MD].
4-13 [4.3.3]	"Class EA8 I/O Slaves do not support XA16-XA23 or XA24*-XA31*" changed to "Class EA8 I/O Slaves do not use XA16-XA23 or XA24*-XA31*." "EA8 I/O Slaves must include AENx* in their decode logic" changed to "EA8 I/O Slaves use AENx* for slot-specific addressing. Slot-specific addressing allows the permanent master to configure each peripheral board at power up. Normal I/O accesses do not use AENx* in the decode logic."
4-15 [4.3.4]	The following was added to the end of section 3.4, Class EA16 Boards: EA Masters use MREQx* and MAKx* to gain control of the bus. BUSRQ* and DREQx* are for SA Master use only. However, an EA Permanent Master may service SA Temporary Masters by acknowledging BUSRQ* with BUSAK*.
4-16 [4.3.4]	The text appearing in italics below was added to EA16 Requirements: EA16 I/O Slaves must include AENx* in their decode logic *for slot-specific configuration only*.
4-17 [4.3.4]	MX option description changed to delete the following: - Temporary Master that requests the bus via BUSRQ*.
4-18 [4.3.4]	Temporary Master product descriptor designator [MB] removed.
4-19 [4.3.5]	The following was added to the end of section 3.5, Class EA32 Boards: EA Masters use MREQx* and MAKx* to gain control of the bus. BUSRQ* and DREQx* are for SA Master use only. However, an EA Permanent Master may service SA Temporary Masters by acknowledging BUSRQ* with BUSAK*.
4-20 [4.3.5]	The text appearing in italics below was added to EA32 Requirements: EA32 I/O Slaves must include AENx* in their decode logic *for slot-specific configuration.*
4-23, 4-24, 4-26, 4-27 [4.4.1]	Arbiter class replaced with the following: Arbiter (Mn, Dn) MREQ0*-MREQ(Mn)*, MAK0*-MAK(Mn)* and/or DREQ0*-DREQ(Dn)*, DAK0*-DAK(Dn)*
4-23, 4-24 [4.4.1]	Temporary Master MREQ* signal replaced with DREQx*.
4-25 [4.4.3]	Permanent Master, Temporary Master, and Arbiter classes removed from the table.
4-26, 4-27 [4.4.5]	Temporary Master MREQ* signal corrected to MREQx*. Temporary Master BUSRQ* signal deleted.

4-28 [4.5.1]	INTRQ3* added to the description and additional signal support for the I product descriptor feature; "and optionally" deleted from the I product descriptor feature description.
4-29 [4.5.1]	The MB description changed from "bus arbitration via BUSRQ*/BUSAK* (Permanent Master)/request (Temporary Slave), low-level asserted" to "bus arbitration via BUSRQ*/BUSAK* -SA Masters only." The MD feature was added. The MX description was changed from "bus Arbitration via MREQx*/MAKx* (Permanent Master)/request (Temporary Slave)" to "bus Arbitration via MREQx*/MAKx* - EAx Masters only."
4-31 [4.5.2]	Table 4-2 was updated. An MD column was added. Corrections were made to the MB and MX columns to match the text.
5-7 [5.4.2]	MERQx* corrected to MREQx*. DREQx* corrected to DREQx*.
5-8 [5.4.2]	Note 7 added to the A0-A23 resistance termination value. IORQ* removed from Table 5-5.
5-9 [5.4.2]	DREQx corrected to DREQx*. DREQx* termination value no longer calls note 4. INTRQ3* added to INTRQ*, INTRQ1*, CNTRL (INTRQ2*). Note 7 added.
6-7 [6.3]	An asterisk was added to DREQ0 and DREQ1.
6-15 [6.4.2]	500µ inches of nickel changed to 200µ inches. 0.125 changed to ∅ 0.109.
6-18 [6.4.2]	Vibration changed from "5 to 500 Hz @ 1.5G" to "10 to 2kHz @ 15Gs/0.06" displacement." Contact normal force changed from "75 grams per contact" to "135 grams per contact." Connector body changed from "high temperature range for vapor and wave soldering" to "glass filled polyphenylene sulfide, UL 94V-0." Insulation resistance changed from "500MΩ min" to "50,000MΩ min." The hertz stress category was removed.
6-19 through 6-23 [6.4.2]	Figures 6-13 through 6-17 moved to pages 6-20 through 6-24, respectively. Pin assignments on these pages changed as follows: MREQx* designations changed to DREQx*. DREQx* designations changed to MREQx*. MAKx* designations changed to DAKx*. DAKx* designations changed to MAKx*. LOCK* was moved to pin E02. INTRQ3* was assigned to pin E67.

APPENDIX D

B-3 [B.1.1]	The following text was added to the end of the second paragraph of the answer to the first question: In addition, INTRQ3* is a bussed interrupt that has been added to relieve backplane interrupt congestion.
B-6 [B.2.3]	The end of the second sentence under section 2.3, Interrupt Generation, now reads "and the bussed signal INTRQ3* defined by STD 32."
B-10, B-11 [B.3.2]	Schematics changed to remove requirement for AENx* during Extended Architecture decoding.
B-12 [B.3.2]	"EA transfers" section replaced with a "slot-specific considerations" section and a revised "EA transfers" section.
B-14 [B.3.2]	"EX16* and EA are used by the master logic..." corrected to "EX16* and EX8* are used by the master logic..."
B-15 [B.3.2]	"[EA mode] command signals are derived from CMD*, W_RL, address, and AENxL*" changed to "... CMD*, W_RL, and BD_SEL."
B-18 through B-25 [see code at end of Appendix]	Application examples replaced.

C.7 VERSION 1.3 RELEASE, MARCH 8, 1994

The following is a list of the changes made to the STD 32 Bus Specification for version 1.3.

Page	Change
Title Page	"STD 32" changed from trademark to registered trademark. Ziatech street address updated.
2-6 [2.2.1]	Table 2-4, renumbered footnote 1 as footnote 2. Inserted new footnote (numbered 1) after "Battery Backup Source"; footnote reads "The Vbat supply is optional."
2-7 [2.2.2]	Added to the end of first paragraph: The Permanent Master must terminate all data pins with a 1.8kΩ±10% resistor to +5V.
2-8 [2.2.3]	End of first paragraph, changed "4.7kΩ±10%" to 1.8kΩ±10%.
2-10 [2.2.4]	Added at the end of the WR* signal description: The Permanent Master must terminate WR* with a 1.8kΩ±10% resistor to +5V.
	Added at the end of the RD* signal description: The Permanent Master must terminate RD* with a 1.8kΩ±10% resistor to +5V.
2-11 [2.2.4]	Added at the end of the IORQ* signal description: The Permanent Master must terminate IORQ* with a 1.8kΩ±10% resistor to +5V.
	Added at the end of the RD* signal description: The Permanent Master must terminate RD* with a 1.8kΩ±10% resistor to +5V.
	Added at the end of the MEMRQ* signal description: The Permanent Master must terminate MEMRQ* with a 1.8kΩ±10% resistor to +5V.
	Added at the end of the IOEXP* signal description: The Permanent Master must terminate IOEXP* with a 1.8kΩ±10% resistor to +5V.
2-12 [2.2.4]	Added at the end of the BHE* signal description: The Permanent Master must terminate BHE* with a 1.8kΩ±10% resistor to +5V.
2-13 [2.2.4]	Added at the end of the ALE* signal description: The Permanent Master must terminate ALE* with a 1.8kΩ±10% resistor to +5V.
	Added at the end of the STATUS1* signal description: The Permanent Master must terminate STATUS1* with a 1.8kΩ±10% resistor to +5V.
2-14 [2.2.4]	Added at the end of the STATUS0* signal description: The Permanent Master must terminate STATUS0* with a 1.8kΩ±10% resistor to +5V.

APPENDIX D

2-15 [2.2.4]	Deleted existing BUSAK* signal description. Inserted the following description for BUSAK*: Bus Acknowledge is an active-low signal driven by the current bus master. BUSAK* can be used to signal a bus ownership change to another bus master or to indicate a DMA acknowledge. BUSAK* is generated in response to BUSRQ*. BUSAK* can be driven by a totem-pole driver or three-state driver. BUSAK* must be actively driven to either logic level and not allowed to float high. The Permanent Master must terminate BUSAK* with a 316Ω±7% resistor to +5V. Deleted existing BUSRQ* signal description. Inserted the following description for BUSRQ*: Bus Request is an active-low signal driven by either a Temporary Master for bus ownership or a DMA slave for DMA request. BUSRQ* is driven by an open collector driver. This signal may be driven asynchronously, and it is the responsibility of the receiving agent to synchronize BUSRQ*. Temporary Masters must manage the signals shown in Table 2-7. The Permanent Master must terminate BUSRQ* with a 316Ω±7% resistor to +5V. Table 2-7 title, changed "Temporary Masters" to "Standard Architecture Temporary Masters."
2-16 [2.2.4]	Under WAITRQ*, deleted "Masters must be capable of inserting two wait states to guarantee PC channel ready timing (CHRDY) compatibility."
2-17 [2.2.4]	Added at the end of paragraph 1: It is recommended that peripheral boards drive WAITRQ* only during SA cycles.
2-18 [2.2.4]	Under both PCO and PCI, added "of STD-80" to the end of the first sentence. Deleted rest of paragraph, for both PCO and PCI, and replaced it with the following: STD 32 peripherals do not use the PCI-PCO feature. PCI and PCO should be connected together on STD 32 designs.
2-21 [2.3.1]	Under "Memory Refresh," changed "during a refresh cycle" to "during an on-board refresh cycle." Under "Bus Termination," changed "MREQx*" to DREQx*, changed "Table 2-9" to Table 2-7, and changed last sentence to "Resistive termination should be 1.8kΩ±10% pullup to +5V, unless otherwise specified." In paragraph 1 under "Optional Signal Management," changed "A true multiple master system" to "A true multiple SA master system."
2-23 through 2-30 [2.4]	In Figures 2-3 through 2-10, added footnote: All times given in nanoseconds. Figure 2-4, changed tD5 MAX from 225 to 140. Figure 2-6, figure title changed to Interrupt Acknowledge Timing.
3-2 [3.1]	Table 3-1, deleted E pins 13, 14, and 68 from the "Logic Ground" section. Added a new line directly below the "Logic Ground" line: +5VDC (E Pins 13, 14, 68).

3-4 [3.2]	Table 3-3, corrected MREQx to MREQx* (pin 70).
3-5 [3.2.2]	"Data Bus Additions," end of first paragraph: changed "10kΩ±10%" to 1.8kΩ±10%. "Address Bus Additions," second paragraph, changed "(BUSRQ*)" to (BUSRQ* or DREQx*).
3-7 [3.2.4]	AENx* description, changed "as an address qualifier" to "as an I/O address qualifier." Deleted "for a slot-specific configuration. A board that supports configuration by the permanent master uses AENx* as an address qualifier. AENx* must not be used to decode normal (nonconfiguration) cycles." Replaced that text with the following: AENx* is driven low for I/O address ranges x100h - x3FFh, x500h - x7FFh, x900h - xBFFh, and xD00h - xFFFh, where "x" is any value. It is driven active to slots 1 through 14 for the above I/O ranges. AENx* is also driven active on a slot-by-slot basis for I/O addresses z000h - z0FFh, z400h - z4FFh, z800h - z8FFh, and zC00h - zCFFh, where "z" corresponds to the specific slot. This allows the Permanent Master to perform slot-specific configuration. Extended Architecture I/O peripherals should decode AENx* as part of its I/O address, but should also provide an option to ignore AENx* for simple Permanent Masters that do not have Slot X capability (and therefore cannot drive AENx*). Added at the end of the BE0*-BE3* description: The Permanent Master must terminate the BE0* - BE3* signals with a 1.8kΩ±10% resistor to +5V. Added at the end of the M-IO description: The Permanent Master must terminate M-IO with a 1.8kΩ±10% resistor to +5V.
3-8 [3.2.4]	Added at the end of the W-R description: The Permanent Master must terminate W-R with a 1.8kΩ±10% resistor to +5V.
3-9 [3.2.4]	At the end of the MSBURST* and START* descriptions, changed "10kΩ±1%" to 1.8kΩ±10%.
3-11 [3.2.4]	At the end of the LOCK* description, changed "1kΩ±1%" to 1.8kΩ±10%.
3-12 [3.2.4]	T-C description, changed "driven by the Permanent Master" to "driven by the Master." Deleted the final sentence, "The Permanent Master must terminate T-C with a 316Ω±7% resistor to +5V."
3-13 [3.3.1]	Under "Bus Termination," changed "10kΩ±10%" to 1.8kΩ±10%. Under "Timing," deleted the three existing sentences and added the following: Figure 3-2 on the following page illustrates the critical timing parameters from the point of view of the Permanent Master. Figures 3-3 - 3-5 illustrate Extended Architecture STATUS, EXRDY/EX32*/EX16*/EX8*, and DATA ACCESS, respectively.

APPENDIX D

3-14 through 3-17 [3.4]	Added Figures 3-3 through 3-5, pages 3-15 through 3-17. Added note to Figures 3-2 through 3-5: All times given in nanoseconds.
4-8 [4.3.1]	SA8 Options: IXP and IXL options, added sentence "Slaves drive and Slot X capable Permanent Masters decode this signal."
	Added SDMABP option, plus description: Standard Architecture DMA using BUSRQ*/BUSAK* for request and acknowledge and the control signals DMAIOR*, DMAIOW*, and T-C.
4-9 [4.3.1]	SA8 Product Descriptors: Permanent Master, deleted "-D8, A24." Added [SDMABP].
	Temporary Master, deleted "-D8, A24." Added [IXP], [IXL].
	I/O Slave, deleted "-D8, A16." Added [SDMABP].
	Memory Slave, deleted "-D8, A24."
	Deleted "Implicit required features are included here for reference in boldfaced type, but would not be included since this information is embedded in the description of the class."
4-10 [4.3.2]	SA16 Requirements: I/O Slaves, deleted the D8 line ("8-bit data [D0-D7] transfer as defined in Chapter 2").
	Memory Slaves, deleted the D8 line ("8-bit data [D0-D7] transfer as defined in Chapter 2").
4-11 [4.3.2]	SA16 Options: IXP and IXL options, added sentence "Slaves drive and Slot X capable Permanent Masters decode this signal."
	MD option, Temporary Master, changed "MREQx*" and "MAKx*" to "DREQx*" and "DAKx*," respectively.
	Added SDMABP option, plus description: Standard Architecture DMA using BUSRQ*/BUSAK* for request and acknowledge and the control signals DMAIOR*, DMAIOW*, and T-C.
	SA16 Product Descriptors: Permanent Master, deleted "-D16, D8, A24." Added [SDMABP].
	Temporary Master, deleted "-D16, D8, A24." Added [IXP], [IXL].
	I/O Slave, deleted "SA8-D16, D8, A16." Added [SDMABP].
	Memory Slave, deleted "SA8-D16, D8, A24."
	Deleted "Implicit required features are included here for reference in boldfaced type, but would not be included since this information is embedded within the description of the class."

4-14 [4.3.3]	EA8 Options: IXP and IXL options, changed the first sentence to read "Slot-specific interrupt support, Permanent Masters/Slaves, positive edge-triggered." Added second sentence, "Slaves drive and Slot X capable Permanent Masters decode this signal."
	EA8 Product Descriptors, changed "product descriptor features" to "product descriptor possibilities."
	I/O Slave, deleted "-XD8, XA16, [D16, A16], [D8, A16]."
	Memory Slave, deleted "-XD8, XA32, [D16, A24], [D8, A24]."
	Deleted "Implicit required features are included here for reference in boldfaced type, but would not be included since this information is embedded within the description of the class."
4-17 [4.3.4]	EA16 Options: IXP and IXL options, added second sentence, "Slaves drive and Slot X capable Permanent Masters decode this signal."
4-18 [4.3.4]	EA16 Product Descriptors, changed "product descriptor features" to "product descriptor possibilities."
	Permanent Master, deleted "-XD16, XA32, [D16, A24], [D8, A24]."
	Temporary Master, deleted "-XD16, XA32, [D16, A24]."
	I/O Slave, deleted "-XD16, XA16, [D16, A16]."
	Memory Slave, deleted "-XD16, XA32, [D16, A24]."
	Deleted "Implicit required features are included here for reference in boldfaced type, but would not be included since this information is embedded within the description of the class."
4-22 [4.3.5]	EA32 Product Descriptors, changed "product descriptor features" to "product descriptor possibilities."
	Permanent Master, deleted "-XD32, XD16, XA32, [D16, A24], [D8, A24]."
	Temporary Master, deleted "-XD32, XD16, XA32, [D16, A24], [D8, A24]."
	I/O Slave, deleted "-XD32, XD16, XA16, [D16, A16], [D8 A16]."
	Memory Slave, deleted "-XD32, XD16, XA32, [D16, A24], [D8, A24]."
	Deleted "Implicit required features are included here for reference in bold face type, but would not be included since this information is embedded within the description of the class."
4-23 [4.4.1]	Under "Temporary Master," corrected "DREQ*" to DREQx*.

APPENDIX D

4-28 [4.5.1] Table 4-1, continued: after the SDMA16 entry, inserted SDMABP entry. Description column: "8- or 16-bit backplane DMA supported via BUSRQ*/BUSAK* and DMAIOR*, DMAIOW*, T-C." Additional Signal Support column: "BUSRQ*, BUSAK*, DMAIOR*, DMAIOW*, T-C."

4-30 [4.5.2] Table 4-2: I/O Slaves section, SA16 class, changed D8 column from "R" to O. Changed D16 column from "-" to R.

Memory Slaves section, SA16 class, changed D8 column from "R" to O.

Options columns, changed "SDMA" heading to SDMA8. Added new columns SDMA16 and SDMABP next to SDMA8 column.

New SDMA16 column: All entries are dashes (= "Not Applicable") except for the following rows, which are O's (= "Optional"): Permanent Masters SA16, Temporary Masters SA16, and I/O Slaves SA16.

New SDMABP column: All entries are dashes (= "Not Applicable") except for the following rows, which are O's (= "Optional"): Permanent Masters SA8 and SA16, Temporary Masters SA8 and SA16, and I/O Slaves SA8 and SA16.

5-5 [5.3.2] After SIGNAL TERMINATIONS heading, inserted the following subsection:

Diode Termination. To minimize signal reflections, a diode termination can be placed at both ends of the backplane signal trace to absorb undershoot and overshoot of signal levels. Figure 5-1 shows an example of diode termination. Several IC vendors (for example, Texas Instruments) offer diode termination arrays explicitly for this purpose. Diode termination is the preferred backplane termination mechanism for STD 32.

Inserted new Figure 5-1, "Diode Termination."

Inserted new heading, **RC Termination**, above the existing text on page 5-5; moved heading and text to page 5-6. End of first sentence in paragraph 1, changed "should be used" to "can be used." Added to end of paragraph 1: "Figure 5-2 shows an example of RC termination." Renumbered old Figure 5-1 as Figure 5-2.

Inserted new heading, **Other Considerations**, before paragraph 3 ("Slot-specific signals..."). Added to paragraph 3: It is suggested that series termination be used on these signals at the source (driver) to minimize undershoot and overshoot at the receiving device.

5-8 [5.4.2] Table 5-5: For signals D0-D31, A0-A23, and XA24*-XA31*, changed the resistance termination value to 1.8kΩ±10%. Deleted CLOCK* from the group of signals beginning with WR*. Added resistance termination value of 1.8kΩ±10% to that group of signals.

APPENDIX D

5-9 [5.4.2]	Table 5-5, continued: For signals MSBURST*, START*, and LOCK*, changed the resistance termination value to 1.8kΩ±10%. Added CLOCK* to the CMD*, SYSRESET* group of signals. Corrected bus signal "TC" to T-C. Note 5, deleted "Termination should be 10kΩ±10% pullup to +5V." Deleted note 7.
6-2 through 6-4 [6.2-6.2.1]	Figures 6-1 through 6-3, below the words "COMPONENT SIDE," changed "0.015 X 45° BEVEL" to 0.015±5% X 45° BEVEL.
6-5 [6.2.3]	Changed "Each STD board" to "Each STD 32 board." In second sentence, deleted "and any future front panel mounting brackets." Deleted third sentence ("Front panel mounting..."). After fourth sentence, added "Additional tooling holes are suggested for manufacturing purposes. Typically, tooling holes are provided at diagonal locations of a PCB to aid board fabrication procedures. The location of any additional tooling holes is at the discretion of the board designer."
6-7 [6.3]	Figure 6-6, deleted "PCI PCO" and accompanying line.
6-9 [6.3.1]	Paragraph 2, changed "locations will be provided" to "locations for power distribution will be provided." Paragraph 3, changed "wiring behind the backplane" to "wiring between the backplane and any sheet metal or enclosure."
6-10 [6.3.2]	Table 6-2, LT column, changed "0.003" to 0.007 in both rows. LH column, changed "0.385" to 0.381, and "0.510" to 0.506.
6-12 [6.4]	Paragraph 4, changed "Card edge fingers must have" to "For use in STD 32 backplanes, it is recommended that card edge fingers have." Changed "50µ inches" to 200µ inches.
6-17 [6.4.2]	Figure 6-13, lower left, changed "2X 0.033 RADIUS" to 2X 0.06 RADIUS.
6-21 [6.4.2]	Figure 6-16, lower right, pin E76, corrected "X30*" to XA30*.
A-3 [A.3]	Card edge connectors, paragraph 2, changed "50µ inches of nickel" to 100µ inches of nickel.
A-4 [A.4]	Termination section: Deleted existing sentence, added "STD 32 recommends diode termination networks for non-Slot X signals to absorb overshoot and undershoot energy and minimize incident wave switching noise."
B-5 [B.2.2]	Next-to-last paragraph, deleted sentence "All cycles start as both SA and EA since all control signals are driven."
B-7 [B.2.4]	In section B.2.4., moved the last two sentences of the first paragraph ("The front end...", "The following signals..."), the list of signals, and the next sentence ("This allows...") to page B-9, just before the "Application Example" section. Inserted the heading "DMA Logic" for the new section on page B-9.

B-8 [B.2.5]	Deleted the last sentence in section B.2.5., "STD 32 assumes that SA memory designs will be used with SA masters and EA memory designs with EA masters."
B-9 [B.3.1]	Under "Design Features," changed "Figure B-1" to "schematic 1 at the end of the STD 32 specification." Changed reference to Figure B-2 similarly, here and in the last sentence on the page.
B-10 [see end of spec]	Moved Figures B-1 and B-2 (schematics) from Appendix B to the end of the specification, on 11 x 17 foldout pages. In Figure B-1 ("schematic 1"), lower right, BACKPLANE E PINS, pin E54, changed "VCC" to GND.
B-12 [B.3.2]	Under "Board Select Decoding," signals in (), added AENx*, IORQ*, M-IO. Moved the first sentence, "BD_SEL* is decoded...," out of the "STD-80 or SA transfers" section; placed it just beneath the (). In the "STD-80 or SA transfers" section, deleted "W10-13 and W32-35 are removed." Moved the "EA transfers" section up before the "Slot-Specific Consideration" section. In the "EA transfers" section, added AENx* as follows: "In the design example, BD_SEL* is latched by START* based on EQ*, M-IO, and AENx*." In the "Slot-Specific Consideration" section, deleted "Only the slot-specific logic can use AENx* in its decode. Normal I/O must not use AENx* in its decode logic." Next paragraph, changed "For normal I/O logic (nonconfiguration logic)" to "For support by masters without AENx* capability". At the end of that paragraph, added "This is done by removing W38 in the application example."
B-13 [B.3.2]	Choice 2, added at end: AENx* may be ignored for masters that do not have AENx* support. "Internal Address Bus and Control" paragraph, added at end: IAB0 and IAB1 are decoded from BE0-3* when in EA mode
B-16 through B-23 [see end of Appendix B]	Deleted Designer's Guide example code, replaced with new code.
C-5 [D.4]	Page 6-18, moved "6-19 through 6-23" to the left-hand column, next to the "Patent Pending" entry.
C-8 [D.6]	Page 4-16, italicized the words "for slot-specific configuration only." Page 4-20, italicized the words "for slot-specific configuration."
C-9 [D.6]	Page 6-15, deleted "RAD." Deleted page 6-16 entry.

D.8 VERSION 2.0 RELEASE, AUGUST 4, 1995

The following is a list of the changes made to the STD 32 Bus Specification for Version 2.0.

Page	Change
E-1 to E-18 [C.1]	Added Appendix E, "Hot Swap Implementation," describing in detail the implementation of Hot Swap on the STD 32 bus.

D.9 VERSION 2.1 RELEASE, JULY 3, 1996

The following is a list of the changes made to the STD 32 Bus Specification for Version 2.1

Section	Change
Title Page	Changed Title page from "Version 1.3" to "Version 2.1". Changed date on title page from "March 8, 1994" to "July 3, 1996".
Copyright Page	Changed date on copyright page from "1994" to "1996".
[2.1]	Added reference to page 14 in last sentence of fourth paragraph.
[2.2.3]	Figure 2-1 was moved from page 2-9 to page 14.
[2.2.4]	BUSAK* (P pin 41) and BUSRQ* (P pin 42) in the "Interrupt and Bus Control" discussion were moved from page 2-15 to page 19.
	INTAK*, INTRQ*, INTRQ1* and INTRQ3* discussion was moved from page 2-16 to page 20.
	Figure 2-2 and NMIRQ* discussion were moved from page 2-17 to page 21.
	PBRESET* discussion was moved from page 2-19 to page 22.
[3.4]	Figure 3.3: corrected SYSCLOCK* to CLOCK*.
	Figure 3.4: corrected SYSCLOCK* to CLOCK*.
	Figure 3.5: corrected SYSCLOCK* to CLOCK*.
	Figure 2-3: corrected IO16* line; tD2a and tD2b MAX values were corrected.
[4.1.3]	First sentence in second paragraph in section was changed from "Boards that are compatible with a class may use signals associated only with that class during transfers of that class level" to "Boards that are compatible with a class may use only those signals associated with that class during transfers of that class level".
[5.4.2]	Table 5-5: top three cells in "Location" column changed from "Permanent Masters" to "All Masters"; added "All Masters" in fourth cell from top in "Location" column; in "Bus Signals" column changed BEO* to BE0* and MAKX* to MAKx*; table 5-5 (Continued): changed second to last cell in "Locations" column from "Permanent Masters" to "All Masters".
[6.2]	Figure 6-1 was corrected.
[6.2.1]	Figure 6-2 was corrected.
[6.2.2]	Figure 6-3 was corrected.
[6.3]	Figure 6-6 was corrected.
[6.4.1]	Figure 6-9 was corrected.
[6.4.2]	Figures 6-11, 6-12, 6-13 and 6-14 were corrected.

STD 32 BUS SPECIFICATION HISTORY

APPENDIX D

[B.3.2] In the WAITRQ* and EXRDY discussion: corrected three instances of SYSCLOCK* to CLOCK*.

[C.1] Incorporated into Version 2.1 of the specification the "Addendum to the STD 32 Bus Specification For Version 2.0" / Appendix E. "Hot Swap Implementation" (referred to in section D.8 above). As a result, the "Hot Swap Implementation" was changed from "Appendix E" to "Appendix D" and the page numbering range was changed from pages E-1 through E-16 to pages 143 through 158.

Corrections were made to several figures in the "Hot Swap Implementation" appendix. These figures are listed below. Additionally, their figure numbers were changed to reflect the renaming of the appendix in which they appear (formerly Appendix E, now Appendix C; see above paragraph.) and their page numbers were changed to single-part page numbering as follows:

Figure E-1, page E-4	was changed to	figure C-1, page 146
Figure E-2, page E-5	was changed to	figure C-2, page 147
Figure E-4, page E-7	was changed to	figure C-4, page 149
Figure E-6, page E-9	was changed to	figure C-6, page 151

[D.1] Renamed the "STD 32 Bus Specification History" from "Appendix C" to "Appendix D" and changed the page numbering range from pages C-1 through C-18 to pages 159 through 178.

[D.9] Section D.9 was added to the specification revision history for version 2.1.

[E.1] Renamed the "Glossary" from "Appendix D" to "Appendix E" and changed the page numbering range from pages D-1 through D6 to pages 179 through 184.

Added the following definition to page 181 (formerly page D-3) of the glossary: **Hot Swap:** The insertion or removal of boards from a powered backplane without disturbing the system.

APPENDIX E

GLOSSARY

arbiter A board residing in Slot X that receives MREQx* and/or BUSRQ*, determines priority or requestors, and drives MAKx* or BUSAK* to the highest priority requestor.

bit A binary digit (either 0 or 1).

byte A group of eight bits ranging in value from 0 to 255.

byte 0 The first of four possible bytes that comprise the 32-bit data path on the STD 32 bus. Byte lane 0 consists of data bus bits 0-7.

byte 1 The second of four possible bytes that comprise the 32-bit data path on the STD 32 bus. Byte lane 1 consists of data bus bits 8-15.

byte 2 The third of four possible bytes that comprise the 32-bit data path on the STD 32 bus. Byte lane 2 consists of data bus bits 16-23.

byte 3 The fourth of four possible bytes that comprise the 32-bit data path on the STD 32 bus. Byte lane 3 consists of data bus bits 24-31.

GLOSSARY

byte copying The duplication of one or more bytes on one or more byte lanes.

byte lane One of four possible bytes that comprise the 32-bit data path on the STD 32 bus. Byte lanes are numbered 0 to 3, from least significant to most significant, respectively.

bus devices May have one or more of these capabilities: DMA Slave, I/O Slave, Memory Slave, Permanent Master, Temporary Master.

cascade address The address generated by a Master PIC in response to an interrupt generated by a Slave PIC. The cascade address is multiplexed on address lines A8-A10, allowing a unique slave PIC to return an interrupt vector.

DMA Direct Memory Access.

dword A group of 32 bits ranging in value from 0 to 4,294,967,296.

DMA slave A peripheral device that requests service from a Permanent Master or Temporary Master and responds to DMA cycles.

E STD bus connector expansion signals located between existing P connector contacts. The E signals are the difference between the STD 32 bus connectors and STD-80 connectors.

EA Extended Architecture.

extensions Additional signals on either side of the existing STD bus card edge signals.

Hertzian stress Stress expressed in p.s.i. that is developed during the elastic deformation phase of establishing contact. The stress is a result of normal force and the geometry of the contact and the modulus of elasticity of the contact material.

Hot Swap The insertion or removal of boards from a powered backplane without disturbing the system.

I/O Slave A peripheral card that responds to I/O cycles from either Permanent or Temporary Masters.

master The card currently in control of the bus driving backplane control signals.

Master PIC Programmable Interrupt Controller on the Permanent or Temporary Master that is optionally multiplexing cascade address on address lines A8-A10.

memory slave Peripheral card that responds to memory cycles from either Permanent or Temporary Masters.

P STD bus connector signals that comprise the STD-80 bus.

packing Combining a word or dword of smaller elements so that they may be transferred in one bus cycle to minimize bus bandwidth usage.

Permanent Master The Permanent Master typically inserts into card cage Slot 0 and optionally Slot X when maintaining the slot specific signals for servicing DMA requests, arbitrating for ownership of the bus, and handling slot specific interrupts. The Permanent Master is responsible for driving CLOCK* and SYSRESET*.

SA Standard Architecture (STD-80).

slave The card currently under control by the Master. Slave cards may be I/O Slaves, Memory Slaves, or Temporary Masters not currently in control of the bus.

slot A physical position in which a board can be installed into an STD bus backplane. The slot may be STD-80 bus compatible with P connector signals implemented or may be STD 32 bus compatible with P and E connector signals implemented.

slot-specific A term used to reference certain signals that are unique to a particular bus connector slot. These signals include MREQx*, MAKx*, DREQx*, DAKx*, IRQx and AENx. The x corresponds to the slot number, from 1 to 14 , into which the board is inserted.

Slot 0 The second to the left-most slot in an STD 32 card cage. This slot is reserved for the Permanent Master.

Slot X The left-most slot in an STD 32 card cage in which the Permanent Master or simple arbiter resides while it receives and drives slot-specific signals.

STD bus The collection of signals defined on P to support 8-bit transfers between masters and slaves.

STD 32 bus The extension to the STD bus to support 16- and 32-bit transfers, backplane DMA, bus arbitration and slot specific configuration.

STD-80 bus STD bus signals that are compatible with Intel 80XX series microprocessors.

Temporary Master Temporary Masters request bus ownership from the Permanent Master via either BUSRQ* or MREQx*. Ownership is granted after arbitration. Temporary Masters who receive ownership manage transfers on the bus.

unpacking Unpacking refers to the expansion of a multiple-byte piece of information into smaller data widths that are to be transferred over multiple cycles.

word A group of two bytes (16-bits) ranging in value from 0 to 65,536.

GLOSSARY

INDEX

- A -

- B -

- C -

- D -

- E -

- F -

- G -

- H -

- I -

- L -

- M -

- N -

- O -

- P -

- *R* -

- *S* -

- T -

- U -

- W -