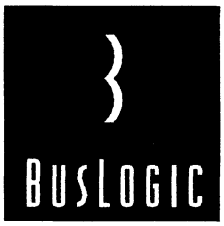
Technical Reference Manual



SCSI HOST ADAPTERS FOR PCI, VESA, EISA, AND ISA SYSTEMS

Revision History

Revision	Change Activity	Date
A	Release	6/1/94

Compliance Statements

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operations.

Declaration of the Manufacturer/Importer

We hereby certify that the SCSI Host Adapter for PCI, VESA, EISA and ISA, in compliance with the requirements of BMPT Vfg 243/1991, is RFI suppressed. The normal operation of some equipment (e.g., signal generators) may be subject to specific restrictions. Please observe the notices in the user's manual.

The marketing and sale of the equipment was reported to the Federal Office for Telecommunication Permits (BZT). The right to retest this equipment to verify compliance with the regulation was given to the ZZF.

Bescheinungung des Herstellers/Importeurs

Hiermit wird bescheinigt, das SCSI Host Adapter for PCI, VESA, EISA and ISA in Uebereinstimmung mit den Bestimmungen der BMPT-AmtsblVfg 243/1991 funk-entstört ist. Der vorschriftsmässige Betreib mancher Geräte (z. B. Messender) kann allerdings gewissen Einschränkungen unterliegen. Beachten Sie deshalb die Hinweise in der Bedienungsanleitung.

Dem Bundesamt für Zulassungen in der Telekommunikation wurde das Inverkehrbringen dieses Gerätes angezeigt und die Berechtigung zur Ueberprüfung der Serie auf die Einhaltung der Bestimmungen eingeräumt.

EN 55 022 Declaration of Conformance

This is to certify that BusLogic SCSI Host Adapters are shielded against the generation of radio interference in accordance with the application of Council Directive 89/336/EEC, Article 4a. Conformity is declared by the application of EN 55 022:1987 Class B (CISPR 22:1985/BS 6527:1988).

Copyright

Trademarks

Company and product names are trademarks or registered trademarks of their respective companies.

Table of Contents

Preface

The BusLogic SCSI Host Adapter Technical Reference Manual is intended for system software programmers desiring to develop device driver software for BusLogic's family of SCSI adapter products.

This manual covers adapter architecture, hardware and software operation and includes electrical signaling information for each bus architecture.

Products covered in this document include:

- BT-946C PCI Fast SCSI Host Adapter (10 MBytes/sec synchronous data transfer rate; no floppy connector)
- BT-440/445C VESA Fast SCSI Host Adapter (10 MBytes/sec synchronous data transfer rate; BT-440C has no floppy connector)
- BT-746/747C EISA Fast SCSI Host Adapter (10 MBytes/sec synchronous data transfer rate; BT-746C has no floppy connector)
- BT-545C PC/AT Fast SCSI Host Adapter (10 MBytes/sec synchronous data transfer rate)
- BT-540CF PC/AT Fast SCSI Host Adapter (with Centronics external SCSI connector)

Contents

This manual is organized in the following manner:

Part 1: Adapter Operation

- Section 1 contains an overview to BusLogic's MultiMaster™ technology.
- Section 2 details adapter hardware components and operation.
- Section 3 discusses software operation between the host adapter and the host system.
- Section 4 describes the adapter's SCSI electrical interface.
- Section 5 describes the adapter's floppy controller interface.
- Section 6 details adapter internal diagnostics.
- Appendix A is a list of industry acronyms.

Part 2: PCI Host Adapters

- Section 1 includes descriptions and specifications for the BT-946C host adapter.
- Section 2 describes the electrical interface for the BT-946C host adapter.
- Section 3 contains the PCI bus timing diagrams.
- Section 4 details the configuration for non-conforming PCI motherboards.

Part 3: VESA Host Adapters

- Section 1 includes descriptions and specifications for the BT-440/445C host adapter.
- Section 2 describes the electrical interface for the BT-440/445C host adapter.
- Section 3 contains the VL-Bus timing descriptions.

Part 4: EISA Host Adapters

- Section 1 includes descriptions and specifications for the BT-746/747C host adapter.
- Section 2 describes the electrical interface for EISA host adapters.
- Section 3 contains the EISA bus master timing diagrams.

Part 5: ISA Host Adapters

- Section 1 includes descriptions and specifications for the BT-540CF and the BT-545C host adapters.
- Section 2 describes the electrical interface for ISA host adapters.
- Section 3 contains the ISA bus master timing diagrams.

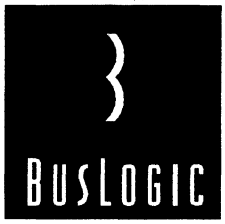
Related Documentation

Refer to the user's guide provided with your BusLogic host adapter for details on host adapter installation and configuration.

Notational Conventions

The following conventions are used throughout this manual:

Convention	Description
UPPERCASE	Used to indicate the names of keys.
-	A hyphen indicates an active low signal.
+	A plus sign indicates an active high signal.



PART 1

ADAPTER OPERATION

Part 1: Contents

Section 1•1: BusLogic MultiMastering Technology

Section 1•2: Hardware Description

Section 1•3: Software Interface

List of Figures

Section 1•1: BusLogic MultiMastering Technology

Section 1•2: Hardware Description

Section 1•3: Software Interface

Section 1•4: SCSI Electrical Interface

List of Tables

1•1

BusLogic MultiMastering Technology

Introduction

Bus mastering data transfers are commonly used by intelligent I/O controllers to increase system performance by off-loading the low level control tasks to a dedicated micro-controller. BusLogic has created a unique technology for implementing a diverse family of high performance bus mastering SCSI host adapters while maintaining a high degree of compatibility among them. This technology is called MultiMaster™ and is implemented in a single ASIC. The MultiMaster technology is utilized on all BusLogic bus mastering adapters allowing them to share common device drivers, firmware and BIOS. The following describes the key features, capabilities and benefits of BusLogic MultiMaster technology.

Bus Mastering DMA

Bus mastering DMA is a form of information transfer between two devices. The advantages of bus mastering in a PC environment are that it requires less host CPU time and offers high performance data transfers. An intelligent bus master device will use a micro-controller or micro-processor to perform complex I/O operations that would otherwise require interaction from the host system CPU. BusLogic intelligent SCSI controllers utilize a 16-bit micro-controller to off-load control of information transfers, thus freeing the system CPU for other tasks and increasing overall system performance as well as I/O performance.

BusLogic's MultiMaster technology allows the use of a single ASIC as the interface to all of our bus mastering adapters. The MultiMaster ASIC contains all the logic required to fully automate a bus master transfer on five different system buses, ISA, EISA (EMB), VL-Bus, PCI and MCA. This technology enables a common high performance microprocessor-based architecture to be applied to many system platforms. It allows the use of a single device driver per operating system to support all BusLogic bus mastering SCSI controllers.

Description

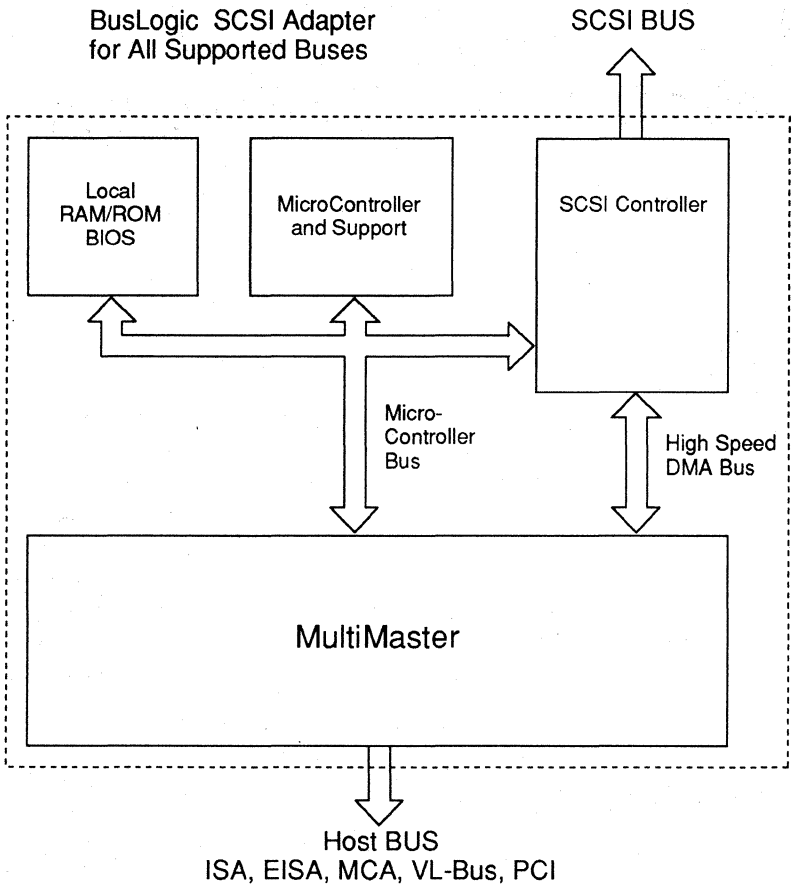


Figure 1-1. Typical MultiMaster SCSI Adapter

MultiMaster provides a complete register and interrupt interface between the host system and the host adapter microcontroller. SCSI command overhead is reduced by reading and updating mailboxes in host memory without intervention by the host adapter microcontroller. Data transfers between the SCSI controller and host system memory are under complete control of the MultiMaster ASIC, from system bus arbitration to the completion of data transfer.

Configuration

The MultiMaster ASIC is configured to interface with a particular system bus by hard wiring three mode select pins. Each host adapter has these pins tied to logic levels representing the host bus the adapter interfaces with. The ASIC contains the bus interface protocols required for each supported system bus. By embedding all the bus specific protocols in the MultiMaster ASIC the microcontroller and firmware design issues become independent of the target host platform.

Registers within the MultiMaster ASIC are configured at power-up to determine host adapter operating parameters.

In an MCA system the MultiMaster ASIC monitors the CDSETUP and address signals to allow the configuration registers to be read and written by the host system. CDSETUP is unique for each slot on the MCA bus allowing MCA adapter cards to use the same set of I/O addresses for configuration registers and avoid bus contention problems. During power-on initialization the host system BIOS reads an

adapter ID from the MultiMaster configuration registers. The ID field is used to look up a set of operating parameters previously stored in the system CMOS memory. If no match is found for the ID field the system will invoke a configuration program allowing the operating parameters for this adapter card to be defined and stored for future power on initializations.

EISA compatible adapter cards use I/O addresses 0zC80h - 0zC84h for configuration purposes. The EISA motherboard generates I/O cycles to these addresses on a per slot basis by only activating the AENz (z represents slot number) signal for the desired slot. This allows each EISA compatible add-in card to decode the same register address range for configuration purposes. Software running on the system sees each card's I/O register at unique addresses.

PCI is very different from MCA or EISA during power-on initialization. The PCI system BIOS will scan the system bus and assign I/O port address, BIOS memory address and interrupt level dynamically each time the system is powered up. This fully automatic initialization is also supported by the BusLogic MultiMaster ASIC.

In an environment such as the ISA and VL-Bus, boot time configuration is not performed by the host system. In these environments the host adapter microcontroller will configure the MultiMaster based on information read from switches or stored in EEPROM (BusLogic Model C boards only).

The MultiMaster ASIC maintains a high level of isolation between the host environment and adapter firmware, allowing a common architecture to be applied to all BusLogic bus mastering SCSI host adapters.

Bus Master Transfers

There are three basic phases of a bus master transfer: arbitration, where the bus master requests and gains access to the system bus; data transfer, in which the bus master has control of the bus and is actively transferring data across the bus. Data transfer is followed by bus release once all available data has been transferred or bus release is being forced to allow another device access or a system memory refresh cycle. Although signals and timing vary from one bus type to another, our MultiMaster technology fully automates the data transfer process.

Arbitration

Once a DMA transfer has been programmed into the MultiMaster by the local microprocessor, the resulting bus master transfer is completely automated by the MultiMaster ASIC. Bus arbitration is initiated by asserting the DMA request or bus request signal appropriate to the particular environment the MultiMaster is residing in. The host system then responds with an acknowledge signal granting access to the system bus. The MultiMaster then asserts a host bus signal to indicate it will be acting as a bus master. The host adapter is now the active bus master and has complete control over the system bus. In an ISA system the MultiMaster arbitrates for the system bus by asserting the DMA request line and waiting for a DMA acknowledge.

Data Transfer

Data transfer begins after the host system has granted the host adapter access to the system bus. The MASTER signal is then asserted by the MultiMaster to inform the host system it is acting as a bus master device. The host adapter is now the active bus master and controls the system address, memory read and memory write signals to perform the current data transfer. This continues until one of three things

happens: (1) the data transfer is complete, (2) the host adapter is unable to continue transferring due to an empty or full FIFO buffer or (3) the host system requires the bus to be released.

Bus Release

During the bus release phase the MultiMaster will relinquish control of the system bus through the appropriate protocol for the currently supported host bus. For example in an ISA system the MASTER signal is released and the address and data lines are tri-stated. DMA request is then released and the host system de-asserts the DMA acknowledge signal.

Preempt

The MultiMaster will release the host system bus when preempted. Most advanced system bus architectures have provisions for preemption of the current bus master. This allows other bus masters to share the system bus. This also is required to allow the host system to perform system memory refresh cycles. The ISA bus does not have a provision for preemption of a bus master. When configured for ISA bus operation the MultiMaster utilizes two timers to control the bus on and bus off times to control the amount of time the MultiMaster can remain as bus master of the system. The bus off time controls the amount of time the MultiMaster must release the system bus before arbitrating to become bus master again. This allows the system to periodically perform refresh cycles to the system memory and allows access to the system bus by other bus masters.

MultiMaster Advantages

MultiMaster technology offers several key advantages: performance, reliability and efficient product design cycles.

Performance

MultiMaster enhances the performance of BusLogic bus mastering SCSI products in several ways. The MultiMaster ASIC automatically retrieves mailbox entries and associated commands from the host system memory in a single operation. This greatly reduces the command overhead associated with maintaining the mailbox interface. Scatter/gather operations are streamlined under control of the MultiMaster. The microcontroller is able to pre-initialize the next segment of a scatter/gather transfer while the current segment is being transferred. This maintains the continuous flow of data during scatter/gather operations.

All BusLogic SCSI bus mastering host adapters utilize the same mailbox structure interface. This mailbox interface allows the host system to perform multitasking I/O operations with minimum overhead. Mailboxes are located in system memory. The mailbox structure allows up to 255 independent I/O commands to be processed by the SCSI adapter simultaneously. The MultiMaster will automatically scan the mailboxes for a valid entry. Once a valid entry is found the MultiMaster will automatically read the mailbox and store the associated CCB (Command Control Block) in the host adapter local RAM and clears the mailbox semaphore. The advanced features of BusLogic MultiMaster ASIC coupled with a 16-bit microcontroller resident on all BusLogic bus mastering SCSI adapters provides high performance and reliability.

BusLogic bus mastering host adapters are able to perform an entire I/O operation without intervention from the host CPU. By automating the data transfer between the system memory and utilizing a dedicated microcontroller to handle the high level control of the operation, BusLogic host adapters deliver superior performance over PIO types of interface cards.

All of these features combine to deliver the user the best possible I/O performance. The MultiMaster ASIC allows this high performance architecture to be implemented on five different host system buses without any significant design changes.

Common OS Drivers/BIOS/Firmware

The MultiMaster ASIC separates the SCSI, interface microcontroller, and firmware from the system bus allowing one design to be applied to all system platforms.

MultiMaster's common system interface allows a single device driver per operating system to support BusLogic's host adapters. Only one device driver needs to be tested, qualified and embedded into the operating system kernel to support all the BusLogic bus mastering adapters. All BusLogic bus mastering SCSI adapters are also supported by common BIOS source code.

MultiMaster provides the isolation from the host system required for a common architecture to be applied to many system platforms. This common architecture allows the use of common firmware across the entire bus mastering host adapter product line.

1•2 Hardware Description

BusLogic host adapters are based on a BusLogic-designed, MultiMaster ASIC technology, offering high-performance interconnection between the bus architecture and Small Computer System Interface (SCSI) peripheral devices. A BusLogic-designed bus master controller ASIC, an advanced SCSI controller chip and a 16-bit microprocessor chip provide higher speed, lower power consumption, fewer parts and higher reliability.

Both internal and external 50-pin connectors are included on the board for flexibility in attaching SCSI devices to the system.

This section describes the adapter's functional hardware operation. It covers the hardware control registers that are mapped into the system's I/O address space. It also describes command use, data flow and hardware management.

Adapter Architecture

The host adapter plugs into a host system and supports the attachment of internal SCSI drives or the connection to external SCSI peripheral devices in add-on enclosures. The system architecture is illustrated in Figure 1-2.

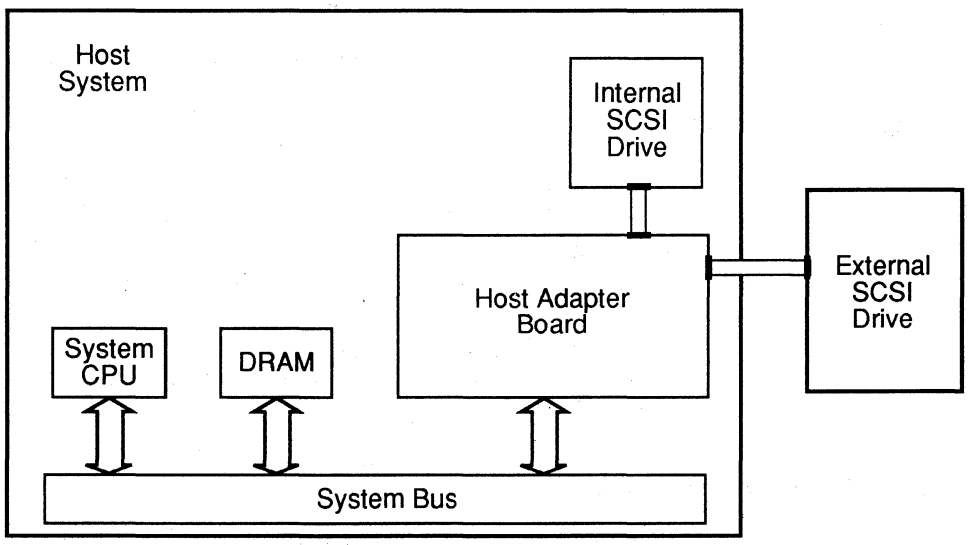
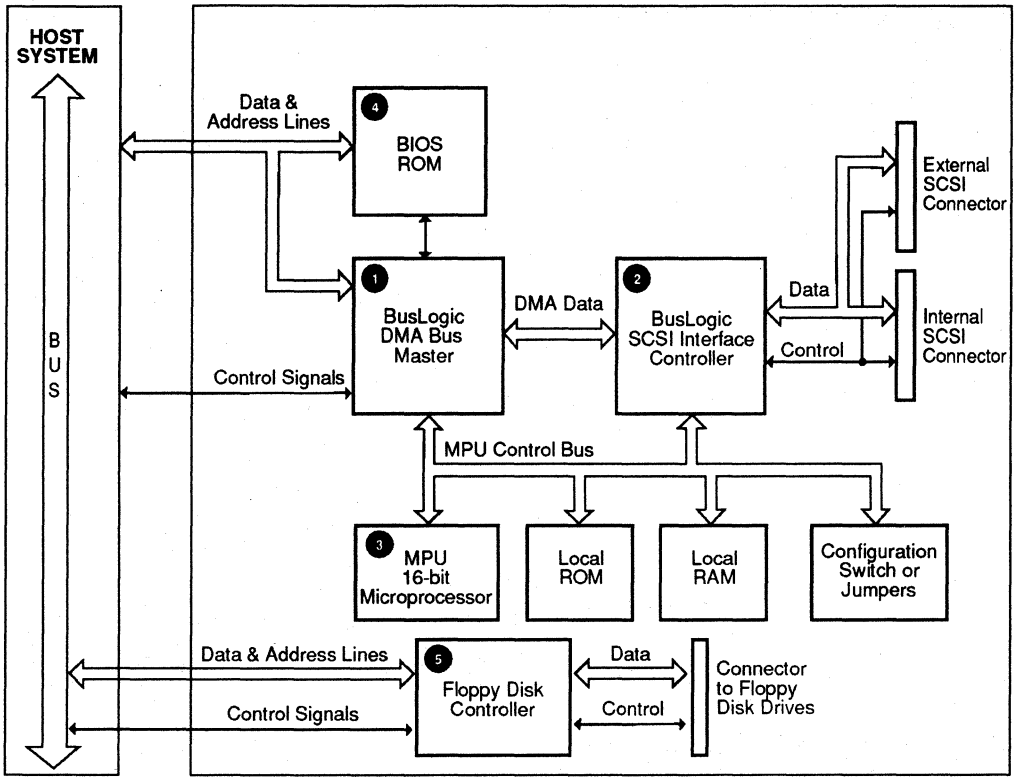


Figure 1-2. System Architecture

Figure 1-3 is a functional block diagram of the BusLogic adapter. The paragraphs that follow describe the numbered components in the figure.



Notes:

1. Not all models have a floppy connector.
2. On some models BIOS exists in local ROM.
3. Depending on the model, hardware configuration may be done via a switch or jumpers.

Figure 1-3. Host Adapter Architecture

Bus Master DMA Controller

All host bus interface logic is provided on the board by a BusLogic designed bus master ASIC ❶. This chip provides bus master capabilities which greatly reduce the involvement of the host system's CPU in I/O control and data transfer activities. Under control of this chip, 32-bit bus master transfers at up to 132 MBytes/sec to and from the main system memory are possible with the use of its internal 128-byte FIFO. A true multitasking mailbox structure supports up to 255 tasks. This performance and improved bus utilization significantly enhances multitasking and multi-user applications.

Advanced SCSI Controller

On-board control of the interface to SCSI peripheral devices is provided by another ASIC, the BusLogic SCSI controller chip ❷. Up to 10 MBytes/sec synchronous and 7 MBytes/sec asynchronous 8-bit SCSI data transfers are supported by the SCSI interface controller. This low-power, high-performance CMOS component completely conforms to the ANSI standard, X3.131-1986 for the Small Computer System Interface. The chip reduces protocol overhead by performing common SCSI algorithms or sequences in response to a single host system command.

Microprocessor Unit (MPU)

An on-board, 16-bit Intel-based 8018X microprocessor unit (MPU) ❸ coordinates all activity on the host adapter under the direction of the board's local ROM. Consequently, the on-board MPU orchestrates such activities as the initialization, command decoding, interrupt generation and the control of the data flow among the board's components.

Local BIOS ROM

The host adapter can be used in place of or in conjunction with a standard hard disk controller. The host adapter's on-board local BIOS ❹ provides a compatible method of attaching a SCSI hard drive to a system just as any other type of hard disk is connected. The host adapter's BIOS intercepts each host software interrupt that requests a disk I/O service and manages these interrupts according to the address of the requested drive. If the designated drive is a disk assigned to the system's internal disk controller, the host adapter's BIOS passes the command on to that disk controller. If the designated disk is one of the SCSI disks attached to the host adapter, the host adapter's BIOS responds to the request and instructs the host adapter to execute the command.

Floppy Disk Controller

On models with a floppy controller, a floppy disk controller chip ❺ provides support for any combination of up to two 3.5" and 5.25" floppy disk drives with 250 KBytes/sec, 300 KBytes/sec and 500 KBytes/sec data transfer rates. On Model C boards, it can also support the newly emerging 2.88 MB floppies with transfer rates of 1 MBit/sec. This floppy controller circuitry is completely independent from the SCSI logic and is accessed by the system BIOS floppy diskette routines.

Host DMA Data Transfer Control

ISA

A typical PC/AT system provides seven direct memory access (DMA) channels. One of the DMA Channels 5, 6, or 7 can be used by the ISA host adapter as defined by the on-board configuration. Channel 5 is the default DMA channel. Whenever the ISA host adapter must transfer 16-bit data to or from memory, it performs a priority arbitration process with other system DMA channels to obtain permission to use the system bus. It asserts its DMA Request signal (Default DRQ5) and waits for the corresponding DMA Acknowledge signal (Default DACK5) to arbitrate for the bus. Once it obtains ownership of the bus, it asserts the -MASTER signal and operates as a bus master. It then transfers data across the bus directly into or out of the main system memory until the Time On Bus period has expired or the transfer has been completed, whichever occurs first. If the transfer was not completed, the ISA host adapter will wait for the specified Time Off Bus period and then will arbitrate for bus access to resume the data transfer.

EISA

A typical EISA system provides up to six EISA bus master (-MREQx) slots. Each EISA slot is also ISA compatible and provides hardware pins to connect to the ISA DMA channel. Because EISA bus master transfers are faster and more efficient than ISA DMA transfers the EISA host adapter assumes the role of a 32-bit EISA bus master for all host data transfers.

VESA

All bus master mode data transfers, including mailboxes, CCBs and SCSI peripherals data are performed via the 32-bit, high-speed VL-Bus. Whenever the VESA host adapter has to transfer this data, it performs an arbitration process to obtain permission to use the VL-Bus. It asserts its slot-specific VL-Bus request signal (-LREQ) and waits for the slot-specific VL-Bus grant signal (-LGNT). Once -LGNT is asserted, the board is granted ownership of the VL-Bus. It then transfers data across the VL-Bus directly into or out of the host memory until the allowed Time On Bus period has expired, or the transfer has been completed; whichever occurs first. If the transfer was not completed, the VESA host adapter will arbitrate for bus access again to resume the data transfer.

PCI

The DMA control logic manages bus arbitration and data transfer coordination. The BT-946C operates as a PCI bus master during data transfers. The BT-946C arbitrates for PCI bus access and, once granted, it takes over control of the bus. It generates the PCI bus address and command strobes. The BT-946C supports both odd and even starting addresses, commonly known as aligned and unaligned transfers. If presented with an even transfer count beginning at an odd memory starting address, the BT-946C will first transfer a single byte (data bits D24-D31). The remaining data is then transferred as double words (32 bits) until the last byte, which is transferred as a single byte (data bits D0-D7). While odd byte transfers are fully supported, it is recommended that when possible, PCI host buffers be double-word aligned to gain better data transfer performance.

Hardware Registers

The host adapter's I/O interface consists of three hardware registers that are used by the host to issue start commands to the host adapter, to gain status information about the adapter's operation, and to manage interrupts generated by the host adapter. These registers are located in the I/O address space at three consecutive addresses. The beginning or base address is determined by the I/O base address switch settings described in the host adapter user's guide. Table 1-1 provides a summary of these 8-bit wide registers.

Table 1-1. Host Adapter Hardware Registers

Address	Type	Description
I/O Base Address + 0	W	Control Register
I/O Base Address + 0	R	Status Register
I/O Base Address + 1	W	Command/Parameter Register
I/O Base Address + 1	R	Data In Register
I/O Base Address + 2	R	Interrupt Register

Control Register (Write Only) I/O Base Address + 0

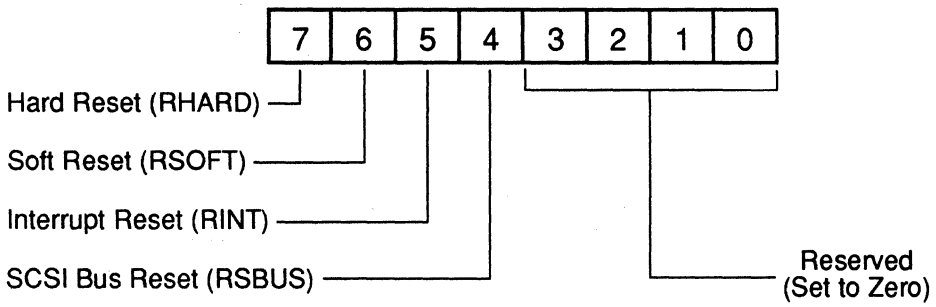


Figure 1-4. Control Register (Write)

The host system's CPU uses this register to specify programmable options within the host adapter (e.g., soft reset, hard reset, SCSI bus reset). For example, the host system's CPU can stop all host adapter activity immediately by setting this register's Bit 6, the Soft Reset bit (RSOFT). This is a write-only register.

Bits 0-3–These bits are reserved and must be set to zero.

Bit 4 Reset SCSI Bus (RSBUS)–When this bit is set to a one, the Reset signal on the SCSI bus is maintained true for at least 25 microseconds. This Reset condition immediately clears all SCSI devices from the bus. The assertion of the SCSI Reset condition supersedes all other activity on the SCSI bus. See the heading "Reset Operations" later in this section for additional details on the Reset operation.

Bit 5 Reset Interrupt (RINT)–The system's CPU sets this bit to acknowledge a host adapter interrupt. When this bit is set to one by the CPU, the host adapter's hardware-generated interrupt is reset and the Interrupt Register is cleared. Note that all bits in the Interrupt Register are cleared by setting this bit.

Bit 6 Soft Reset (RSOFT)–When this bit is set to one, it causes all host adapter activity to stop immediately. All mailboxes, command control blocks, and any pending commands are discarded by the host adapter. Previous mailbox pointers must be cleared by host processes. The primary difference between a hard and soft reset is that this bit does not effect a SCSI Bus Reset condition. Once the soft reset activity has been completed by the host adapter, the host adapter must be reinitialized for any future operation. This state is indicated by the setting of the Host Adapter Ready bit (HARDY) and the Initialization Required bit (INREQ) in the Status Register.

Bit 7 Hard Reset (RHARD)–The setting of this bit causes the host adapter to enter an initial condition power-on state. Any command in process is stopped and pending commands are abandoned. The host adapter will execute its internal diagnostic function and report any errors. During reset, the Diagnostic Active bit (DACT) is set. Once the hard reset activity has been completed by the host adapter, the host adapter must be reinitialized for any future operation. This state is indicated by the setting of the Host Adapter Ready bit (HARDY) and the Initialization Required bit (INREQ) in the Status Register.

Specific operation bits in the Control Register are automatically reset by the host adapter when the specific operation is completed. The host-controlling software is not required to reset the operational control bit for a specific function.

Status Register (Read Only) I/O Base Address + 0

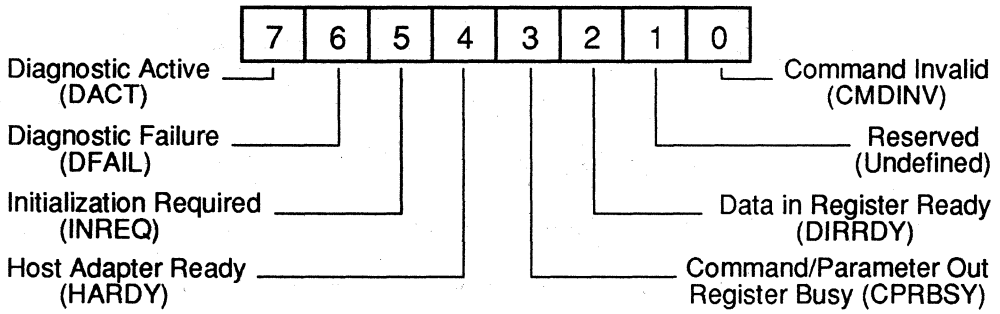


Figure 1-5. Status Register (Read)

The host adapter uses this register to report the status of its condition to the system's CPU. For example, the host adapter sets this register's Data In Register Ready bit (DIRRDY) when it has written data to its Data In Register. By setting this bit, the host adapter notifies the host system's CPU that there is fresh data in the Data In Register that should be read. As soon as the host system's CPU reads the Data In Register, the host adapter immediately resets the Data In Register Ready bit (DIRRDY). The host adapter resets this bit to zero to ensure that the host system's CPU does not reread the same data in the Data In Register. When the host adapter writes fresh data to its Data In Register, it will then set the Data In Register Ready bit (DIRRDY) to one again.

The host system's CPU can also read this register to check for error conditions. For example, the host adapter sets this register's Command Invalid bit (CMDINV) when it detects an invalid command or parameter byte in its Command/Parameter Register. Consequently, the host system's CPU can check the Command Invalid bit (CMDINV) to see if such an error condition currently exists. This is a read-only register.

Bit 0 Command Invalid (CMDINV)–The host adapter sets this bit immediately upon detection of an invalid command or parameter byte in the Command/Parameter Register. When the host is sending a single or multibyte command, the host adapter terminates the data transfer sequence by setting the Command Complete bit (CMDC) in the Interrupt Register and by generating a hardware interrupt. The host adapter terminates all commands (valid or invalid) by this method. Invalid commands are indicated by the setting of this bit. The condition of this bit is only meaningful while the Command Complete bit (CMDC) is true.

Bit 1 This bit is reserved and is set to zero.

Bit 2 Data In Register Ready (DIRRDY)–This status bit is used to synchronize the transfer of status information from the host adapter to the host system. The host adapter sets this bit to one when it has placed a byte of data in the Data In Register. This condition notifies the host that it may read and process the data. When the host reads the data byte in the Data In Register, this bit is reset to zero by the host adapter. This sequence is repeated for multibyte data transfers.

Bit 3 Command/Parameter Register Busy (CPRBSY)–This status bit is used to synchronize the transfer of command and associated parameter bytes from the system host to the host adapter. When this bit is reset to zero the host may place a command or parameter byte in the Command/Parameter Register. When the host writes a byte to the Command/Parameter Register, the host adapter will set this

bit to a one indicating a Busy condition. The host adapter will reset this bit when it has read and processed the command/parameter byte. This sequence is repeated for multibyte data transfers.

Bit 4 Host Adapter Ready (HARDY)–This bit indicates the ready or not ready internal command state of the host adapter. When this bit is set, the host adapter is ready for a new host adapter command. In general the host system's processor may only issue host adapter commands while this bit is set.

Note: The multitasking design of the host adapter's firmware permits the following commands to be issued regardless of the busy or not-busy state of the host adapter.

- Start SCSI (02) command
- Enable Outgoing Mailbox Ready Interrupt (05) command.

See Section 1-3, "Software Interface," for details on these commands

Bit 5 Initialization Required (INREQ)–When this bit is set to one, it indicates that the mailbox structures must be initialized. This bit is typically set immediately after the completion of self-diagnostic tests following a reset. This bit is not set if diagnostics fail (the Diagnostic Failure bit (DFAIL) is set). The host system must now issue an Initialize Mailbox command (01) to inform the host adapter of the base memory address of the mailbox structure area. The host adapter will reset this bit after successful completion of the Initialize Mailbox command.

Bit 6 Diagnostic Failure (DFAIL)–When this bit is set, it indicates that the host adapter's internal self diagnostic has detected an error. This bit may be reset only by a hard reset initiated either by hardware or by the host software. See Section 1-6, "Internal Diagnostics," for more details.

Bit 7 Diagnostic Active (DACT)–This bit is set when the host adapter begins its self-testing activity immediately after a power-on reset or a programmed hard reset (Control Register, Bit 7 is set). This bit is reset upon the successful completion of the self-test activity. If the diagnostics fail, this bit may not be reset indicating that the self-test programs could not be completed. In the case of most failures, the Diagnostic Failure bit (DFAIL) will also be set. See Section 1-6, "Internal Diagnostics," for more details.

Command/Parameter Register (Write Only) I/O Base Address + 1

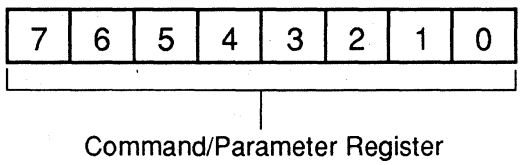


Figure 1-6. Command/Parameter Register (Write)

The Command/Parameter Register serves as the input port through which the host system software may issue commands and associated parameter bytes to the host adapter. The commands issued to the host adapter by this method provide initialization and establish control specifications for subsequent operations. SCSI-bus related commands are issued through a mailbox command structure (See Section 1-3, "Software Interface."). The host is responsible for issuing the correct number of command and parameter bytes for each operation. Otherwise, incorrect opera-

tion may occur along with the cessation of command acceptance by the host adapter. This condition is indicated by the setting of the Command Invalid bit (CMDINV) in the Status Register.

The coordination of command and parameter data byte transfers between the host adapter and the host system is governed by the Host Adapter Ready bit (HARDY), the Command/Parameter Register Busy bit (CPRBSY), and the Data In Register Ready bit (DIRRDY). All host adapter commands, with the exception of Enable OMBR Interrupt (05) and Start SCSI (02), require the Host Adapter Ready bit (HARDY) to be set. Parameter data bytes are written to the Command/Parameter Register. The host must first test the Command/Parameter Register Busy bit (CPRBSY) for a not set condition to determine if the host adapter is ready to accept a command parameter data byte. When the host writes a command or parameter byte to the Command/Parameter Register, the host adapter sets the Command/Parameter Register Busy bit (CPRBSY) to a one. When the local MPU has read the byte, the host adapter will reset the Command/Parameter Register Busy bit (CPRBSY) to indicate that the host can write another byte.

When all the bytes for a particular command have been transferred, and the command has been completed, the host adapter will set the Command Complete bit (CMDC) in the Interrupt Register. If an error is detected in the command, the Command Invalid bit (CMDINV) will be set in the Status Register.

Data In Register (Read Only) I/O Base Address + 1

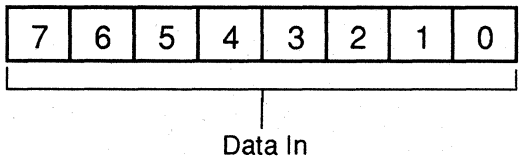


Figure 1-7. Data In Register (Read)

The host adapter uses this register to return information bytes to the host system. For commands that return information bytes to the host, the Data In Register Ready bit (DIRRDY) is used to synchronize data byte transfers. The host adapter sets the Data In Register Ready bit (DIRRDY) to a one when a data byte is available to be read by the host system. The Data In Register Ready bit (DIRRDY) is automatically reset to zero by the host adapter when the host system reads the data byte from the host adapter's Data In Register. For multiple data byte transfers, the host should wait for the Data In Register Ready bit (DIRRDY) to return to the set state before reading additional data. When the last byte of an information block (single or multiple byte) has been transferred, the host adapter will set the Command Complete bit (CMDC) in the Interrupt Register. This is a read-only register.

Interrupt Register (Read Only) I/O Base Address + 2

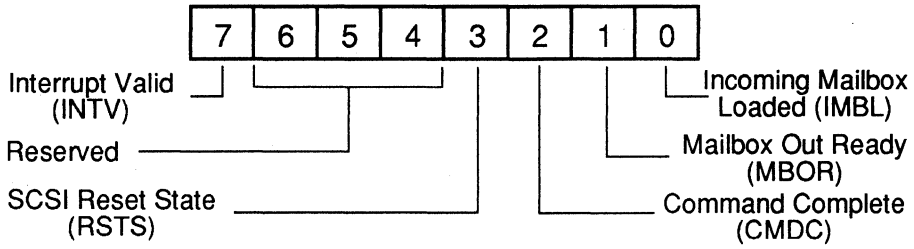


Figure 1-8. Interrupt Register (Read Only)

The host adapter uses this read-only register to tell the host system the reason why it generated a hardware interrupt signal. The following are the four conditions under which the host adapter can generate a hardware interrupt to the host system:

1. Incoming Mailbox Loaded interrupt (IMBL): the host adapter has made an entry in an incoming mailbox.
2. Mailbox Out Ready interrupt (MBOR): an outgoing mailbox location(s) is ready for the host system to use.
3. Command Complete interrupt (CMDC): the host adapter has completed a command.
4. SCSI Reset State interrupt (RSTS): the host adapter has detected a SCSI Bus Reset condition.

When the host adapter generates a hardware interrupt signal for one of the preceding four reasons, the host adapter sets bits in this register to provide the host system with more information about the interrupt. The host adapter completes the following actions:

1. It sets Bit 7, the Interrupt Valid bit (INTV), to indicate that the interrupt is valid.
2. It also sets one of the other unreserved bits to indicate why it generated a hardware interrupt signal to the host system. These bits (Bits 0-3) are collectively referred to as interrupt cause bits. This group of bits include the Incoming Mailbox Loaded bit (IMBL), the Mailbox Out Ready bit (MBOR), the Command Complete bit (CMDC), and the SCSI Reset State bit (RSTS).

In response to this interrupt request by the host adapter, the host system should execute the following sequence to service the interrupt:

1. Read the Interrupt Register. The host should maintain this value internally for further interrupt processing.
2. Clear the Interrupt Register. This is accomplished by setting the host adapter's Control Register's Reset Interrupt bit (RINT).
3. Determine the interrupt cause (from the saved Interrupt Register value in the preceding Step 1) and then execute the appropriate interrupt service routine.

Note: A hard reset, a soft reset, or initial board power-on condition will cause the Interrupt Register to be cleared, i.e., no interrupts pending.

The host adapter sets the priority of certain interrupt conditions. This topic will be discussed before describing the individual bits of the Interrupt Register. Refer to the following heading "Interrupt Timing and Synchronization."

Interrupt Timing and Synchronization. The host adapter sets the priority of certain interrupt conditions. The posting of a mailbox-related interrupt is withheld if a SCSI Reset State interrupt (RSTS) or a Command Complete interrupt (CMDC) is pending service from the host. Once the SCSI Reset State interrupt (RSTS) or the Command Complete interrupt (CMDC) is cleared, the host adapter will post the mailbox interrupt(s). Likewise, a SCSI Reset State interrupt (RSTS) or a Command Complete interrupt (CMDC) will only be presented if the Interrupt Register has been cleared, and the Data In Ready bit (DIRRDY) shows no additional data is pending.

Because all outbound host mailboxes are typically controlled by host-resident software, it is not necessary to enable the Outgoing Mailbox Ready interrupt (OMBR) unless all mailboxes are being utilized. This technique also lessens the possibility of missed interrupt notification for an Incoming Mailbox Loaded (IMBL) condition. If all outgoing mailboxes are in use, then the host could enable the Outgoing Mailboxes Ready interrupt (OMBR) to gain notification of a Mailbox Ready condition. See the Enable Outgoing Mailbox Ready command description in Section 1-3, "Software Interface," for instructions on how to enable the Outgoing Mailboxes interrupts (OMBR).

Bit 0 Incoming Mailbox Loaded (IMBL)–When this bit is set, it indicates that the host adapter has made an entry in an incoming mailbox location. This bit is qualified by the Interrupt Valid bit (INTV) also being set and by the generation of a bus interrupt signal. The host should service this interrupt as soon as possible to allow additional host adapter interrupts to be posted. The multitasking firmware of the host adapter continues to process outstanding SCSI commands after the posting of an Incoming Mailbox Loaded interrupt (IMBL). If additional outstanding commands are completed before the servicing of a previous Incoming Mailbox Loaded interrupt (IMBL), the status of completed commands will be placed in an available incoming mailbox location. The host should therefore scan all mailboxes to determine if additional data has been provided. The host adapter will use Incoming Mailbox locations in a round-robin order permitting the host to scan in the same manner. When a vacant mailbox is found the host may discontinue its scan.

Bit 1 Outgoing Mailbox Ready (OMBR)–When this bit is set to a one, it indicates that one or more of the outgoing mailbox locations is available for use by the host. This bit is qualified by the Interrupt Valid bit (INTV) also being set and by the generation of a bus interrupt signal. The Outgoing Mailboxes Ready interrupt (OMBR) is generated only when Outgoing Mailbox interrupts (OMBR) have been enabled and an outgoing mailbox entry is cleared by the host adapter. An Outgoing Mailbox Ready interrupt (OMBR) is suppressed if a SCSI Reset State (RSTS) or a Command Complete (CMDC) interrupt is pending service. When these previous interrupts are cleared by the host, the pending Outgoing Mailboxes Ready interrupt (OMBR) will be issued by the host adapter.

Application Note: It is recommended that the Outgoing Mailbox Ready interrupt (OMBR) not be enabled unless all outgoing mailbox locations are in use. For most applications, the host adapter will process command requests faster than a host will issue them. If the situation does occur where all outgoing mailbox locations are busy, the host may issue the Enable Outgoing Mailbox Ready Interrupt command without waiting for the status of the Host Adapter Ready bit (HARDY).

Bit 2 Command Complete (CMDC)–This bit is set to a one when the Command/Parameter Register is ready to accept a command. Any previous command will have been completed, either normally or abnormally. If a previous command completed with an error condition or was aborted for any reason, this bit will still be set along with the Command Invalid bit (CMDINV) in the Status Register. Normally completed commands are indicated by the setting of this bit without any accompanying error condition status bits. This bit is qualified by the Interrupt Valid bit (INTV) also being set and by the generation of a bus interrupt signal. A Command Complete interrupt (CMDC) is suppressed if an Interrupt Valid bit (INTV) is set (indicating an interrupt is pending service) or if the Data In Register Ready bit (DIRRDY) is set. When these previous interrupts are cleared by the host, the temporarily withheld Command Complete interrupt (CMDC) will be issued by the host adapter.

Bit 3 SCSI Reset State (RSTS)–When this bit is set, it indicates that a SCSI Bus Reset condition has been detected by the host adapter. This bit is qualified by the Interrupt Valid bit (INTV) also being set and by the generation of a bus hardware interrupt signal. In cooperation with host system driver software, the host adapter can implement the SCSI specification soft reset option. host adapter queued operations will resume once the SCSI bus has returned to the operational state. If a currently running command was aborted due to the SCSI Bus Reset condition, the host adapter may have to restart the command.

The host may convert the SCSI bus soft reset to a SCSI bus hard reset by setting the Soft Reset bit (RSOFT) in the host adapter's Control Register. In this case, all queued commands are abandoned and the host adapter must be reinitialized. See the heading "Reset Operations" later in this section for details on the reset conditions.

Bit 4 Reserved–Value read is zero.

Bit 5 Reserved–Value read is zero.

Bit 6 Reserved–Value read is zero.

Bit 7 Interrupt Valid (INTV)–When this bit is set, it indicates that a valid interrupt has been generated by the host adapter. This bit reflects the state of the host adapter generated interrupt signal on the bus. The specific reason for the interrupt condition is determined by Bits 0-3 of this register.

Reset Operations

Host adapter reset conditions are initiated from two different vantage points: the host system and the SCSI bus. A description of each follows.

Host-Initiated Reset Operations

The host system may reset the host adapter to an initial power-on condition through two different operations, one system and the other software.

System Reset–The host adapter is fully reset and initialized to a power-on initial condition when the Reset signal is true on the host system bus. The Reset signal is asserted to a true condition by the host (1) during power on, (2) host detected low-power conditions, or (3) a user-invoked reset switch reset. The Reset signal is applied universally to all installed host adapters in the system bus. A host adapter system reset causes a power-on initialization process which also results in a reset of all the devices on the SCSI bus.

Software Reset–The host adapter may be fully reset to an initial state by a software command, just the same as if a system reset had been received by setting the Hard Reset bit (RHARD). When RHARD is set to true, a Reset condition occurs immediately. However, this Reset condition will only affect *one* host adapter, unlike the system reset (described above) which resets *all* host adapters installed in the system bus. An RHARD reset may issue a subsequent reset to the SCSI device depending on the AutoSCSI utility configuration setting for the **Enable SCSI Bus Reset** option (see the manual for your host adapter for more information).

Either type of host-initiated reset will cause the following conditions on the host adapter.

- The control registers of all intelligent logic modules on the host adapter will be initialized to a known state.
- All pending operations are aborted and all data structures are initialized to a no operation pending state.
- The host adapter executes all internal diagnostic functions. While the diagnostic functions are in process, the host adapter will indicate this condition by setting the Diagnostic Active bit (DACT) to true in the Status Register.
- During a system reset, the host adapter places a SCSI Bus Reset condition on the SCSI bus. This may also occur during a software reset (using the RHARD bit), but depends on the AutoSCSI utility configuration setting for the **Enable SCSI Bus Reset** option. This will reset all peripheral devices, whether a target or initiator.
- After completion of a system reset, the host adapter indicates that it is now in an initial condition by asserting the Initialization Required bit (INREQ) in the host adapter's Status Register. This condition requires that all mailbox, command control blocks, and host adapter operation parameters be established before operations may begin.

Note: *The host may intiate a soft reset by setting the RSOFT bit. This reset reinitializes the specific host adapter, but there is no subsequent SCSI bus reset.*

SCSI Bus Reset Operations

The SCSI Bus Reset condition is used to clear all SCSI devices immediately from the SCSI bus. When the SCSI bus Reset signal is asserted, the SCSI Bus Reset condition takes precedence over all other bus phases. A SCSI Bus Reset condition may be forced by any device on the bus, whether a target or initiator. Whenever a SCSI Reset condition occurs, a Bus Free phase always follows the Reset condition.

The five ways in which a SCSI bus reset may be either asserted or sensed by the host adapter are as follows:

R1 The SCSI Bus Reset condition is always asserted when the host adapter is reset by the host system (host system **system** reset only). This is described earlier in this section under the heading "Host-Initiated Reset Operation."

R2 The SCSI Bus Reset condition may be asserted when the host adapter is **soft** reset by the host system when RHARD is set to true (depending on the AutoSCSI utility configuration setting for the **Enable SCSI Bus Reset** option). This is also described under the heading "Host-Initiated Reset Operations."

R3 The SCSI Bus Reset condition is asserted if the Reset SCSI Bus bit (RSBUS) is set by the host system control software.

R4 The host adapter may initiate a SCSI Bus Reset condition in reaction to a detected bus phase error. The host adapter constantly monitors the SCSI bus for invalid conditions. If an invalid phase is detected, the host adapter will perform a normal SCSI Bus Reset operation which includes the assertion of the SCSI bus Reset signal.

R5 The host adapter will detect and respond to a SCSI Bus Reset condition that is asserted by another device on the bus. Other SCSI devices may normally assert the Reset signal during either initialization or certain error recovery states. The device driver requires an acknowledgement from the host adapter that a SCSI Reset operation has taken place, so the SCSI Reset State bit is set in the interrupt register.

While the SCSI specification defines two methods by which the SCSI bus may be reset, either by the hard reset option or the soft reset option, *BusLogic host adapters only support the hard reset option.*

The SCSI bus hard reset option restores ALL SCSI devices, target or initiator, to the initial power-on condition. All system activity is lost, and all devices must be completely reinitialized before normal operations may be restored. SCSI devices that implement the hard reset option perform the following operations:

- Clear all uncompleted commands.
- Release all SCSI device reservations.
- Return any SCSI device operating modes to their default condition.

Here is how the host adapter implements a SCSI hard reset:

- When the host adapter detects a SCSI reset, it issues an interrupt indicating a SCSI reset has occurred to the host system.
- The host adapter sets both the SCSI Reset State bit (RSTS) and the Interrupt Valid bit (INTV) in the host adapter's Interrupt Register.
- All host adapter command control blocks (CCBs) are abandoned and the host adapter readies itself to accept new initialization commands.
- BusLogic recommends that after the device driver receives the RSTS, it issue a RHARD or an RSOFT to ensure that the host adapter is re-initialized.
- All the mailbox and command control blocks must be reinitialized.

1•3

Software Interface

For the host adapter to operate properly, the host system must issue the correct command and associated parameters to the host adapter. The host adapter has its own set of executable instructions that the host system can issue to the host adapter. This command set can be subdivided into the following three groups:

1. Host adapter commands
2. Mailbox commands
3. BIOS commands

The host system uses the host adapter commands to initialize and to establish control specifications for subsequent operations of the host adapter. The host system uses the host adapter's Command/Parameter Register as an input port through which it issues any host adapter commands and associated parameter bytes to the host adapter. For more details on each of these host adapter commands refer to the heading "Host Adapter Commands" later in this section.

The second category of commands, mailbox commands, is issued to the host adapter when multithreaded operations are required. Mailboxes are reserved storage areas which reside at a fixed contiguous memory location in the host system's main memory. The mailboxes coordinate communications between the host system and the host adapter when the host adapter is operating in multithreaded mode. This software interface enables the host adapter to execute multiple commands concurrently for multiple targets with minimal intervention from the host system. For more details on these mailbox commands refer to the heading "Mailbox Commands" later in this section.

The third category of commands, BIOS commands, is issued to the host adapter when single-threaded operations are required. In this case the BIOS commands work with the host adapter's on-board BIOS. These commands in unison with the on-board BIOS function in a manner fully compatible with DOS and the standard BIOS interface as defined in the host system's technical reference manual. For more details on each of these BIOS commands refer to the heading "BIOS Command Interface" later in this section.

Before these commands are discussed in detail, a brief look at how the host adapter acts as bus master to transfer data on the memory bus and how it requests interrupt service from the host is presented.

Bus Master Direct Memory Access (DMA)

The DMA control logic manages bus arbitration and data transfer coordination. The host adapter operates as a bus master during data transfers. The host adapter arbitrates for bus access and once granted, it takes over control of the bus. It generates the bus address and command strobes. The host adapter supports both odd and even starting addresses, commonly known respectively as aligned and unaligned transfers. If presented with an even transfer count beginning at odd memory starting address, the host adapter will first transfer a single byte (data bits D24-D31). The remaining data is then transferred as double words (32 bits) until the last byte which is transferred as a single byte (data bits D0-D7). While odd byte data transfers are fully supported, it is recommended that when possible host buffers be double-word aligned to gain better data transfer performance.

Interrupt Processing

Several interrupt channels are available to the host adapter. The channel to be used by each host adapter board is specified by configuration settings as described in the host adapter user's guide.

The host system contains a programmable interrupt controller which receives all interrupts and directs the host's CPU to a corresponding vector location which in turn contains a memory address for the software Interrupt Service routine which performs the necessary actions required by each interrupt. It also contains a Mask Register whose bits may be set to mask or cleared to permit corresponding interrupt channels to be acknowledged.

In order to respond correctly to host adapter interrupts during normal operation, the host interrupt controller must be programmed appropriately. The software driver will have to program the host's Interrupt Mask Register and interrupt vector before attempting to use interrupts from the host adapter. The interrupt vector locations and Interrupt Mask Register's bits which need to be cleared to permit acknowledgment of each interrupt channel are listed as follows:

Hardware Interrupt Line		Vector Location In Memory (Hex)	Interrupt Mask Register Hex Address A1
	IRQ9	1C4-1C7	Bit 1
	IRQ10	1C8-1CB	Bit 2
Default	IRQ11	1CC-1CF	Bit 3
	IRQ12	1D0-1D3	Bit 4
	IRQ14	1D8-1DB	Bit 6
	IRQ15	1DC-1DF	Bit 7

If the host adapter is configured for Interrupt Channel 11, for example, the system's interrupt controller must be initialized by **clearing** Bit 3 in its Interrupt Mask Register. The address of the interrupt service routine for Channel 11 will be contained in the four bytes of memory beginning at memory address 1CCH.

The remainder of this section describes the structure and operation of the three categories of commands that the host can issue to the host adapter.

Host Adapter Commands

Host adapter command codes and associated parameter bytes are supplied to the host adapter's Control Register under the coordination of certain bits in the host adapter's Status Register. Table 1-2 provides a summary of these commands.

Table 1-2. Host Adapter Commands

Operation Code Hex Value	Host Adapter Command
00	Test CMDC interrupt
01	Initialize Mailbox
02	Start Mailbox command
03	Start BIOS command
04	Inquire Board ID
05	Enable OMBR interrupt
06	Set SCSI Selection Time-Out
07	Set Time On Bus
08	Set Time Off Bus
09	Set Bus Transfer Rate
0A	Inquire Installed Devices
0B	Inquire Configuration
0D	Inquire Set-up Information
1A	Write Adapter Local RAM
1B	Read Adapter Local RAM
1C	Write Bus Master Chip FIFO
1D	Read Bus Master Chip FIFO
1F	Echo Data Byte
20	Host Adapter Diagnostic
21	Set Adapter Options
81	32-Bit Mode Initialize Mailbox
8D	Inquire Extended Set-up Information

The host system can write a command to the Command/Parameter Register only after checking to see that the Host Adapter Ready bit (HARDY) is set to one, except that the Start Mailbox command and the Enable OMBR Interrupt command may be written at any time. After writing a command, the host may write a predetermined number of parameter bytes to the Command/Parameter Register after checking that the Command/Parameter Register Busy bit (CPRBSY) is zero, indicating that the Command/Parameter Register is not busy and can accept another parameter byte.

In response to some commands, the host adapter may transfer a predetermined number of parameter bytes back to the host. The host adapter places each byte into the Data In Register and then sets the Data In Register Ready bit (DIRRDY) to indicate to the host that the byte is ready to be read by the host. After the host has read the input data byte, the host adapter resets the Data In Register Ready bit (DIRRDY).

The following table lists each host adapter command code along with the associated number of parameter bytes coming into or being sent out from the host adapter and a brief description of the function performed.

Operation Code	Command	Parameter Byte Count	Direction
00	***TEST CMDC INTERRUPT***	None	None

Description. The host adapter's only response to this command is to set the Command Complete bit (CMDC) in the Interrupt Register. When this bit is set, the host can verify proper functioning of this bit.

Operation Code	Command	Parameter Byte Count	Direction
01	***INITIALIZE MAILBOX***	4	Out

Description. This command specifies the number of mailboxes used by the host adapter, and the base memory location of the mailbox array to be used when executing mailbox commands. Four parameter bytes follow the command byte to provide the following information:

Byte	Description
0	Number of mailboxes needed - must be greater than zero.
1 - 3	Base mailbox address - specifies the location of the first byte of the mailbox array. Byte 1 is the most significant byte (MSB).

Each mailbox location in memory will occupy four outgoing mailbox bytes and four incoming mailbox bytes. The Command Invalid bit (CMDINV) will be set with the Command Complete bit (CMDC) if the number of mailboxes is specified as zero. At command completion, the Command Complete bit (CMDC) is set to one and the Initialization Required bit (INREQ) is reset to zero to acknowledge that initialization is unnecessary. See the heading "Mailbox Commands" later in this section for more information on the mailbox structure.

Operation Code	Command	Parameter Byte Count	Direction
02	***START MAILBOX COMMAND***	None	None

Description. This command is normally issued every time the host makes an outgoing mailbox entry. Upon receipt of this command, the host adapter begins scanning for active outgoing mailbox entries and continues scanning until all outgoing mailbox entries have been serviced. This can be accomplished by either beginning the requested operations or queuing the commands and executing them later. To avoid unnecessary interrupt servicing by the host, the Command Complete bit (CMDC) is *not* set after receipt of this command. If this command is received before the Initialize Mailbox command, however, the host adapter will then set both the Command Invalid bit (CMDINV) and the Command Complete bit (CMDC) in the Status Register.

Operation Code	Command	Parameter Byte Count	Direction
03	***START BIOS COMMAND***	None	None

Description. This command is used exclusively by the host adapter's BIOS to communicate with the host adapter's firmware. This command is *not* used by application programs.

Operation Code	Command	Parameter Byte Count	Direction
04	***INQUIRE BOARD ID***	4	In

Description. Upon receipt of this command, the host adapter sends four bytes of data to the host which contain identification and revision information about itself. Refer to these byte contents:

Byte	Description
0	BusLogic Board Type - value allows software support for PC/AT, PCI and Micro Channel BusLogic host adapters.

Hex Value	Meaning
41	Board is a BT-54X, BT-74X, BT-44X, or BT-946C with 64-head BIOS
42	Board is a BT-640A with 64-head BIOS
Other	Reserved

1 Custom Features - indicates what custom features may be supported by the host adapter.

Hex Value	Meaning
41	Standard host adapter
Other	Reserved

2 Firmware Revision Level - a byte designating the revision level of the host adapter firmware. ASCII "0" - "9"

3 Firmware Version - a byte designating the revision level of the installed firmware

Once the four bytes of data have been transferred, the Command Complete bit (CMDC) is set indicating normal command completion.

Operation Code	Command	Parameter Byte Count	Direction
05	***ENABLE OMBR INTERRUPT***	1	Out

Description. This command specifies whether the Outgoing Mailbox Ready bit (OMBR) should be set when an outgoing mailbox entry is cleared by the host adapter. The single parameter byte from the host instructs the host adapter as follows:

Hex Value	Meaning
00	The Outgoing Mailbox Ready Interrupt bit (OMBR) is *not* to be set.
01	The Outgoing Mailbox Ready Interrupt bit (OMBR) is to be set once the outgoing mailbox has been cleared by the host adapter.

To avoid unnecessary interrupt servicing by the host, the Command Complete bit (CMDC) is not set after receipt of this command. If the parameter byte contains a value other than 00H or 01H, however, the Command Invalid bit (CMDINV) is set, as well as the Command Complete bit (CMDC), to indicate receipt of an invalid command.

Operation Code	Command	Parameter Byte Count	Direction
06	***SET SCSI SELECTION TIME-OUT***	4	Out

Description. This command specifies the wait time used to determine whether or not a SCSI selection was successful. If the SCSI Busy signal is *not* returned within the specified time-out period, the selection will be terminated and the appropriate error message recorded in the returned CCB. The contents of the four parameter bytes received with this command are as follows:

Byte	Description
0	Enable/Disable SCSI Selection Time-Out - specifies whether or not the SCSI Selection time-out is used. Refer to the following values.

Hex Value	Meaning
00	No time-out is performed.
01	The time specified in Bytes 02 and 03 is used as the SCSI time-out period.

1 Reserved - must be set to zero.

2 - 3 Time-Out Value - specifies the SCSI selection time-out period in milliseconds. The default value is 250 milliseconds. Byte 2 is the most significant byte.

After command completion, the Command Complete bit (CMDC) is set indicating normal command completion. The Command Invalid bit (CMDINV) is set only if data Byte 0 is invalid (neither 00H nor 01H) or data Byte 1 is not zero which indicates an invalid command.

Operation Code	Command	Parameter Byte Count	Direction
07	***SET PREEMPT TIME ON BUS***	1	Out

Description. This command specifies the time the host adapter is allowed on the bus after being preempted. One parameter byte is sent to the host adapter indicating the length of time in microseconds. This time can be from 2 to 15 microseconds. The default value is 7 microseconds. After command completion, the Command Complete bit (CMDC) is set indicating normal command completion. If the data byte value is greater than 15, the Command Invalid bit (CMDINV) is set indicating that an invalid command was received.

When the host adapter requires a host data transfer, it asserts the Master (x) Request signal and waits for the host arbitration logic to respond with the Master (x) Acknowledge signal. Once it obtains ownership of the bus it then transfers data across the bus directly into or out of main system memory. The host adapter releases the bus when the transfer is completed within Preempt Time On Bus after a preempt condition (deassertion of the Master (x) Acknowledge signal).

Operation Code	Command	Parameter Byte Count	Direction
08	***SET TIME OFF BUS***	1	Out

Description. This command specifies the time the host adapter will spend off the bus. One parameter byte is sent to the host adapter indicating the length of time in microseconds. This command is treated as a no operation command. It is supported for software compatibility to ISA software.

Operation Code	Command	Parameter Byte Count	Direction
09	***SET BUS TRANSFER RATE***	1	Out

Description. This command is treated as a no operation command. It is supported for software compatibility to ISA software.

Operation Code	Command	Parameter Byte Count	Direction
0A	***INQUIRE INSTALLED DEVICES***	8	In

Description. This command asks the host adapter to indicate the devices connected to the SCSI bus. The host adapter issues the SCSI Test Unit Ready command to each target/Logical Unit Number (LUN) combination and reports the results using eight bytes of data returned to the host through the Data In Register. Each byte has an associated target device; i.e., Byte 2 represents Target 2. If a bit has a value of one, the associated LU (Logical Unit) is installed. Each bit within a byte has an associated LU; i.e., Bit 3 represents LU 3, etc.

The byte associated with the host adapter will always be zero. Once all information has been transferred, the Command Complete bit (CMDC) is set to indicate normal command completion.

Operation Code	Command	Parameter Byte Count	Direction
0B	***INQUIRE CONFIGURATION***	3	In

Description. The host adapter returns three bytes of data describing the host DMA channel, the interrupt channel, and the SCSI ID values set during configuration set up.

Except for BusLogic ISA host adapters, BusLogic adapters do not use the ISA DMA channels but may still be required to support the ISA software command Inquire Configuration. This command requires the adapter to specify an ISA DMA channel which is used during host data transfers. ISA software programs the requested DMA channel to function in cascade mode and enables it to receive DMA requests. The adapter's response to the Inquire Configuration command is a configurable option described in the adapter user's guide. Most ISA software executes properly when the adapter specifies no DMA channel in response to this command. Some ISA software, however, requires a specific DMA channel (DRQ5, DRQ6, or DRQ7) to be selected. Because the

specification requires the host system hardware to pull down (deassert) the DRQ lines this option also functions properly. The ISA software believes 16-bit DMA transfers are taking place while the adapter performs 32-bit bus master transfers.

Byte	Description	Byte	Description	Byte	Description
0	Host DMA channel	1	Interrupt channel	2	SCSI ID
	Bit 0 = Reserved		Bit 0 = IRQ9		Bit 0-2 = SCSI ID, binary value
	Bit 1 = Reserved		Bit 1 = IRQ10		Bit 3-7 = Reserved, set to zero
	Bit 2 = Reserved		Bit 2 = IRQ11		
	Bit 3 = Reserved		Bit 3 = IRQ12		
	Bit 4 = Reserved		Bit 4 = Reserved		
	Bit 5 = DMA Channel 5		Bit 5 = IRQ14		
	Bit 6 = DMA Channel 6		Bit 6 = IRQ15		
	Bit 7 = DMA Channel 7		Bit 7 = Reserved		

Operation Code	Command	Parameter Byte Count	Direction
0D	***INQUIRE SETUP INFORMATION***	1	Out
		16	In

Description. This command asks the host adapter to provide information on its the current set-up status. This command is followed by a parameter which specifies the number of bytes which the host adapter will send to the host. The host adapter normally transfers 16 bytes of information.

Parameter Byte

Byte	Description
0	Specifies the number of data bytes, from 0 to 255, to be sent to the host.

Data In Byte/Bit Assignments

Byte	Description		
0	SCSI synchronous negotiation and parity status.		
	Bit	**Meaning**	
	0	A value of zero indicates that SCSI synchronous negotiation will not be initiated by the host adapter.	
		A value of one indicates that SCSI synchronous negotiation will be initiated by the host adapter when appropriate. See the adapter configuration description in the adapter user's guide.	
	1	A value of zero indicates that inbound SCSI transfers are not parity checked.	
		A value of one indicates that parity checking on inbound SCSI transfers is enabled. See the adapter configuration description in the adapter user's guide.	
	2 - 7	Reserved (zero).	
1	BUS TRANSFER RATE - Returns the value set by the Set Bus Transfer Rate command.		
2	TIME ON BUS- Returns the time set by the Set Time On Bus command.		
3	TIME OFF BUS - Indicates the time set by the Set Time Off Bus command.		
4	NUMBER OF MAILBOXES - Returns the number of mailboxes established by a previous Initialize Mailbox command. If the Initialize Mailbox command has not yet been completed successfully, the returned number will be 00H.		
5 - 7	Base Mailbox Address - Returns the base address of the mailbox array established by a previous Initialize Mailbox command. The MSB is Byte 5. If the Initialize Mailbox command has not been completed successfully, these bytes have no meaning.		

8 SYNCHRONOUS VALUES FOR TARGET 0 - Contains information resulting from synchronous negotiation with Target 0. If the address is that of the host adapter or a non-existent target, this byte will contain 00H.

Bit	Meaning
0 - 3	Contain the negotiated **offset** value, normally between 1 and 15.
4 - 6	Contain a value between 0 and 7 that defines the synchronous **transfer period** according to the following equation. Period = 200 + 50 (value) nanoseconds.
7	Set (one) if synchronous transfer is negotiated. Reset (zero) if asynchronous transfers are used.

The following bytes have the same structure as Byte 8, but for Targets 1-7.

9 SYNCHRONOUS VALUES FOR TARGET 1
10 SYNCHRONOUS VALUES FOR TARGET 2
11 SYNCHRONOUS VALUES FOR TARGET 3
12 SYNCHRONOUS VALUES FOR TARGET 4
13 SYNCHRONOUS VALUES FOR TARGET 5
14 SYNCHRONOUS VALUES FOR TARGET 6
15 SYNCHRONOUS VALUES FOR TARGET 7
16 RETURN OF BYTE 1 OF OPC0DE 21

Operation Code	Command	Parameter Byte Count	Direction
1A	***WRITE ADAPTER LOCAL RAM***	3	Out

Description. The three parameter bytes contain a 24-bit address (first parameter byte is MSB) which points to a 64-byte area in the main system memory. Upon receipt of this command the host adapter performs a bus master DMA transfer of the designated 64 bytes from the host's main memory into its own local RAM memory. Once the 64 bytes have been successfully transferred to the host adapter, the Command Complete bit (CMDC) is set indicating that this command has been completed.

Operation Code	Command	Parameter Byte Count	Direction
1B	***READ ADAPTER LOCAL RAM***	3	Out

Description. The three parameter bytes contain a 24-bit address (first parameter byte is MSB) which points to a 64-byte area in the main system memory. Upon receipt of this command the host adapter performs a bus master DMA transfer of 64 bytes of data from its own local RAM into the designated 64 bytes in the host's main memory. Once the 64 bytes have been successfully transferred to the host's memory, the Command Complete bit (CMDC) is set indicating that this command has been completed.

Operation Code	Command	Parameter Byte Count	Direction
1C	***WRITE BUS MASTER CHIP FIFO***	3	Out

Description. The three parameter bytes contain a 24-bit address (first parameter byte is MSB) which points to a 54-byte area in main system memory. Upon receipt of this command the host adapter performs a bus master DMA transfer of the designated 54 bytes from the host's main memory into its own bus master chip FIFO. Once the 54 bytes have been successfully transferred to the host adapter, the Command Complete bit (CMDC) is set indicating that this command has been completed.

Operation Code	Command	Parameter Byte Count	Direction
1D	***READ BUS MASTER CHIP FIFO***	3	Out

Description. The three parameter bytes contain a 24-bit address (first parameter byte is MSB) which points to a 54-byte area in the main system memory. Upon receipt of this command the host adapter performs a bus master DMA transfer of 54 bytes of data from its bus master chip FIFO into the designated 54 bytes in the host's main memory. Once the 54 bytes have been successfully transferred to the host's memory, the Command Complete bit (CMDC) is set indicating that this command has been completed.

Operation Code	Command	Parameter Byte Count	Direction
1F	***ECHO COMMAND DATA***	1	Out
		1	In

Description. This command is used to test the Command/Parameter and Data In Registers and the associated control bits in the remaining I/O registers. The host adapter receives one parameter byte in the Command/Parameter Register and then instructs the host to read the same byte back from the Data In Register by setting the Data In Register Ready bit (DIRRDY). After the host has read the byte, the Command Complete bit (CMDC) is set indicating that this command has been completed.

Operation Code	Command	Parameter Byte Count	Direction
20	***HOST ADAPTER DIAGNOSTIC***	0	Out
		0	In

Description. This command instructs the host adapter to conduct its self-diagnostic tests. A hard reset of the host adapter will occur without issuing a SCSI bus reset. After this command is executed, the host adapter must be reinitialized before normal operation can continue. After issuing this command, the host should monitor the host adapter's Status Register to obtain this command's status. It should also wait until the Diagnostic Active bit (DACT) of the Status Register is reset indicating that the self-diagnostics have been completed successfully. It then should wait for one or more of the following bits to be set: the Diagnostic Failure bit (DFAIL), the Host Adapter Ready bit (HARDY), or the Data In Register Ready bit (DIRRDY). If the Host Adapter Ready bit (HARDY) is set and the Diagnostic Failure bit (DFAIL) is reset, then an error did not occur during the diagnostics.

If the Diagnostic Failure bit (DFAIL) or the Data In Register Ready bit (DIRRDY) is set, then an error did occur during the diagnostics. In this case, after the Data In Register Ready bit (DIRRDY) is set, then the Data In Register should be read for the error code that will equal the number of failed diagnostic tests. No data byte is returned if there is no error. When this command is completed, the Interrupt Register will be updated as follows: the Command Complete bit (CMDC) and the Interrupt Valid bit (INTV) will be set.

Operation Code	Command	Parameter Byte Count	Direction
21	***SET ADAPTER OPTIONS***	3	Out

Description. The host can use this command to specify certain configuration options for the host adapter. The host sends the specified configuration options to the host adapter via a parameter list. This three-byte parameter list follows the command opcode. As the default, Bytes 1 and 2 of the parameter list are set. The parameter list is as follows:

Parameter Byte

Byte	Description
0	Specifies the number of bytes remaining in the parameter list. This specified length equals the total number of bytes in the parameter list minus one.
1	Disables the SCSI disconnection option. Each bit corresponds to a SCSI device (e.g., Bit 0 represents the SCSI device assigned to SCSI address 0). When the bit is set, the host adapter will prevent the corresponding SCSI device from dis-

connecting. When the bit is reset, the host adapter will allow the corresponding SCSI device to disconnect. Byte 16 of the Inquire Setup command reflects the state of this byte.

2 Disables the SCSI Busy retry option. Each bit corresponds to a SCSI device (e.g., Bit 0 represents the SCSI device assigned to SCSI address 0). When the bit is set, the host adapter will prevent the corresponding SCSI device from being retried when the SCSI device returns a Busy status. When the bit is reset, the host adapter will allow the corresponding SCSI device to be retried when the SCSI device returns a Busy status.

Operation Code	Command	Parameter Byte Count	Direction
81	***INITIALIZE EXTENDED MAILBOX***	5	Out

Description. Use the host adapter command 81H to send the host adapter a new command called **Initialize Extended Mailbox**. This command is just like the original Initialize Mailbox command except that it sends a one-byte mailbox count followed by a four-byte, 32-bit address pointing to the first mailbox that can be located anywhere within the 4 Gigabyte memory range. The description of the bytes is as follows:

Byte	Description
0	Mailbox count(Greater than 0)
1	Base Mailbox Address(LSB)
2	Base Mailbox Address
3	Base Mailbox Address
4	Base Mailbox Address(MSB)

The software driver may return the host adapter to the original 24-bit address mode using old data structures by issuing the Initialize Mailbox command (01H).

Operation Code	Command	Parameter Byte Count	Direction
8D	***INQUIRE EXTENDED SETUP***	1	Out
	INFORMATION	4	In

Description. This command asks the host adapter to provide information concerning its set up. This command is followed by a parameter which specifies the number of bytes which the host adapter will send to the host. The host adapter normally transfers four bytes of information.

Parameter Byte

Byte	Description
0	Specifies the number of data bytes, from 0 to 255, to be sent to the host.

Data In Byte/Bit Assignments

0 ASCII code specifying host adapter bus type
"A" for PC/AT Bus
"E" for PCI Bus
"M" for Micro Channel

1 BIOS Address Code
C8 for 0C8000H
CC for 0CC000H
D0 for 0D0000H
D4 for 0D4000H
D8 for 0D8000H
DC for 0DC000H
00 for BIOS Disabled

2-3 Maximum number of segments permitted in the scatter-gather list for this host adapter
Byte 2 is LSB = 00H

Operation Code	Command	Parameter Byte Count	Direction
8F	***ENABLE STRICT ROUND ROBIN MODE***	1	Out

Description. BusLogic host adapters provide two modes for processing outgoing mailboxes.

Parameter Byte

Byte	Description
00	**Strict Round Robin Mode.** In this mode the host adapter firmware maintains a round robin outgoing mailbox pointer. During the outgoing mailbox scanning process, the host adapter only examines the mailbox pointed by the pointer. If that mailbox is active, then the firmware will process it and the pointer is advanced to the next mailbox address. This method guarantees that commands sent to the same target will be executed in the order sent by the host.
01	**Aggressive Round Robin Mode.** In this mode the host adapter firmware follows a round robin sequence to hunt for the outgoing mailbox. However, if the current outgoing mailbox pointed by the pointer is not active, the firmware will continue to advance the pointer to scan the next mailbox until the entire outgoing mailbox structure is tested. The mode is implemented because there are existing device drivers that do not use a strict round robin sequence to update outgoing mailboxes. This method does not guarantee that commands sent to the same target will be executed in the order they were written to the mailbox structure. Setting the mode for strict round robin where the device driver does not support it, also offers no guarantee that the commands will be processed in the order they are sent.

Addressing Mode

BusLogic adapters support both 24-bit and 32-bit addressing modes. 24-bit addressing support offers backward compatibility to early versions of BusLogic host adapter products as well as compatibility with products currently on the market.

However, 24-bit addressing offers limited memory access, allowing only up to 16 MBytes of memory. Current BusLogic adapters support 32-bit addressing, allowing the host adapter unrestricted access to more than 4 Gigabytes of memory.

Under 24-bit mode addressing section below, you'll find a complete description of how 24-bit addressing is implemented by BusLogic host adapters. The description of 32-bit addressing that follows covers the differences between 24-bit and 32-bit addressing operation.

24-Bit Mode Mailbox Commands

With today's application requirements, sophisticated operating systems, and performance capabilities of the latest systems using faster CPUs and higher bus transfer rates, it is desirable to have several different tasks running simultaneously in support of many different users. These tasks require a wide variety of I/O peripheral devices such as hard disks, tape backup units and optical drives.

One advantage offered by the SCSI bus is that up to seven I/O devices may be connected to one host adapter. An effective method is needed to manage the processing of each separate task in a system where individual users make use of many different SCSI I/O devices. A method that takes advantage of the local microprocessor, bus

master ASIC, and other intelligence features of the host adapter is needed. Otherwise, the main system CPU would be loaded down. One such method is the mailbox structure of communication between the host system and the host adapter.

Mailboxes are reserved storage areas which reside at a fixed contiguous memory location in the host system. The mailboxes coordinate communications between the host system and the host adapter. Outgoing mailboxes are used for sending command information to the host adapter and incoming mailboxes are used by the host adapter to return status information about completed commands to the host.

This interface architecture provides a means of passing SCSI device commands from a task, by means of a software driver, to the host adapter in a multi-tasking, multi-user environment with minimal host intervention. In this mode the host adapter can concurrently execute multiple commands for multiple targets. The host adapter's on-board intelligence allows it to complete bus master data transfers, to manage SCSI disconnects and reconnects and to perform other activities which reduce host intervention; thereby, increasing overall system I/O throughput. The following few pages describe the organization and operation of this approach.

When multiple mailboxes are established by the host driver, it is recommended that the outgoing mailbox be used in "round robin" fashion.

24-Bit Mode Mailbox Initialization

With most procedures in a computer system, some means of initialization must be performed to establish starting reference conditions. After power is applied to a system, many system initialization functions are carried out before any application programs are activated.

Similarly, before any mailbox commands can be supplied to the host adapter, another specific initialization process must be conducted.

First, the host system allocates storage space at a fixed contiguous location somewhere in the main system memory where the communication mailboxes will be placed. The host creates the required number of 4-byte outgoing mailboxes, followed immediately in memory by *an equal number* of 4-byte incoming mailboxes. The host then sends an **Initialize Mailbox** command to the host adapter. This command tells the host adapter how many mailboxes will be used and the base address of the first outgoing mailbox in the main system memory as described in the heading "Host Adapter Commands" earlier in this section. There should typically be at least one set of Out/In mailboxes for each active and independent task. Figure 1-9 illustrates an array of four outgoing and four incoming mailboxes.

Before describing the use of outgoing and incoming mailboxes, their structure and contents will be defined.

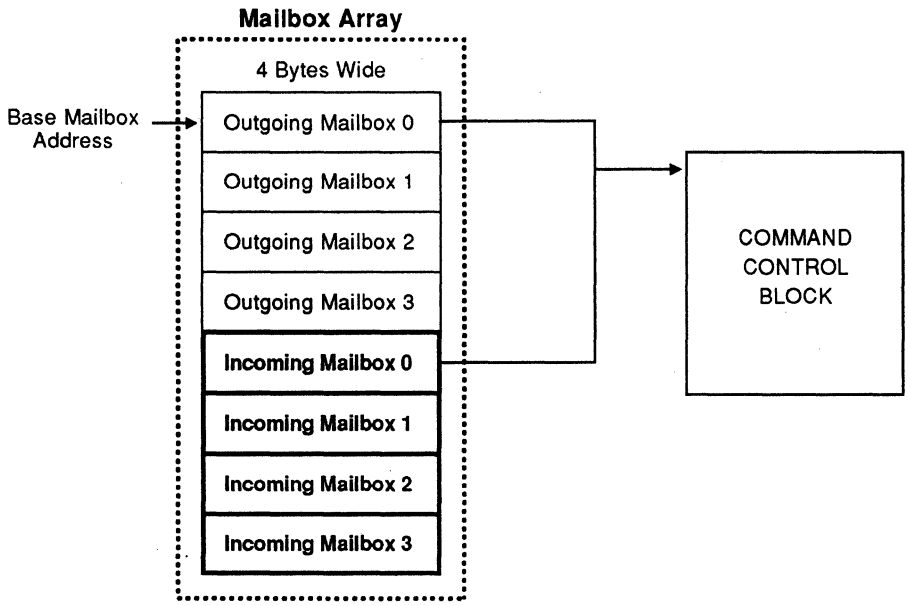


Figure 1-9. Mailbox Array

24-Bit Mode Outgoing Mailbox Structure

The following are the possible contents of the four bytes of each outgoing mailbox.

Byte	Function	
0	Action Code	
	Hex Value	**Definition**
	00	Outgoing Mailbox is not in use
	01	Start a Mailbox command
	02	Abort a Mailbox command
1-3	CCB Address (Byte 1 is MSB)	

The first byte of the outgoing mailbox is the "action" code. If the byte is zero in any particular outgoing mailbox, the host can place a CCB address in the last three bytes of that mailbox and set the first byte in the same mailbox to start or to abort the command contained in the designated CCB. After the host adapter has copied the information in an outgoing mailbox in the course of executing commands, it releases that mailbox by clearing its first byte so the host can use it again.

24-Bit Mode Incoming Mailbox Structure

When a CCB has been provided to the host adapter, the host adapter uses incoming mailboxes to provide information regarding the completion of the CCB to the host. The possible contents of the four bytes of each incoming mailbox of this type are as follows:

Byte	Function	
0	Completion Code	
	Hex Value	**Definition**
	00	Incoming Mailbox is not in use
	01	CCB completed without error
	02	CCB aborted at request of host
	03	Aborted CCB not found
	04	CCB completed with error
1-3	CCB Address (Byte 1 is MSB)	

If the first byte in an incoming mailbox is zero in any particular incoming mailbox, the host adapter can place the address of the completed CCB in the last three bytes of that mailbox and can set the first byte in the same mailbox to indicate the manner in which the command was completed.

Host Adapter as Initiator on the SCSI Bus

To establish communication between the host adapter and an attached SCSI device to execute some command, either the host adapter or the SCSI device can initiate the SCSI command. The other party in the communication is the target. The default case is for the host adapter to be the initiator.

Assuming that a task is running and that communication with a SCSI device is required the host system performs the following preparation to process the I/O request:

- Allocates an area in the main system memory for data storage or retrieval
- Creates a CCB which identifies the SCSI I/O device, provides an address which points to the data area in the main memory and specifies other detailed information about the SCSI command
- Places an address pointer to the CCB and an "action" code in an outgoing mailbox
- Sends a Start Mailbox command to the host adapter.

When the host adapter has received the Start Mailbox command, it performs the following activities:

- Begins scanning the outgoing mailboxes to find entries with action codes using a round-robin scheme
- Copies outgoing mailbox information into its local RAM and then releases the outgoing mailbox
- Copies the CCB pointed to by the outgoing mailbox into its local RAM
- Executes the SCSI command specified in the CCB as soon as the SCSI bus is not busy.

The action taken by the host adapter to release an outgoing mailbox once its information has been copied to local RAM depends on the host adapter command **Enable OMBR Interrupt.** If this command has been issued to the host adapter to enable the Outgoing Mailbox interrupt (OMBR), the host adapter will do one of the following:

- If no interrupt bits are set in the Interrupt Register, the host adapter will set the Outgoing Mailbox Ready (OMBR) and the Interrupt Valid (INTV) bits in the Interrupt Register, and then will assert the Interrupt Request signal on the bus.
- If the Outgoing Mailbox Ready bit (OMBR) is already set to indicate that a previous outgoing mailbox was released and the interrupt was not yet cleared by the host then no additional notification to the host is required.
- If interrupts other than Outgoing Mailbox Ready interrupt (OMBR) are pending, the host adapter will wait for all of them to be cleared before setting the Outgoing Mailbox Ready bit (OMBR).

After processing an outgoing mailbox, the host adapter will scan for another active entry beginning with the outgoing mailbox following the one just completed. The host can ensure that the host adapter will always find the next command with minimum overhead by placing commands in the outgoing mailboxes in consecutive, round-robin order. The host adapter will look for additional outgoing mailbox entries until it finds an empty outgoing mailbox, at which time it will stop scanning. It will start scanning again when the host issues a Start Mailbox command to indicate that another entry has been made. Active entries are transferred into local RAM and placed in a command queue. The local RAM has enough space to store up to 32 CCBs at any one time.

The commands are taken from the queue in a first-in first-out basis to be executed as soon as the SCSI bus is not busy. Commands will not necessarily be completed in the same order because the execution time of each command may be different.

If a Busy condition from the target device temporarily prevents the execution of a command, it is returned to the end of the queue and will be retried when queued up again. This process will continue until the Busy condition is resolved and the command can be completed.

When an outgoing mailbox is found to contain an abort code, the associated CCB pointer is used by the host adapter to locate the mailbox command to be aborted. The designated CCB may be active or queued. If the CCB can be found the command is terminated as soon as possible. The host adapter then makes an incoming mailbox entry to indicate that the command was terminated.

If the CCB can not be found, the command may have already been completed in a normal manner or may have been previously aborted. This situation will also be reported to the host in an incoming mailbox entry.

At the completion of each mailbox command, the target device can report that the completion was "GOOD" or that it has additional status which must be checked. In the latter case, the host adapter may automatically issue a Request Sense command to get the additional status data from the target device. This sense data is stored in a designated area at the end of the CCB in host memory.

At the completion of a mailbox command, the host adapter writes status information into the BTSTAT and SDSTAT fields in the CCB in the main system memory. The host adapter then writes into an incoming mailbox a completion status byte and a pointer to the completed CCB.

After placing an entry into an incoming mailbox, the host adapter will take one of the following actions:

- If no interrupt bits are set in the Interrupt Register, the host adapter will set the Incoming Mailbox Loaded (IMBL) and the Interrupt Valid (INTV) bits in the Interrupt Register and then will assert the Interrupt Request signal on the bus.
- If the Incoming Mailbox Loaded bit (IMBL) is already set to indicate that a previous incoming mailbox was loaded and the interrupt has not yet been cleared by the host then no additional notification to the host is required.
- If interrupts other than Incoming Mailbox Loaded interrupt (IMBL) are pending, the host adapter will wait for all of them to be cleared before setting the Incoming Mailbox Loaded bit (IMBL).

Whenever the host sees that the Incoming Mailbox Loaded bit (IMBL) is set, it begins to scan the incoming mailboxes for active entries. The host will read each active incoming mailbox to obtain status information and pointers to the completed CCBs and then will release the incoming mailboxes for future use. When all active incoming mailboxes have been processed, the host then clears the Incoming Mailbox Loaded bit (IMBL) and the Interrupt Invalid bit (INTV) in the host adapter's Interrupt Register.

The host adapter places **no** restrictions on the data segment address boundaries and lengths that are allowed. For maximum performance, however, it is recommended that all starting addresses (mailboxes, CCBs and data pointers) be on 32-bit (double-word) boundaries and all transfer byte counts be a multiple of four.

Command Control Block (CCB) Structure (24-Bit Mode)

A CCB contains detailed information about a SCSI command. The basic structure is presented in the following table. Detailed descriptions of each byte or field in the CCB follow this table.

Table 1-3. Command Control Block Format

Byte	Description	
0	CCB Operation Code	
	Hex Value	Meaning
	00	Initiator CCB
	01	Reserved
	02	Initiator CCB with scatter-gather
	03	Initiator CCB with residual data length returned
	04	Initiator CCB with scatter-gather, and residual data length returned
	81	SCSI bus device reset
1	SCSI ID and Direction Control	
	Bits 2-0	Logical Unit Number (LUN)
	Bits 4-3	Specify direction of data transfer and whether data length is checked
	Bits 7-5	Target ID if an initiator CCB
2	Length of SCSI Command Descriptor Block	
3	Request Sense Allocation Length/Disable Auto Sense	

Table 1-3. Command Control Block Format (Continued)

Byte	Description
4 - 6	Data Length (Byte 4 is MSB)
7 - 9	Data Pointer (Byte 7 is MSB)
10 - 12	Link Pointer (Byte 10 is MSB)
13	Command Linking Identifier
14	host adapter Status (BTSTAT)
15	SCSI Device Status (SDSTAT)
16 - 17	Reserved and set to zero
18 - n	SCSI Command Descriptor Block
	(Length specified by Byte 2)
n - m	Reserved For Request Sense Information Bytes
	(Length of reserved space is specified in Byte 3)

Table 1-4. Command Control Block Field Definitions

Byte	Field	Description
0	CCB Operation Code (Hex)	
	00	The host adapter acts as the initiator to issue the SCSI command specified in the CDB field of the CCB to the specified target SCSI device.
	01	Not used.
	02	The host adapter acts as the initiator to issue a command to the specified target device in which scatter-gather data transfers are performed. In this case, the Data Length and Data Pointer fields of the CCB have a different meaning as described later in this section.
	03	The host adapter functions as described in the preceding operation code 00H. The only difference between operation code 00H and 03H is the updating of Bytes 4-6 (Data Length field) after the command has been completed. Refer to the description of the Data Length field later in this section for more information on this topic.
	04	The host adapter functions as described in the preceding operation code 02H. The only difference between operation code 02H and 04H is the updating of Bytes 4-6 (Data Length field) after the command has been completed. Refer to the description of the Data Length field later in this section for more information on this topic.
	81	A BUS DEVICE RESET message is sent by the host adapter to the specified target. This forces the host adapter to abort all outstanding tasks against the selected target and to ignore all remaining CCB bytes.
1	Address and Control	Identifies the address of the devices involved in the command and provides information about the expected direction of data flow.
	Bits 7 - 6 - 5	Specifies the target SCSI ID if the CCB is an initiator CCB.
	Bits 4 - 3	Set to determine the direction of the data transfer as follows:

Table 1-4. Command Control Block Field Definitions (Continued)

Byte	Field	Description		
		Initiator CCB		
		Bit	**Bit**	
		4	3	Host Adapter Action
		0	0	Direction of data transfer determined by the SCSI command being executed.
		0	1	Data transferred from SCSI device to host adapter.
				Data transfer will be a Data In phase.
				Data length will be checked.
		1	0	Data transferred from host adapter to SCSI device.
				Data transfer will be a Data Out phase.
				Data length will be checked.
		1	1	No data transfer.
		Target CCB		
		0	0	Return Invalid Target Direction code in Btstat field.
		0	1	Data transferred from SCSI device to host adapter. Data transfer will be a Data Out phase.
		1	0	Data transferred from host adapter to SCSI device. Data transfer will be Data In phase.
		1	1	Return Invalid Target Direction code in Btstat field.
	Bits 2 - 0	Target LUN if an initiator CCB. If the target accepts an IDENTIFY message out, these bits will be provided in the LUN field of the message byte. The LUN field in the SCSI CDB is expected to be zero. If the target does not accept an IDENTIFY message out, the LUN field in the SCSI CDB must contain the correct LUN address. All SCSI-II devices or devices supporting Common Command Set (CCS) accept the IDENTIFY message out. Any device not meeting these requirements should be examined individually to determine whether the LUN address should be placed in Byte 1 of the CCB or in the SCSI CDB.		
2	Length of SCSI Command Descriptor Block	Specifies the number of bytes in the SCSI CDB beginning at Byte 18 of the CCB.		
3	Request Sense Allocation Length/	Indicates the number of bytes in the CCB following the CDB reserved for information that may be obtained by allocation length as its byte count in the CDB for the Request Sense command it issues in response to a Check Condition status received from a target SCSI device at the completion of a command. Sense information is placed in the specified request sense allocation area with a length not exceeding the request sense allocation length. This byte also provides a software method for disabling the Automatic Sense function to override the switch settings described in the adapter user's guide. The following values are defined for this byte:		
		Hex Value		**Meaning**
		00		Allocate 14 bytes for request sense data
		01		Disable automatic request sense
		02-07		Reserved
		08-FF		Valid allocation lengths for SCSI sense data

Table 1-4. Command Control Block Field Definitions (Continued)

Byte	Field	Description
4 - 6	Data Length	Specify the byte length of the data transfer. Error code 12H is posted in the Btstat field if a data overrun occurs. If a scatter-gather operation is specified by the CCB, the Data Length field contains the total number of bytes in the Data Segment List. With operation code 00H or 02H these bytes are not changed after the command is completed; however, with operation code 01H the host adapter will set these bytes to the actual number of bytes transferred after the command is completed. With operation codes 03H and 04H, the host adapter will set these bytes to the difference in the data length originally specified by the host and the actual data length (number of bytes) transferred across the SCSI bus. With operation code 04H the original data length is the segment data length's sum.
7 - 9	Data Pointer	Specify the real address of the first byte of the data area to be used during the data phase of a SCSI command. If a scatter-gather operation is specified by the CCB, the Data Pointer field contains the pointer to the first byte in the Data Segment List.
10 - 12	Link Pointer	Used when a Link or Link With Tag bit is set in a SCSI command. Upon completion of a linked command, the host adapter uses the contents of this field as a pointer to the next CCB to execute. If the Linked Flag bit is set, an interrupt will be generated before the next command is begun. A completed CCB is always reported back in an incoming mailbox. However, Incoming Mailbox interrupts (IMBL) are only reported if the linked set of commands is finished or if a Link with Flag message is presented. There must be enough incoming mailbox entries to receive the entire set of linked commands.
13	Command Linking Identifier	Used in conjunction with linked commands. Set by the host to identify commands in a command chain.
14	Btstat Hex Value	Host adapter status reported to the host.
	00	CCB completed normally with no errors.
	0A	Linked command completed with no errors. The SCSI command was completed and linked with no errors.
	0B	Linked command was completed with no errors and an interrupt was generated. The SCSI command was completed and was linked with a LINKED COMMAND COMPLETE WITH FLAG message.
	11	SCSI Selection time out. Initiator selection or target reselection did not complete within the set SCSI selection time-out period.
	12	Data over run/under run. The target attempted to transfer more or less data than was allocated by the Data Length field or the sum of the Scatter-Gather Data Length fields.
	13	Unexpected bus free.
	14	An invalid bus phase or sequence was requested by the target. The host adapter generated a SCSI Reset state, notifying the host with a SCSI Reset State interrupt (RSTS).
	15	Invalid action code in Byte 0 of the outgoing mailbox.
	16	Invalid operation code in Byte 0 of the CCB.
	17	Linked CCB does not have the same LUN as the first CCB.
	1A	Invalid parameter in CCB or segment list.
	1B	Auto request sense failed.
	1C	SCSI-2 tagged queueing message was rejected by the target.

Table 1-4. Command Control Block Field Definitions (Continued)

Byte	Field	Description
	20	The host adapter hardware failed.
	21	The target did not respond to SCSI ATN and the host adapter consequently issued a SCSI bus reset to clear up the failure.
	22	The host adapter asserted a SCSI bus reset.
	23	Other SCSI devices asserted a SCSI bus reset.
	1B	Auto Request Sense failed.
	1C	A SCSI II Tagged Queuing message was rejected by the Target.
	20	The host adapter hardware failed.
	21	The Target did not respond to SCSI ATN so the host adapter issued a SCSI RST to clear up the failure.
	22	The host adapter asserted SCSI RST.
	23	Other SCSI devices asserted SCSI RST.
	24	The target device reconnected improperly (without tag). An Abort message was issued.
	25	The host adapter issued BUS DEVICE RESET.
	26	Abort Queue generated.
15	SDSTAT	SCSI Device Status. If the host adapter is the initiator, the target will send a status byte to the host adapter at the termination of each SCSI command. The host adapter places that status code in this byte of the CCB to report it to the host. If a Busy status is returned in the SCSI command, the command is executed a second time. The host adapter requeues the command and automatically restarts it until the command completes with a status other than Busy. Status codes reported to the initiator by the target and reported to the host in this byte may have the following values: **Hex Value** — **Status Meaning** 00 — Good 02 — Check Condition. See CCB Byte 3 discussion of Request Sense in response to Check Condition. 08 — Busy
16 - 17	Reserved	Must be set to zero.
18 - n	SCSI Command Descriptor Block	Contains the SCSI CDB Its length is defined in CCB Byte 2. For initiator CCB's, the CDB provided by the host is transmitted to the target.
n - m	Sense Data	If the host adapter detects a Check Condition status once an operation is completed on the SCSI bus, the host adapter automatically executes a Request Sense command with the number of bytes specified by the Request Sense Allocation Length in CCB Byte 3. The bytes returned, up to the maximum indicated by the Request Sense Allocation Length, are placed in this area.

32-Bit Mode Mailbox Structure

Communication between the host and the host adapter is coordinated by the use of outgoing and incoming mailboxes. These mailboxes contain control and status information, and address pointers to CCBs that contain the details of each SCSI command to be processed. The mailbox structures have been changed to the following format:

OUTGOING MAILBOXES

Byte 0	Byte 1	Byte 2	Byte 3
LSB	32-Bit CCB	Pointer	MSB
Reserved	Reserved	Reserved	Action Code
Byte 4	Byte 5	Byte 6	Byte 7

Valid Action Codes are as follows:

Hex Value	Definition
00	Outgoing mailbox is not in use
01	Start a mailbox command
02	Abort a mailbox command.

INCOMING MAILBOXES

Byte 0	Byte 1	Byte 2	Byte 3
LSB	32-Bit CCB	Pointer	MSB
BTSTAT	SDSTAT	Reserved	Completion Code
Byte 4	Byte 5	Byte 6	Byte 7

Valid Completion Codes are as follows:

Hex Value	Definition
00	Incoming mailbox is not in use
01	CCB completed without error
02	CCB aborted at request of host
03	Aborted CCB not found
04	CCB completed with error.

32-Bit Mode CCB Structure

Byte	Description
0	Operation Code
1	Data Direction Control
2	Length of CDB
3	Length of Sense Area
4	LSB
5	Data
6	Length
7	MSB
8	LSB
9	Data
10	Pointer
11	MSB
12	Reserved
13	Reserved
14	BTSTAT
15	SDSTAT
16	Target ID
17	LUN & Tag
18-29	Command Descriptor Block (12 Bytes)
30	CCB Control
31	Link ID
32	LSB
33	Link
34	Pointer
35	MSB
36	LSB
37	Sense
38	Pointer
39	MSB

Note: *In the extended address mode, Bits 3 and 4 of Byte 1 (the Data Direction Control byte) are used as in standard mode. Bits 0-2 and 5-7, however, are reserved in extended mode.*

All reserved bytes and bits should always be set to zero.

32-Bit Mode CCB Description

Many fields of the 32-bit mode CCB are identical to the 24-bit CCB. Its data address fields expand to four-byte fields to contain full 32-bit address pointers. The byte order is reversed, with the LSB coming first rather than the MSB. To enhance symmetry, the Data Length field expands to four bytes as well.

Target ID and LUN numbers are given their own bytes rather than having to share a byte with specification of data direction and length checking in Byte 1 of the CCB.

A Sense Pointer is included. This offers the option of allocating a storage area anywhere in the main memory for the information returned in response to a Request Sense command. One option is to have the Sense Pointer point to the bytes immediately following itself at the end of the CCB. Refer to the following table for a description of 32-bit mode-specific CCB field definitions.

Table 1-5. Command Control Block Field Definitions

Byte	Field	Description		
17	Logical Unit Number (LUN) and Tag			
	Bits 2-0	Specifies the Logical Unit Number (LUN).		
	Bits 4-3	Reserved.		
	Bit 5	Tag enable. When this bit is set, the host adapter will support the tag queueing feature according to the SCSI-2 specifications. When this bit is reset, the host adapter will not support this feature.		
	Bits 6,7	Specifies the tag type. These two bits have no meaning if bit 5 (the Tag Enable bit) is not set. When bit 5 is set, bits 6 and 7 have the following meaning:		
		Bit 7	Bit 6	Message to be sent to the target after IDENTIFY
		0	0	Simple Queue Tag (20H) + Unique Tag ID
		0	1	Head of Queue Tag (21H) + Unique Tag ID
		1	0	Ordered Queue Tag (22H) + Unique Tag ID
		1	1	Reserved (Do not use this value.)
30	CCB Control			
		Bit	**Name**	**Description when the bit is set to 1**
		Bits 2-0		Reserved.
		Bit 3	NoDisc	No disconnect. When this bit is set, the host adapter will select the target with IDENTIFY MESSAGE byte value 80H which will disallow the target from disconnecting the current selection. When this bit is reset, the host adapter will act according to the value of the host adapter command 21H's (Set Adapter Options command) Byte 1. If that value is not programmed it will always allow disconnect.

Table 1-5. Command Control Block Field Definitions (Continued)

Byte	Field	Description	
		Bit 4 NoUnd	No underrun error report. When this bit is set, the host adapter will not report a data overrun/under-run error (BTSTAT value 12H). If this bit is reset, the host adapter will report a data overrun/under-run error whenever the count of the data transfer between the target and the host differs from the count specified in the CCB data count bytes and/or the combined scatter-gather count. The residual count will be posted in the CCB data counts bytes if the CCB opcode is 03 or 04.
		Bit 5 NoData	No data transfer. When this bit is set, the host adapter will not transfer any data between the host adapter and host memory. If this bit is reset, the host adapter will transfer data between the host adapter and host memory.
		Bit 6 NoStat	No CCB status if zero. When this bit is set, the host adapter will not update any CCB status byte if the status to be reported is zero. If this bit is reset, all status bytes will be updated. This bit can be used for performance improvement.
		Bit 7 NoIntr	No interrupts. When this bit is set, the host adapter will not interrupt the host after a com-mand is completed. If this bit is reset, the host adapter will interrupt the host after a command is completed.

Scatter-Gather Operations

In normal CCB operations using SCSI Initiator (00H) and SCSI Target (01H) codes, the CCB contains the pointer (CCB, Bytes 7-9) to the first byte of a contiguous area of data of a specified length (CCB, Bytes 4-6).

Unlike the preceding operation codes, the SCSI initiator with Scatter-Gather code (02H) uses CCB Bytes 7-9 as a pointer to a list of data segments to be transferred. Bytes 4-6 in the CCB specify the length of the Data Segment List. Each entry in the list contains a three-byte field specifying the length of a data segment and a second three-byte field containing a 24-bit address which points to the corresponding data segment in host memory. The Data Segment List is arranged in the order in which data is to be "gathered" or "scattered." The first entry in the list, pointed to by the Data Segment List Pointer in CCB Bytes 7-9, will be used first. A Data Segment List can identify up to 8,192 separate segments of memory. An invalid Data Segment List error will be posted in the Btstat field (1AH) if a list contains zero or more than 8,192 segments.

The structure of the Data Segment List is as follows:

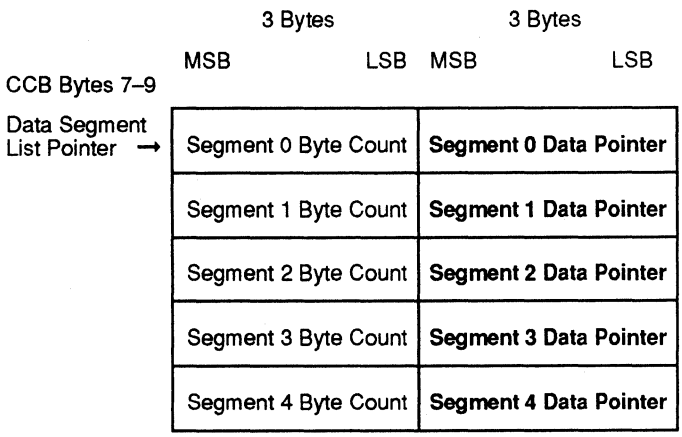


Figure 1-10. Scatter-Gather Data Segment List–CCB Bytes 7-9

The host adapter places **no** restrictions on the data segment address boundaries and lengths that are allowed.

Note: *To enhance performance, it is recommended that all starting addresses (mailboxes, CCBs and data pointers) be on 32-bit (double-word) boundaries and that all transfer byte counts be a multiple of four. If an error occurs during execution of a Scatter-Gather command, the entire command must be retried. It is not possible to determine which segment produced the error.*

Scatter-Gather Operation for 32-Bit Mode

Within 24-bit mode design when a scatter-gather data transfer is performed, the CCB points to a list of the separate data segments involved in the transfer. This list contains a three-byte field specifying the length of each individual data segment and a second three-byte field containing a 24-bit address that points to the corresponding data segment.

With 32-bit addressing, the scatter-gather list must be expanded to pairs of four-byte fields for each data segment in place of the present pairs of three-byte fields. These 32-bit addresses provide unrestricted access to any area of the 4 Gigabyte memory space. The structure of the 32-bit data segment list is as follows:

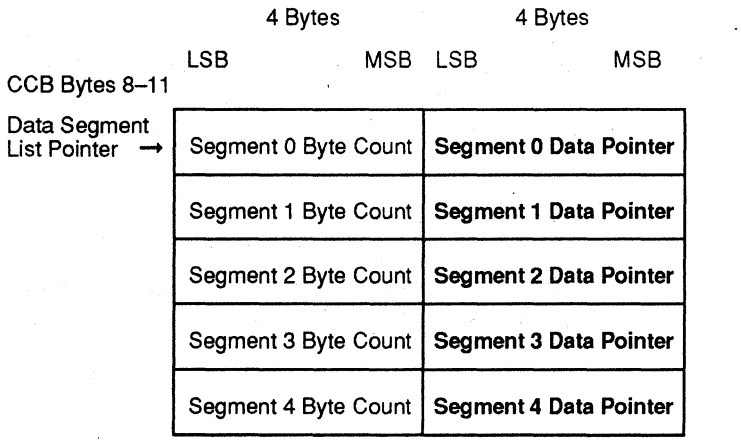


Figure 1-11. Scatter-Gather Data Segment List–CCB Bytes 8-11

In the BusLogic implementation of the host adapter, such a data segment list can have from 1 to 8192 segments.

32- Bit Mode Command Queueing

The host adapter supports command queuing in the 32-bit mode. The following are some device driver notes related to command queuing.

1. Once a device queues up Tag Queue commands, the Initiator should not issue a Non-Tag Queue command (with the exception of Contingent Allegiance condition) until all queued commands have been processed. Therefore, avoid mixing Tag Queue commands with Non-Tag Queue commands.

2. If a CCB is aborted in a Tag queue environment, the host adapter can no longer issue an Abort message to the target. It will have to wait until the Target reestablishes the I_T_L_Q NEXUS. Consequently, it is recommended that a CCB not be aborted in a Tag Queue environment.

3. When a Bus Device Reset is issued, the Target will flush all queued I/O commands. Consequently, the host adapter must return SCSI Reset status for all the CCBs sent to the Target.

4. When a SCSI Bus Reset is issued, all Targets will flush all queued I/O commands. Note that the host adapter will flush all existing TCB in the Disconnect TCB Link List and the Abort TCB Link List. Consequently, the host adapter must return SCSI Reset status for all the CCBs sent out.

5. If there are any outstanding Tag Queue commands and a Target attempts to reconnect without a Tag message, the host adapter will generate an Abort message. The host adapter will flush all outstanding CCBs sent to that Target. All these CCBs will be returned to the host with Host Adapter Status 24H. Refer to the description of the BTSTAT field for new host adapter status.

6. Because a Tag message must be sent right after an ID message, synchronous transfer negotiation initiated by the Initiator takes precedence over a Tag Queue message. This is because most devices that support Tag Queuing may not be able to handle an ID message, followed by a Tag Message, followed by a Synchronous Transfer Negotiations message.

   Consequently, it is recommended that the first couple of commands be sent in Non-tag Queue fashion. This will allow the host adapter and Target to establish Synchronous Transfer mode after each Reset condition.

7. The Auto Sense capability is a useful feature that can be enabled or disabled. If Auto Sense is enabled and a Contingent Allegiance condition occurs, the host adapter will issue a Request Sense command. When a Request Sense command is issued, the Contingent Allegiance condition will be cleared and the Target can continue processing the queued commands.

   When Auto Sense is disabled, the host can deal with the Contingent Allegiance condition itself and decide which recovery procedure to use. However, this may reduce adapter performance. For better performance BusLogic recommends that AutoSense be enabled when using tag queueing.

Implementation Requirements

Existing drivers must be modified to issue the Initialize Extended Mailbox command and to set up the new mailbox and CCB structures. To support scatter-gather transfers an expanded scatter-gather list structure must be provided.

Host adapter firmware will be modified to recognize the Initialize Extended Mailbox command to find the designated section of memory where the CCBs will be located. It will then recognize the specified 32-bit data pointer in each CCB when performing DMA data transfers and the expanded scatter-gather list.

Project Relationship

BusLogic will be pleased to work closely with the software supplier or customer in planning and implementing this extended addressing concept that will enhance SCSI data storage capacities on systems.

BIOS Command Interface

Bus compatible systems provide for a Basic Input/Output System (BIOS) interface in ROM on the system motherboard or on I/O option boards. These BIOS ROMs contain programs which control communication between the Disk Operating System (DOS) and the corresponding I/O peripheral device. Access to each BIOS occurs through a software interrupt of the host CPU. In the case of a hard disk, the software interrupt is Interrupt 13H.

On a standard system, the motherboard BIOS includes support for up to two device-level interfaced hard disks. The BIOS on the host adapter provides equivalent support for up to two SCSI hard disks by intercepting the Interrupt 13H call and by responding to its SCSI device IDs. Control of any additional hard disk drives requires the use of a software driver using mailbox commands.

If two standard hard disks are present in a system, the host adapter's BIOS cannot support additional SCSI drives without a software driver. If no standard hard disks are installed, the host adapter's BIOS can support up to two SCSI hard disks. In this case, Drive 0 (C:) is Device 0, LUN0. Drive 1 (D:) is SCSI Device 1, LUN0. Booting can only be done from SCSI Device 0, LUN0.

If one standard hard disk is installed, it is accessed as Drive 0 (C:). The system may be booted only from this internal hard disk. One SCSI hard disk may be concurrently supported by the host adapter. Its identity is Device 0, LUN0 (D:).

Parameters required to execute commands associated with the control of the hard disks are transferred to and from the BIOS program routines with the use of the host CPU's general registers and segment registers. This interface is capable only with single-threaded operation.

The host adapter's BIOS can accept functions from the DOS operating system that are required for normal operation, system booting, basic maintenance and verification functions.

The host adapter is notified by its on-board BIOS when Interrupt 13H operations are in process by the Start BIOS host adapter command (03H). The host adapter will not respond properly to the Start BIOS command if it is issued by any other source than the on-board BIOS.

BIOS Commands and Input Parameters

The BIOS command code is passed to the host adapter through the CPU's Register AH. The drive number is provided to the host adapter through the CPU's Register DL. The drive number for each command will be 80H or 81H. Refer to the following table for a summary of valid BIOS disk functions. Other input parameters required by the host adapter to execute some of these commands will be described following the table.

Table 1-6. Valid Host Adapter BIOS Disk Functions

Command Code in AH Register (Hex Value)	Command	Description
00	Reset Disk System	The BIOS issues a reset to the SCSI bus. It then sends this command on to the standard BIOS so it can reset other floppy or hard disks in the system.
01	Read Status of Last Operation	The host adapter reports the status of the last operation performed by the specified disk. No SCSI activity occurs and the disk status is reset to zero.
02	Read Desired Sectors Into Memory	The requested sectors, defined by the input parameters, are read from the disk to the system memory. This function maps to a SCSI Read command (SCSI Opcode 08).
03	Write Desired Sectors From Memory	The requested sectors, defined by the input parameters, are written from the system memory to the indicated disk. This function maps to a SCSI Write command (SCSI Opcode 0A).
04	Verify Desired Sectors	The requested sectors, defined by the input parameters, are verified to be written correctly on the SCSI disk. This function maps to a SCSI Verify Command (SCSI Opcode 2F). In some special cases, for targets that do not support the 2F command, this function maps to a SCSI Read command (SCSI Opcode 08) and discards the received data.
06	Identify SCSI Devices	This command is used to determine the number of the first SCSI drive attached to the host adapter.
08	Read Drive Parameters	This function maps to a SCSI Read Capacity command (SCSI Opcode 25). The total logical capacity is then converted to pseudo-physical parameters.
09	Initialize Drive Pair Characteristics	Because SCSI CCS drives are self-configuring, this command performs no operation.
0C	Seek	This function performs a Seek command (SCSI Opcode 0B) to the logical block address as defined by the physical parameters. This command is not required for CCS commands nor for proper SCSI device operations. Because the Seek operation is performed automatically by the Read and Write operations, it is not necessary to generate a Seek command to access data.
0D	Alternate Disk Reset	The BIOS sends a SCSI bus reset to the target specified in the DL Register. A reset function request is also passed to the system's BIOS so that any internally installed hard or floppy disk(s) can be reset.

Table 1-6. Valid Host Adapter BIOS Disk Functions (Continued)

Command Code in AH Register (Hex Value)	Command	Description
10	Test Drive Ready	This function maps to a SCSI Test Unit Ready command (SCSI Opcode 00). After executing a Reset function, the host adapter's BIOS issues this function internally until the Target is no longer busy and the Unit Attention condition is cleared.
11	Recalibrate	This function maps to a Re-zero Unit command (SCSI Opcode 01).
15	Read DASD Type	The BIOS of the host adapter checks the Peripheral Device Type Qualifier (returned by the SCSI Inquiry command) to verify that the device is a Direct Access Device. The BIOS then returns the logical capacity reported by the SCSI Read Capacity command in the CX and DX Registers.

Additional input parameters are required by the Read (02H), Write (03H) and Verify (04H) BIOS commands and are supplied in the following registers:

CPU Register	Input Parameter
AL	Number of Sectors
CH	Low-Order Byte of the Cylinder Number
CL	Cylinder and Sector Numbers
	Bits 7 and 6: High-Order Cylinder Bits
	Bits 5 to 0: Sector Number
DH	Head number
DL	Drive Number
ES:BX	Address of Data Buffer Area

The Seek BIOS command (0C) requires only the cylinder and head numbers from Registers CL, CH and DH. In this case, the sector number bits in Register CL are zero.

The physical starting disk address provided in the preceding registers is converted by the host adapter's BIOS into a logical block address before being sent to the designated SCSI device. The physical address consists of 10 bits to specify up to 1024 cylinders, 8 bits to specify up to 64 heads and 6 bits to specify up to 32 sectors. These bits are combined to form a 21-bit logical block address for the SCSI drive as follows:

Physical Cylinder Number	Physical Head Number	Physical Sector Number-1
10 bits	6 bits	5 bits

21-bit Logical Block Address

BIOS Command Completion Status

When the host adapter has completed the BIOS command, control is returned to the requesting program at the next instruction after the software Interrupt 13H.

The host adapter's BIOS places a completion code in the Carry Flag (CF). If CF is zero, the BIOS command was completed normally and there is no additional status to report. If CF is set to one, normal command completion did not occur and a non-zero status byte will be placed into the CPU's Register AH by the host adapter. This status byte is to be interpreted as follows:

Completion Status Byte Hex Value	Meaning	Hex Sense Code Returned to the Host Adapter from the Request Sense Command
00	No error. Normal Completion.	
01	Invalid Command Request	
02	Address Mark Not Found	12 - No AM Found on Disk
		21 - Illegal Logical Block Address
03	Write Protect Error	27 - Write Protected
04	Read Error	14 - No Record Found
		16 - Data Sync Error
10	Uncorrectable ECC Error	10 - ID ECC Error
		11 - Unrecovered Read Error
11	ECC Corrected Data Error	17 - Recovered Read Error w/o ECC
		18 - Recovered Read Error w/ ECC
20	Controller Failure or one	01 03 05 06 07
	of many Additional Sense	08 09 1B 1C 1D
	Codes was returned	40-49
40	Seek Operation Failed	15 - Seek Positioning Error
		02 - No Seek Complete
80	Selection Time-Out	Drive did not respond to Host Adapter
AA	Device Not Ready	04 - LUN Not Ready
		28 - Medium Changed
		29 - Power On or Reset or Bus Device Reset Occurred
		2A - Mode Select Parameter Changed
BB	Unknown Target Sense Error	Unknown Additional Sense Code from SCSI Device
FF	Sense Operation Failed	No sense information from

Additional output parameters are required by three BIOS commands and are supplied in the following CPU registers:

Command (HEX)	CPU Register	Output Parameter
06	AL	Drive Number of First SCSI Drive Attached
Identify SCSI Devices		80 if no standard hard disk
		81 if one standard hard disk
08	DL	Number of SCSI Drives Attached
Read Drive Parameters	DH	Max value for head number (3F)
	CH	Max value for Cylinder Range (Low Byte)
	CL	Max value for Sector and Cylinder
		Bits 7-6 High Order Cylinder Bits
		Bits 5-0 Max Sector Number (20)
15	AH	Status of Operation
Read DASD		00 Drive not present or DL invalid
		01 Reserved
		02 Reserved
		03 Fixed Disk installed
	CX,DX	Number of 512 byte blocks available on disk

BIOS Disk Commands

The host adapter's BIOS can accept functions from the DOS operating system that are required for normal operation, system booting, and normal disk operation, basic maintenance and verification functions. Refer to the following table for a summary of valid BIOS disk functions. Detailed explanations of the operation of each BIOS command follow the table.

Table 1-7. Valid BIOS Disk Functions

Command Value in AH Register (Hex Value)	Description
00	Reset Disk System
01	Read Status of Last Operation
02	Read Desired Sectors to Memory
03	Write Desired Sectors from Memory
04	Verify Desired Sectors
06	Identify SCSI Devices
08	Read Drive Parameters
09	Initialize Drive Pair Characteristics
0C	Seek
0D	Alternate Disk Reset
10	Test Drive Ready
11	Recalibrate
15	Read DASD Type

In the following descriptions, all references to the SCSI operation codes or parameters input from the host and output back to the host through the various CPU registers are stated in their Hex value for each BIOS command.

00—Reset Disk System. The BIOS issues a reset to the SCSI bus. It then sends this command on to the standard BIOS so it can reset other floppy or hard disks in the system.

Input Parameters: AH=00H
DL=Drive Number (80H or 81H)

Output Parameters: AH=Status of Operation
CF=Return Code

01—Read Status of Last Operation. The host adapter reports the status of the last operation performed by the specified disk. No SCSI activity occurs and the disk status is reset to zero.

Input Parameters: AH=01
DL=Drive Number (80H or 81H)

Output Parameters: AH=Status of Operation
CF=Completion Code

02—Read Desired Sectors to Memory. The requested sectors, defined by the input parameters, are read from the disk to the system memory. This function maps to a SCSI Read command (SCSI Opcode 08).

Input Parameters: AH=02H
DL=Drive Number (80H or 81H)
DH=Head Number
CH=Low-order Byte of Cylinder Number
CL=High-cylinder Bit and Sector Numbers
AL=Number of Sectors to Read
ES:BX=Address of Data Buffer Area

Output Parameters: AH=Status of Operation
CF=Completion Code

03—Write Desired Sectors from Memory. The requested sectors, defined by the input parameters, are written from the system memory to the indicated disk. This function maps to a SCSI Write command (SCSI Opcode 0A).

Input Parameters: AH=03H
DL=Drive Number (80H or 81H)
DH=Head
CH=Low-order Byte of Cylinder Number
CL=High-cylinder Bit and Sector Numbers
AL=Number of Sectors to Write
ES:BX=Address of Buffer Area

Output Parameters: AH=Status of Operation
CF=Return Code

04—Verify Desired Sectors. The requested sectors, defined by the input parameters, are verified to be written correctly on the SCSI disk. This function maps to a SCSI Verify command (SCSI Opcode 2F). In some special cases, for targets that do not support the SCSI Verify command, this function maps to a SCSI Read command (SCSI Opcode 08) and discards the received data.

Input Parameters: AH=04H
DL=Drive Number (80H or 81H)
DH=Head
CH=Low-order Byte of Cylinder Number
CL=High-cylinder Bit and Sector Numbers
AL=Number of Sectors to Verify
ES:BX=Address of Buffer Area

Output Parameters: AH=Status of Operation
CF=Completion Code

06—Identify SCSI Devices. This command is used to determine the number of the first SCSI drive attached to the host adapter.

Input Parameters: AH=06

Output Parameters: AH=Status of Operation
AL= Drive Number of First SCSI Drive Attached
80H if no standard hard disk
81H if one standard hard disk
CF=Completion Code

08—Read Drive Parameters. This function maps to a SCSI Read Capacity command (SCSI Opcode 25). The total logical capacity is then converted to pseudo-physical parameters.

Input Parameters: AH=08H
DL=Drive Number (80H or 81H)

Output Parameters: AH=Status of Operation
DL=Number of SCSI Drives Attached
DH=Max value for head number (3Fh)
CH=Max value for Cylinder Range (Low Byte)
CL=Max value for Sector and Cylinder
Bits 7-6 High-order Cylinder Bits
Bits 5-0 Max Sector Number (20h)
CF=Completion Code

09—Initialize Drive Pair Characteristics. Because SCSI CCS drives are self-configuring, this command performs no operation.

Input Parameters: AH=09H
DL=Drive Number (80H or 81H)

Output Parameters: AH=Status of Operation
CF=Completion Code

0C—Seek. This function performs a Seek operation (SCSI Opcode 0B) to the logical block address as defined by the physical parameters. This command is not required for CCS commands nor for proper SCSI device operations. Because the Seek operation is performed automatically by the Read and Write operations, it is not

necessary to generate a Seek command to access data. If the addressed device reports that the Extended Seek command is not supported, the BIOS command will be completed as normal.

Input Parameters: AH=0CH
DL=Drive Number (80H or 81H)
DH=Head
CH=Cylinder
CL=High Cylinder (Sector bits=0)

Output Parameters: AH=Status of Operation
CF=Completion Code

0D—Alternate Disk Reset. The BIOS sends a SCSI bus reset to the target specified in the DL Register. A reset function request is also passed to the system's BIOS so that any internally installed hard or floppy disk(s) can be reset.

Input Parameters: AH=0DH
DL=Drive Number (80H or 81H)

Output Parameters: AH=Status of Operation
CF=Completion Code

10—Test Unit Ready. This function maps to a SCSI Test Unit Ready command (SCSI Opcode 00). After executing a Reset function, the host adapter's BIOS issues this function internally until the target is no longer busy and the Unit Attention condition is cleared.

Input Parameters: AH=10H
DL=Drive Number (80H or 81H)

Output Parameters: AH=Status of Operation
CF=Completion Code

11—Recalibrate. This function maps to a Re-zero Unit command (SCSI Opcode 01).

Input Parameters: AH=11H
DL=Drive Number (80H or 81H)

Output Parameters: AH=Status of Operation
CF=Completion Code

15—Read DASD. The BIOS of the host adapter checks the Peripheral Device Type Qualifier (returned by the SCSI Inquiry command) to verify that the device is a Direct Access Device. The BIOS then returns the logical capacity reported by the SCSI Read Capacity command in the CX and DX Registers.

Input Parameters: AH=15H
DL=Drive Number (80H or 81H)

Output Parameters: AH=Status of Operation
00 Drive not present or DL invalid
01 Reserved
02 Reserved
03 Fixed Disk installed
CX,DX= Number of 512 byte blocks available on disk
CF=Completion Code

1•4

SCSI Electrical Interface

The host adapter interfaces the host system bus to a SCSI general purpose 8-bit bi-directional bus. The SCSI port is controlled by a SCSI interface chip which supports arbitration, selection, and reselection with a minimum need for processor attention. The SCSI interface controller supports target mode and synchronous SCSI transfers. The host adapter includes single-ended drivers and receivers (built into the SCSI interface chip) which allow a maximum cable length of six meters.

A minimum conductor size of 28 AWG should be employed to minimize noise effects and ensure proper distribution of terminator power.

The internal SCSI connector is a 50-pin, non-shielded SCSI device connector consisting of two rows of 25 male pins with adjacent pins 2.54 mm (0.1 in) apart. The external SCSI connector is a 50-contact, shielded SCSI device connector.

The host adapter uses active termination. All assigned signals are terminated with 110 ohms to the 2.85 volts voltage regulator. All signals must use open-collector or three-state drivers.

Single-Ended Output Characteristics. Each signal driven has the following output characteristics when measured at the connector:

Signal assertion	= 0.0 volts dc to 0.4 volts
Minimum driver output capability	= 48 milliamps (sinking) at 0.5 volts dc (7438 or equivalent)
Signal negation	= 2.5 volts dc to 5.25 volts dc.

Devices receiving the host adapter's output should be of the SCHMITT trigger type to improve noise immunity, 74LS14, 74LS240, or the equivalent. The device should not load the bus with more than two standard low-power Shottky (LS) input loads per line, and should terminate the controller output signals with active 110 ohm terminators or passive 220/330 ohm terminators.

Single-Ended Input Characteristics. Each signal received by the controller should have the following input characteristics when measured at the SCSI device's connector:

Signal true	= 0.0 volts dc to 0.8 volts dc
Maximum total input load	= -0.4 milliamps at 0.4 volts dc
Signal false	= 2.0 volts dc to 5.25 volts dc
Minimum input hysteresis	= 0.2 volts dc.

Terminator Power (Pin 26). BusLogic recommends that the two devices at each end of the cable provide termination.

VTerm. =	4.25 volts dc to 5.25 volts dc
	1.0 amp minimum source drive capability
	1.0 milliamp maximum sink capability

Refer to Figure 1-12 for a schematic representation of how terminator power is provided on the host adapter.

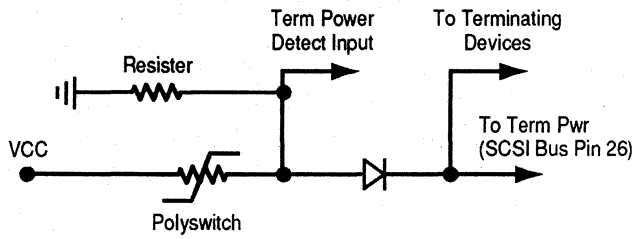


Figure 1-12. Terminator Power Schematic

Terminators. SCSI devices are daisy chained together using a common cable. All signals are common between all SCSI devices, and both ends of the cable are terminated with small hardware components called terminators. Terminators, which are connected to SCSI devices or SCSI cables, make data transfer on a SCSI network more reliable.

Devices connected to SCSI chains must have the correct number of terminators for proper operation and to prevent damage to the SCSI chip on the host adapter. There can be no more than two terminators in a chain of SCSI devices—each at one physical end of the chain. Therefore, if more than two SCSI devices are connected in a SCSI daisy chain, the middle devices in the SCSI cable must have termination disabled.

SCSI Signal Interface

The host adapter's single-ended SCSI interface signals for the internal and external SCSI connectors are shown in Table 1-8. A plus sign (+) denotes an active high signal. A hyphen (-) denotes an active low signal.

Table 1-8. Single-Ended SCSI Interface Signal Pin Assignments

Signal Pin	Signal Name	Signal Pin	Signal Name	Direction (Initiator)
1	Ground	2	-DB0	I/O
3	Ground	4	-DB1	I/O
5	Ground	6	-DB2	I/O
7	Ground	8	-DB3	I/O
9	Ground	10	-DB4	I/O
11	Ground	12	-DB5	I/O
13	Ground	14	-DB6	I/O
15	Ground	16	-DB7	I/O
17	Ground	18	-DBP	
19	Ground	20	Ground	
21	Ground	22	Ground	
23	Reserved	24	Reserved	
25	Open	26	TERMPWR	
27	Reserved	28	Reserved	
29	Ground	30	Ground	
31	Ground	32	-ATN	Output
33	Ground	34	Ground	
35	Ground	36	-BSY	I/O
37	Ground	38	-ACK	Output
39	Ground	40	-RST	I/O
41	Ground	42	-MSG	Input
43	Ground	44	-SEL	I/O
45	Ground	46	-C/D	Input
47	Ground	48	-REQ	Input
49	Ground	50	-I/O	Input

SCSI Signal Definitions

The definitions for SCSI interface signals are shown in Table 1-9.

Table 1-9. SCSI Interface Signal Descriptions

Single-Ended Signal	Definition
–RST	**Reset:** This "OR Tied" signal, which is asserted by the initiator, causes the SCSI bus to cease all operations and return to the Idle condition. This signal is normally used during a power-up sequence. A reset during a Write operation would cause incorrect data to be written on the disk.
–SEL	**Select:** When this signal is asserted by the initiator, along with an initiator ID and target ID data bit (0 -7), it causes the addressed target to be selected. This signal must be deasserted by the initiator after the target asserts the Busy (-BSY) signal in response to a proper selection.
–BSY	**Busy**: When this "OR Tied" signal is asserted, it indicates that the bus is being used.
–C/D	**Control/Data:** When this signal is asserted by the target, it indicates that control information is to be transferred on the data bus. Deassertion of this signal indicates that data information is to be transferred on the data bus.
–I/O	**Input/Output:** When this signal is asserted by the target, it indicates that information will be transferred to the initiator from the target. Deassertion indicates that information will be transferred to the target from the initiator. This signal is also used to distinguish between the Selection and Reselection phases.
–REQ	**Request**: When this signal is asserted by the target, it indicates that an 8-bit byte is to be transferred on the data bus. The Request (REQ) signal is deasserted following the assertion of the Acknowledge (ACK) signal from the host. The Request (REQ) and Acknowledge (ACK) signals control the handshaking.
–ACK	**Acknowledge:** When this signal is asserted by the initiator, it indicates data has been accepted by the initiator or that data is ready to be transferred from the initiator to the target.
–ATN	**Attention:** This signal is driven by the initiator to indicate the Attention condition.
–MSG	**Message:** When this signal is asserted by the target, it indicates the Message phase. The state of the Input/Output (-I/O) signal when it is asserted indicates MESSAGE IN or MESSAGE OUT.
–DB0–7 &–DBP	**Data Bits & Parity**: These eight bidirectional data lines and one odd parity signal are used to transfer 8-bit parallel data over the SCSI bus. Bit 7 is the MSB and has highest priority during the Arbitration phase. Parity is not valid during the Arbitration phase. The use of the parity bit is an option. See the adapter's user's guide to determine how to enable or disable SCSI parity on your adapter.

1•5

Floppy Drive Pin Assignments

The floppy disk controller (if applicable to your model host adapter) allows for the connection of any IBM-compatible standard floppy disk drive to the host adapter. The floppy disk controller part of the host adapter is completely independent, and is accessed by the IBM-compatible BIOS floppy diskette program interface. The floppy disk controller can be disabled. Refer to the adapter user's guide to determine how to enable/disable the floppy disk controller. Table 1-10 describes the floppy disk controller pin assignments for the host adapter.

Table 1-10. Floppy Disk Controller Pin Assignments

Ground Pin	Signal Pin	Signal Name	Direction
1	2	-LD	Output
3	4	Reserved	Output
5	6	Data Rate 0	Output
7	8	-INDEX	
9	10	MOTOR ENB DRIVE A	Output
11	12	DRIVE SELECT B	Output
13	14	DRIVE SELECT A	Output
15	16	MOTOR ENB DRIVE B	Output
17	18	-DIRECTION	Output
19	20	-STEP	Output
21	22	-WRITE DATA	Output
23	24	-WRITE ENABLE	Output
25	26	-TRACK 0	Input
27	28	-WRITE PROTECT	Input
29	30	-READ DATA	Input
31	32	-HEAD SELECT	Output
33	34	-DSKCHNG	Input

1•6

Internal Diagnostics

When the host adapter is powered up, an onboard diagnostic routine is run to verify that the major functional components of the board are operating correctly. The bus master chip, the SCSI controller chip, the firmware PROM, the local RAM and internal data buses are tested. Results of the tests are indicated by an LED on the board.

The LED will first turn on when power is applied. If the diagnostics find no malfunctions, the LED will then go off. In normal operation, the LED will be illuminated when command or SCSI bus activity occurs on the board.

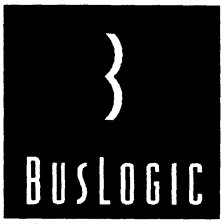
If an error is detected by the diagnostics, the LED will repeatedly flash a specific number of times, with a long pause between flashes, to indicate the board function which failed. This will continue until the board is powered down or reset. Failure interpretation from the number of flashes is as follows:

Number of LED Flashes	Interpretation of Failure
Always On	host adapter is not operating
1	Firmware ROM checksum failure
2	Local RAM test failure
3	SCSI controller chip or SCSI interface failure
4	Internal data bus failure
5	Internal address bus failure
6	Bus master chip failure
Constantly Flashing	Term power failure

1•A

List of Acronyms

BIOS	Basic Input/Output System
CCB	Command Control Block
CCS	Common Command Set
CDB	Command Descriptor Block
CPU	Central Processing Unit
DMA	Direct Memory Access
DRAM	Dynamic Random-Access Memory
EISA	Extended Industry Standard Architecture
FCC	Federal Communications Commission
FIFO	First-In First-Out
I/O	Input/Output
ISA	Industry Standard Architecture
LSB	Least Significant Bit
LU	Logical Unit
LUN	Logical Unit Number
MPU	Microprocessor Unit
MSB	Most Significant Bit
PCB	Printed Circuit Board
PCI	Peripheral Component Interconnect
POS	Programmable Option Select
PROM	Programmable Read-Only Memory
RAM	Random-Access Memory
RFI/EMI	Radio Frequency Interference/Electromagnetic Interference
ROM	Read-Only Memory
SCSI ID	Small Computer System Interface Identification
VL-BUS	VESA (Video Electronics Standards Association) Local Bus



PART 2

PCI HOST ADAPTERS

Part 2: Contents

List of Figures

List of Tables

Section 2•2: Electrical Interface

2•1

Product Overview

The BusLogic BT-946C host adapter is an intelligent PCI to SCSI bus master host adapter product based on a BusLogic-designed MultiMaster ASIC technology. It provides a high-performance interconnection between the PCI (Peripheral Component Interconnect) bus and Small Computer System Interface (SCSI) peripheral devices. The BT-946C is designed for DOS/Windows and multitasking applications such as Windows NT, NetWare, OS/2, UNIX™ and XENIX™.

BusLogic has embedded driver support in most popular operating systems. No additional drivers are needed when your system runs with the current versions of the following operating systems:

- NetWare 3.12/4.X
- Windows NT
- Interactive UNIX
- SCO UNIX
- IBM OS/2
- UNIXWare
- Solaris (for *x*86)
- Vines
- NeXTStep

Software drivers for NetWare 3.11, OS/2, SCO/UNIX and DOS are also available separately.

A BusLogic-designed bus master controller ASIC, an advanced SCSI controller chip and a 16-bit microprocessor chip provide higher speed, lower power consumption, fewer parts and higher reliability. The BT-946C supports a full 32-bit address path, and can access up to four Gigabytes of system memory. The total memory supported is thus limited only by the packaging constraints of the individual product, rather than by the system architecture.

Bus master 32-bit data transfers are performed at speeds up to 132 MBytes/sec on the PCI bus. The BT-946C supports single-ended SCSI drives with asynchronous data rates of up to 7 MBytes/sec and synchronous data rates of up to 10 MBytes/sec with the proper termination and cabling. The BT-946C uses an integrated active terminator interface controller to provide active termination on the SCSI bus. Both internal and external 50-pin connectors are included on the board for flexibility in attaching SCSI devices to the system.

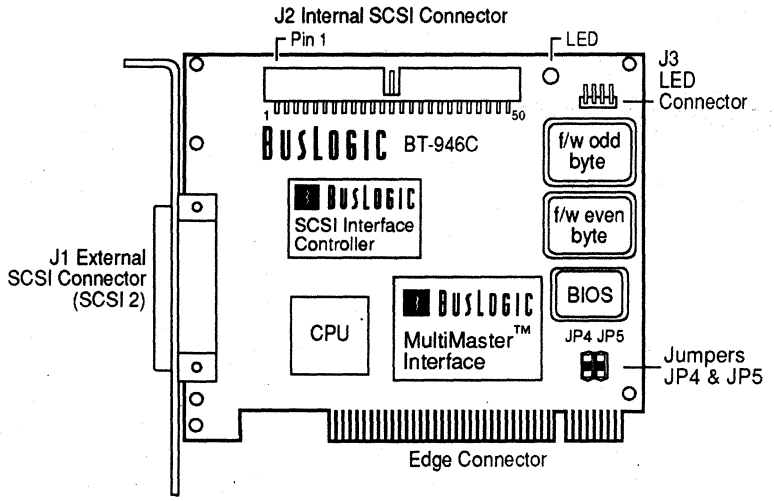


Figure 2-1. The BT-946C Host Adapter Board

Specifications

Dimensions:	5" x 4.2"
Electrical:	
Operating Voltage	5 ± 0.25V
Operating Current	.5 A Max.
Max. Ripple/Noise	100 mV
Environmental:	
Temperature	0°C to 60°C (32°F to 128°F)
Relative Humidity	10% to 95% non-condensing
Altitude	0 to 10,000 ft. operating
	0 to 15,000 ft. non-operating
Connectors:	
SCSI Internal	50-pin double-row connector
SCSI External	50-pin shielded SCSI connector
System Interface	PCI standard edge connector
MTBF	90,000 hours

2•2

Electrical Interface

This section provides the user with a complete description of the name, function, and applicable logic level of all signals between the BT-946C and the PCI host system. It also describes the signals processed by the SCSI protocol chip. This section is divided into two parts:

- PCI System Bus Electrical Interface
- SCSI Bus Electrical Interface

PCI System Bus Electrical Interface

The BT-946C is electrically and mechanically compatible with the Input/Output (I/O) bus used in PCI computers. Physically, this Input/Output bus is contained on the card edge connector. The bus master control logic on the BT-946C controls the PCI system bus arbitration and data transfer operations. During bus master data transfers, the BT-946C takes control of the system bus and transfers data directly to and from the main system memory. Both odd and even starting addresses are supported by the BT-946C.

The PCI system I/O bus provides the necessary hardware interface to the host Central Processing Unit (CPU) to allow it to communicate with the BT-946C. Figure 2-2 identifies the positions of connector rows on the BT-946C's board edge.

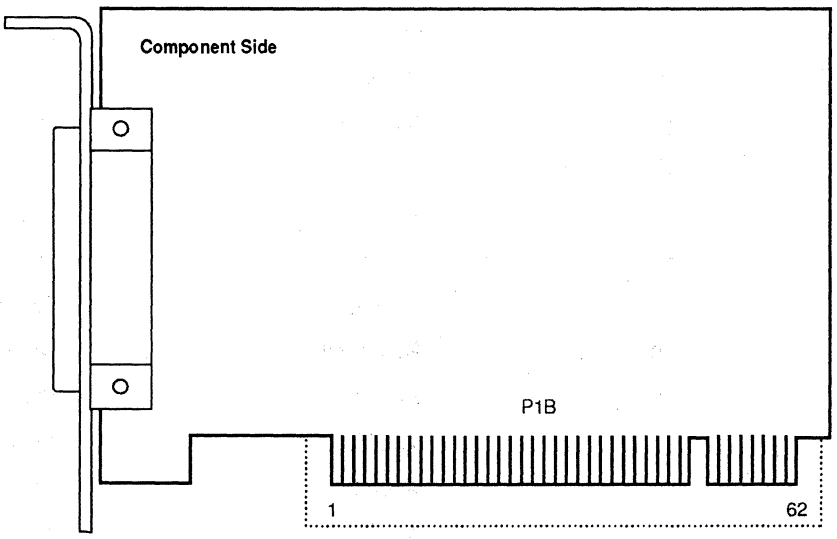


Figure 2-2. Connector Rows on the BT-946C's Board Edge

Summary of PCI Signals

Tables 2-1 and 2-2 summarize the pin assignments for the PCI board's 124-pin edge connector. Following are the signal type definitions used in the tables:

in	Input is a standard input-only signal.
out	Totem Pole Output is a standard active driver.
t/s	Tri-State ® is a bi-directional, tri-state input/output pin.
s/t/s	Sustained Tri-State is an active low tri-state signal owned and driven by one and only one agent at a time. The agent that drives an s/t/s pin low must drive it high for at least one clock before letting it float. A new agent cannot start driving an s/t/s signal any sooner than one clock after the previous owner tri-states it. A pull-up is required to sustain the inactive state until another agent drives it, and must be provided by the central resource.
o/d	Open Drain allows multiple devices to share as a wire-OR.

Table 2-1. PCI Board Pin-outs for Side A (Solder Side)

Pin #	Signal Name	Type
1	TRST#	in
2	+12V	Power
3	TMS	in
4	TDI	in
5	+5V	Power
6	INTA#	o/d
7	INTC#	o/d
8	+5V	Power
9	Reserved	Not Used
10	+5V	Power
11	Reserved	Not Used
12	Ground	Ground
13	Ground	Ground
14	Reserved	Not Used
15	RST#	In
16	+5V	Power
17	GNT#	t/s
18	Ground	Ground
19	Reserved	Not Used
20	AD[30]	t/s
21	+3.3V	Power
22	AD[28]	t/s
23	AD[26]	t/s
24	Ground	Ground
25	AD[24]	t/s
26	IDSEL	in

Table 2-1. PCI Board Pin-outs for Side A (Solder Side) (Continued)

Pin #	Signal Name	Type
27	+3.3V	Power
28	AD[22]	t/s
29	AD[20]	t/s
30	Ground	Ground
31	AD[18]	t/s
32	AD[16]	t/s
33	+3.3V	Power
34	FRAME#	s/t/s
35	Ground	Ground
36	TRDY#	s/t/s
37	Ground	Ground
38	STOP#	s/t/s
39	+3.3V	Power
40	SDONE	in/out
41	SBO#	in/out
42	Ground	Ground
43	PAR	t/s
44	AD[15]	t/s
45	+3.3V	t/s
46	AD[13]	t/s
47	AD[11]	t/s
48	Ground	Ground
49	AD[09]	t/s
50	Keyway	Keyway
51	Keyway	Keyway
52	C/BE[0]#	t/s
53	+3.3V	Power
54	AD[06]	t/s
55	AD[04]	t/s
56	Ground	Ground
57	AD[02]	t/s
58	AD[00]	t/s
59	+5V	Power
60	REQ64#	s/t/s
61	+5V	Power
62	+5V	Power

Table 2-2. PCI Board Pin-outs for Side B (Component Side)

Pin #	Signal Name	Type
1	-12V	Power
2	TCK	in
3	Ground	Ground
4	TDO	out
5	+5V	Power
6	+5V	Power
7	INTB#	o/d
8	INTD#	o/d
9	PRSNT1#	?
10	Reserved	Not Used
11	PRSNT2#	?
12	Ground	Ground
13	Ground	Ground
14	Reserved	Not Used
15	Ground	Ground
16	CLK	in
17	Ground	Ground
18	REQ#	t/s
19	+5V	Power
20	AD[31]	t/s
21	AD[29]	t/s
22	Ground	Ground
23	AD[27]	t/s
24	AD[25]	t/s
25	+3.3V	Power
26	C/BE[3]#	t/s
27	AD[23]	t/s
28	Ground	Ground
29	AD[21]	t/s
30	AD[19]	t/s
31	+3.3V	Power
32	AD[17]	t/s
33	C/BE[2]#	t/s
34	Ground	Ground
35	IRDY#	s/t/s
36	+3.3V	Power
37	DEVSEL#	s/t/s
38	Ground	Ground

Table 2-2. PCI Board Pin-outs for Side B (Component Side) (Continued)

Pin #	Signal Name	Type
39	LOCK#	s/t/s
40	PERR#	s/t/s
41	+3.3V	Power
42	SERR#	o/d
43	+3.3V	Power
44	C/BE[1]#	t/s
45	AD[14]	t/s
46	Ground	Ground
47	AD[12]	t/s
48	AD[10]	t/s
49	Ground	Ground
50	Keyway	Keyway
51	Keyway	Keyway
52	AD[08]	t/s
53	AD[07]	t/s
54	+3.3V	Power
55	AD[05]	t/s
56	AD[03]	t/s
57	Ground	Ground
58	AD[01]	t/s
59	+5V	Power
60	ACK64#	s/t/s
61	+5V	Power
62	+5V	Power

PCI Edge Connector Signal Descriptions

This section lists the PCI signal descriptions.

Table 2-3. Signal Descriptions

Signal	Definition
AD[31::00]	*Address and Data* are multiplexed on the same PCI pins. A bus transaction consists of an address[1] phase followed by one or more data phases. PCI supports both read and write bursts. The address phase is the clock cycle in which **FRAME#** is asserted. During the address phase **AD[31::00]** contain a physical address (32 bits). For I/O, this is a byte address; for configuration and memory it is a DWORD address. During data phases **AD[07::00]** contain the least significant byte (lsb) and **AD[31::24]** contain the most significant byte (msb). Write data is stable and valid when **IRDY#** is asserted, and read data is stable and valid when **TRDY#** is asserted. Data is transferred during those clocks where both **IRDY#** and **TRDY#** are asserted.
C/BE[3::0]#	*Bus Command and Byte Enables* are multiplexed on the same PCI pins. During the address phase of a transaction, **C/BE[3::0]#** define the bus command. During the data phase **C/BE[3::0]#** are used as Byte Enables. The Byte Enables are valid for the entire data phase and determine which byte lanes carry meaningful data. **C/BE[0]#** applies to byte 0 (lsb) and **C/BE[3]#** applies to byte 3 (msb).
CLK	*Clock* provides timing for all transactions on PCI and is an input to every PCI device. All other PCI signals, except **RST#**, **IRQA#**, **IRQB#**, **IRQC#**, and **IRQD#**, are sampled on the rising edge of **CLK**, and all other timing parameters are defined with respect to this edge. PCI operates up to 33 MHz, and in general, the minimum frequency is DC (0 Hz).
DEVSEL#	*Device Select*, when actively driven, indicates the driving device has decoded its address as the target of the current access. As an input, **DEVSEL#** indicates whether any device on the bus has been selected.
FRAME#	*Cycle Frame* is driven by the current master to indicate the beginning and duration of an access. **FRAME#** is asserted to indicate a bus transaction is beginning. While **FRAME#** is asserted, data transfers continue. When **FRAME#** is deasserted, the transaction is in the final data phase.
GNT#	*Grant* indicates to the agent that access to the bus has been granted. This is a point to point signal. Every master has its own **GNT#**.
IDSEL	*Initialization Device Select* is used as a chip select during configuration read and write transactions.
INTA# **INTB#** **INTC#** **INTD#**	Interrupts on PCI are optional and are defined as "level sensitive," asserted low (negative true), using open drain output drivers. The assertion and deassertion of INTx# is asynchronous to CLK. PCI defines one interrupt line for a single function device and up to four interrupt lines for a multi-function (i.e., a single device into which several independent functions have been integrated. Each function on a multi-function device has its own configuration space.) device or connector. For a single function device, only INTA# may be used while the other three interrupt lines have no meaning. *Interrupt A* is used to request an interrupt. *Interrupt B* is used to request an interrupt and only has meaning on a multi-function device. *Interupt C* is used to request an interrupt and only has meaning on a multi-function device. *Interrupt D* is used to request an interrupt and only has meaning on a multi-function device.
IRDY#	*Initiator Ready* indicates the initiating agent's (bus master's) ability to complete the current data phase of the transaction. **IRDY#** is used in conjunction with **TRDY#**. A data phase is completed on any clock both **IRDY#** and **TRDY#** are sampled asserted. During a write, **IRDY#** indicates that valid data is present on **AD[31::00]**. During a read, it indicates the master is prepared to accept data. Wait cycles are inserted until both **IRDY#** and **TRDY#** are asserted together.

[1] The DAC uses two address phases to transfer a 64-bit address.

Table 2-3. Signal Descriptions (Continued)

Signal	Definition
LOCK#	*Lock* indicates an atomic operation that may require multiple transactions to complete. When **LOCK#** is asserted, non-exclusive transactions may proceed to an address that is not currently locked. A grant to start a transaction on PCI does not guarantee control of **LOCK#**. Control of **LOCK#** is obtained under its own protocol in conjunction with **GNT#**. It is possible for different agents to use PCI while a single master retains ownership of **LOCK#**. If a device implements Executable Memory, it must also implement **LOCK#** and guarantee complete access exclusion in that memory. A target of an access that supports **LOCK#** must provide exclusion to a minimum of 16 bytes (aligned). Host bridges that have system memory behind them must also implement **LOCK#**.
PAR	*Parity* is even[2] parity across **AD[31::00]** and **C/BE[3::0]#**. Parity generation is required by all PCI agents. **PAR** is stable and valid one clock after the address phase. For data phases **PAR** is stable and valid one clock after either **IRDY#** is asserted on a write transaction or **TRDY#** is asserted on a read transaction. Once **PAR** is valid, it remains valid until one clock after the completion of the current data phase. (**PAR** has the same timing as **AD[31::00]** but delayed by one clock.) The master drives **PAR** for address and write data phases; the target drives **PAR** for read data phases.
PERR#	*Parity Error* is only for the reporting of data parity errors during all PCI transactions except a Special Cycle. The **PERR#** pin is sustained tri-state and must be driven active by the agent receiving data two clocks following the data when a data parity error is detected. The minimum duration of **PERR#** is one clock for each data phase that a data parity error is detected. (If sequential data phases each have a data parity error, the **PERR#** signal will be asserted for more than a single clock.) **PERR#** must be driven high for one clock before being tri-stated as with all sustained tri-state signals. There are no special conditions when a data parity error may be lost or when reporting of an error may be delayed. An agent cannot report a **PERR#** until it has claimed the access by asserting **DEVSEL#** and completed a data phase.
REQ#	*Request* indicates to the arbiter that this agent desires use of the bus. This is a point to point signal. Every master has its own **REQ#**.
RST#	*Reset* is used to bring PCI-specific registers, sequencers, and signals to a consistent state. What effect **RST#** has on a device beyond the PCI sequencer is beyond the scope of this specification, except for reset states of required PCI configuration registers. Anytime **RST#** is asserted, all PCI output signals must be driven to their benign state. In general, this means they must be tri-stated. **SERR#** (open drain) is floated. **SBO#** and **SDONE**[3] may optionally be driven to a logic low level if tri-state outputs are not provided here. **REQ#** and **GNT#** must both be tri-stated (they cannot be driven low or high during reset). To prevent **AD**, **C/BE#**, and **PAR** signals from floating during reset, the central device may drive these lines during reset (bus parking) but only to a logic low level--they may not be driven high. **REQ64#** has meaning at the end of reset. **RST#** may be asynchronous to **CLK** when asserted or deasserted. Although asynchronous, deassertion is guaranteed to be a clean, bounce-free edge. Except for configuration accesses, only devices that are required to boot the system will respond after reset.
SERR#	*System Error* is for reporting address parity errors, data parity errors on the Special Cycle command, or any other system error where the result will be catastrophic. If an agent does not want a non-maskable interrupt (NMI) to be generated, a different reporting mechanism is required. **SERR#** is pure open drain and is actively driven for a single PCI clock by the agent reporting the error. The assertion of **SERR#** is synchronous to the clock and meets the setup and hold times of all bused signals. However, the restoring of **SERR#** to the deasserted state is accomplished by a weak pullup (same value as used for s/t/s) which is provided by the system designer and not by the signaling agent or central resource. The pullup may take two to three clock periods to fully restore **SERR#**. The agent that reports **SERR#**s to the operating system does so anytime **SERR#** is sampled asserted.

[2] The number of "1"s on **AD[31::00]**, **C/BE[3::0]#**, and **PAR** equal an even number.

[3] **SDONE** and **SBO#** have no meaning until **FRAME#** is asserted indicating the start of a transaction.

Table 2-3. Signal Descriptions (Continued)

Signal	Definition
STOP#	*Stop* indicates the current target is requesting the master to stop the current transaction.
TRDY#	*Target Ready* indicates the target agent's (selected device's) ability to complete the current data phase of the transaction. **TRDY#** is used in conjunction with **IRDY#**. A data phase is completed on any clock both **TRDY#** and **IRDY#** are sampled asserted. During a read, **TRDY#** indicates that valid data is present on **AD[31::00]**. During a write, it indicates the target is prepared to accept data. Wait cycles are inserted until both **IRDY#** and **TRDY#** are asserted together.

2•3

PCI Bus Master Transaction Diagrams

Read Transactions

Figure 2-3 illustrates a read transaction and starts with an address phase which occurs when **FRAME#** is asserted for the first time and occurs on clock 2. During the address phase **AD[31::00]** contain a valid address and **C/BE[3::0]#** contain a valid bus command.

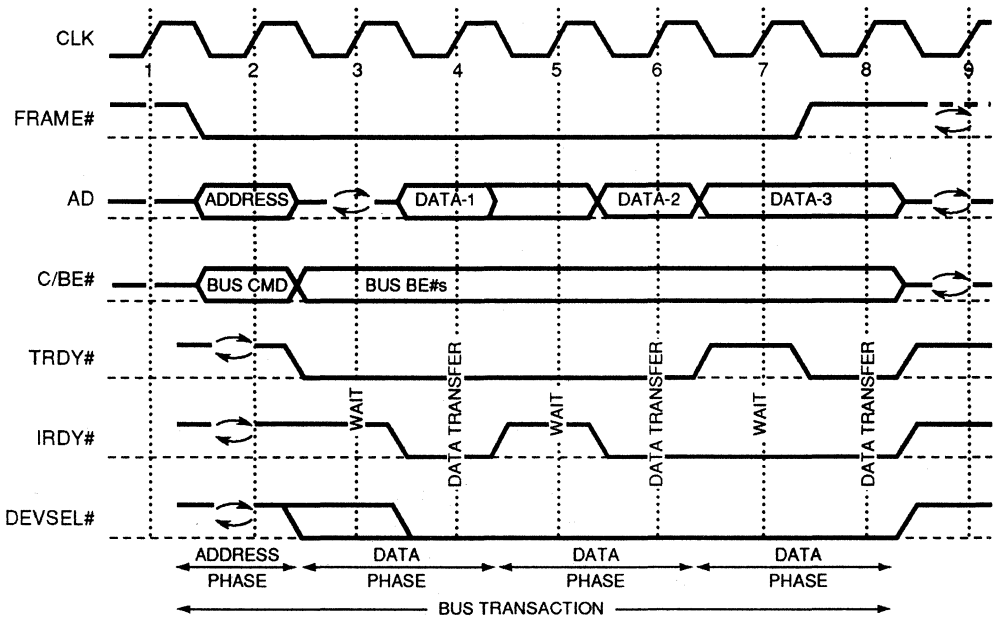


Figure 2-3. Basic Read Operation

The first clock of the first data phase is clock 3. During the data phase **C/BE#** indicate which byte lanes are involved in the current data phase. A data phase may consist of a data transfer and wait cycles. The **C/BE#** output buffers must remain enabled (for both read and writes) from the first clock of the data phase through the end of the transaction. This ensures **C/BE#** are not left floating for long intervals.

The first data phase on a read transaction requires a turnaround cycle (enforced by the target via **TRDY#**). In this case the address is valid on clock 2 and then the master stops driving **AD**. The earliest the target can provide valid data is clock 4. The target must drive the **AD** lines following the turnaround cycle when **DEVSEL#** is asserted. Once enabled, the output buffers must stay enabled through the end of the transaction. (This ensures **AD** are not left floating for long intervals.)

A data phase completes when data is transferred, which occurs when both **IRDY#** and **TRDY#** are asserted on the same clock edge. (**TRDY#** cannot be driven until **DEVSEL#** is asserted.) When either is deasserted, a wait cycle is inserted and no data is transferred. As noted in the diagram, data is successfully transferred on clocks 4, 6, and 8, and wait cycles are inserted on clocks 3, 5, and 7. The first data phase completes in the minimum time for a read transaction. The second data phase is extended on clock 5 because **TRDY#** is deasserted. The last data phase is extended because **IRDY#** was deasserted on clock 7.

The master knows at clock 7 that the next data phase is the last. However, because the master is not ready to complete the last transfer (**IRDY#** is deasserted on clock 7), **FRAME#** stays asserted. Only when **IRDY#** is asserted can **FRAME#** be deasserted, which occurs on clock 8.

Write Transaction

Figure 2-4 illustrates a write transaction. The transaction starts when **FRAME#** is asserted for the first time which occurs on clock 2. A write transaction is similar to a read transaction except no turnaround cycle is required following the address phase because the master provides both address and data. Data phases work the same for both read and write transactions.

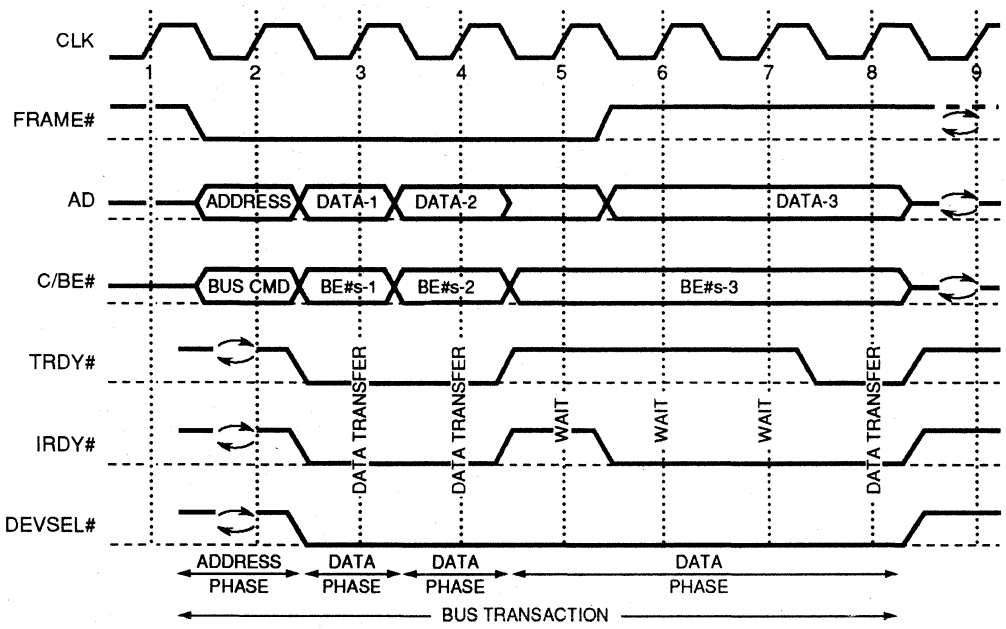


Figure 2-4. Basic Write Operation

In Figure 2-4, the first and second data phases complete with zero wait cycles. However, the third data phase has three wait cycles inserted by the target. Notice both agents insert a wait cycle on clock 5. **IRDY#** must be asserted when **FRAME#** is deasserted indicating the last data phase.

The data transfer was delayed by the master on clock 5 because **IRDY#** was deasserted. Although this allowed the master to delay data, it did not allow the byte enables to be delayed. The last data phase is signaled by the master on clock 6, but does not complete until clock 8.

Arbitration Signaling Protocol

An agent requests the bus by asserting its **REQ#**. Agents must only use **REQ#** to signal a true need to use the bus. An agent must never use **REQ#** to "park" itself on the bus. If bus parking is implemented, it is the arbiter that designates the default owner. When the arbiter determines an agent may use the bus, it asserts the agent's **GNT#**.

The arbiter may deassert an agent's **GNT#** on any clock. An agent must ensure its **GNT#** is asserted on the clock edge it wants to start a transaction. If **GNT#** is deasserted, the transaction must not proceed. Once asserted, **GNT#** may be deasserted according to the following rules.

1. If **GNT#** is deasserted and **FRAME#** is asserted, the bus transaction is valid and will continue.
2. One **GNT#** can be deasserted coincident with another **GNT#** being asserted if the bus is not in the IDLE state. Otherwise, a one clock delay is required between the deassertion of a **GNT#** and the assertion of the next **GNT#**, or else there may be contention on the **AD** lines and **PAR**.
3. While **FRAME#** is deasserted, **GNT#** may be deasserted at any time in order to service a higher priority[1] master, or in response to the associated **REQ#** being deasserted.

Figure 2-5 illustrates basic arbitration. Two agents are used to illustrate how an arbiter may alternate bus access.

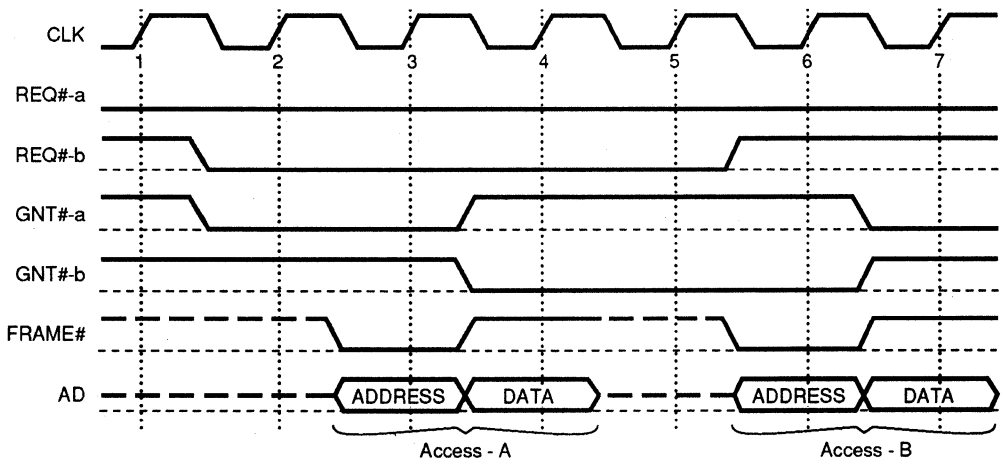


Figure 2-5. Basic Arbitration

REQ#-a is asserted prior to or at clock 1 to request use of the interface. Agent A is granted access to the bus because **GNT#-a** is asserted at clock 2. Agent A may start a transaction at clock 2 because **FRAME#** and **IRDY#** are deasserted and **GNT#-a** is asserted. Agent A's transaction starts when **FRAME#** is asserted on clock 3. Since agent A desires to perform another transaction, it leaves **REQ#-a** asserted.

When **FRAME#** is asserted on clock 3, the arbiter determines agent B should go next and asserts **GNT#-b** and deasserts **GNT#-a** on clock 4.

---

[1] Higher priority here does not imply a fixed priority arbitration, but refers to the agent that would win arbitration at a given instant in time.

When agent A completes its transaction on clock 4, it relinquishes the bus. All PCI agents can determine the end of the current transaction when both **FRAME#** and **IRDY#** are deasserted. Agent B becomes the owner on clock 5 (because **FRAME#** and **IRDY#** are deasserted) and completes its transaction on clock 7.

Notice that **REQ-b** is deasserted and **FRAME#** is asserted on clock 6 indicating agent B requires only a single transaction. The arbiter grants the next transaction to agent A because its **REQ#** is still asserted.

The current owner of the bus keeps **REQ#** asserted when it requires additional transactions. If no other requests are asserted or the current master has highest priority, the arbiter continues to grant the bus to the current master.

GNT# gives an agent access to the bus for a single transaction. If an agent desires another access, it should continue to assert **REQ#**. An agent may deassert **REQ#** anytime, but the arbiter may interpret this to mean the agent no longer requires use of the bus and may deassert its **GNT#**. An agent should deassert **REQ#** in the same clock **FRAME#** is asserted if it only wants to do a single transaction. When a transaction is terminated by a target (**STOP#** asserted), the master must deassert its **REQ#** for a minimum of two PCI clocks, one being when the bus goes to the IDLE state (at the end of the transaction where **STOP#** was asserted) and either the clock before or the clock after the IDLE state. If the master intends to complete the transaction, it must reassert its **REQ#** following the deassertion of **REQ#** or a potential starvation condition may occur. If the master does not intend to complete (because it was prefetching or a higher priority internal request needs to be serviced), the agent asserts **REQ#** whenever it needs to use the interface again. This allows another agent to use the interface while the previous target prepares for the next access.

The arbiter can assume the current master is "broken" if it has not started an access after its **GNT#** has been asserted (its **REQ#** is also asserted), and the bus is IDLE for 16 PCI clocks. However, the arbiter may remove **GNT#** at any time to service a higher priority agent.

2•4

Configuration for Non-Conforming PCI Motherboards

One of the objectives of products designed under the Peripheral Component Interconnect (PCI) Specification 2.0 is to offer a plug-and-play capability-simplified installation, optimized host resources and automatic resolution of configuration conflicts. Configuration and set-up are performed by the BIOS residing in the host system's motherboard. The BT-946C is designed to take advantage of this support. However, while the BT-946C complies with the PCI 2.0 specification, BusLogic has encountered numerous PCI motherboards that do not conform to specification, particularly with regard to PCI SCSI host adapter implementation.

The following discusses what to do to configure the BT-946C when the motherboard is not fully compliant. Refer to your motherboard system documentation to determine which of these configuration adaptations is the correct solution for your particular host system.

The non-conformance problem occurs in how the motherboard BIOS handles interrupt request line (IRQ) assignments. There are inconsistencies among PCI motherboards in how mapping requirements are met. Most operating systems require an IRQ be assigned to a PCI slot or interrupt PIN to operate. Where the mother board supports the BT-946C, its BIOS will automatically assign the IRQ value for all installed host adapters. Where it is not, you will need to first define a BIOS address for the host adapter and then assign IRQs.

Note: After you have manually configured the IRQ and BIOS address values, the adapter onboard BIOS automatically assigns the I/O Port Address to the adapter upon power-up.

For operating systems other than DOS, you need to have an IRQ defined. Currently, onboard BIOS running in the DOS environment does not require an IRQ and you can successfully use the BT-946C's default configuration.

Note: BusLogic's SCSI Adapter Driver Software Kit offers SCSI management for all major operating systems.

The BT-946C supports the PCI configuration space header and PCI data structure as defined in Chapter 6 of the PCI 2.0 specification. Refer to the description of Register 3Ch for details on interrupt line requirements.

BIOS Address Configuration

While the BT-946C is designed to operate with a mother board-assigned BIOS address, if your motherboard is non-compliant, you need to select a fixed BIOS address for the adapter. Selecting a fixed BIOS address allows you to then configure an IRQ value for the adapter as well as other system resource parameters. Use JP4 and JP5 to assign a fixed BIOS address for your BT-946C. The jumpers offer several address options (be sure to select a value that does not conflict with other installed adapters).

Configure JP4 and JP5 as follows:

JP5	JP4	BIOS Address
ON	ON	DC00:0000 (default)
ON	OFF	D800:0000
OFF	ON	C800:0000
OFF	OFF	Allow system BIOS to assign all system resources

ON=installed; OFF=not installed

Where the mother board is fully compliant, remove both jumpers to allow the motherboard BIOS to automatically assign resource parameters: BIOS address, I/O port address and IRQ.

Handling Motherboard Variations

There are five general categories among PCI motherboards. Following are configuration guidelines for handling each kind.

1. If your motherboard offers hardware jumpers to assign IRQ values (IRQ9, IRQ10, IRQ11, IRQ12, IRQ14, IRQ15):

   A. Configure the jumper for one of these IRQ values (as long as it does not conflict with other installed adapters).

   B. Select a BIOS address for the adapter using JP4 and JP5 (as described above).

   C. Then use AutoSCSI to assign the selected IRQ value (the IRQ value must match the value configured via the motherboard jumper) to the BT-946C.

2. If the CMOS utility for your motherboard's system BIOS offers an option to assign the IRQ to a corresponding PCI slot and interrupt PIN type:

   A. Select a BIOS address for the adapter using JP4 and JP5 (as described above).

   B. Configure CMOS to the desired IRQ, PCI slot (the slot the BT-946C is installed in) and PIN values.

   C. Then use AutoSCSI to assign the selected IRQ value (the IRQ value must match the value configured in CMOS) to the BT-946C.

3. In some cases you will need to both set the motherboard jumpers and update CMOS:

   A. Configure the jumper for one of the IRQ values listed in Guideline 1 (as long as it does not conflict with other installed adapters).

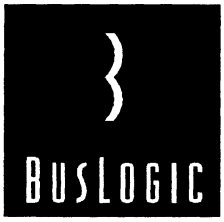
   B. Configure CMOS to the same IRQ selected in Step A, PCI slot (the slot the BT-946C is installed in) and PIN values.

   C. Select a BIOS address for the adapter using JP4 and JP5 (as described above).

   D. Then use AutoSCSI to assign the selected IRQ value (the IRQ value must match the value configured via the motherboard jumper) to the BT-946C.

4. If your motherboard's system BIOS is PCI compliant and supports the BT-946C, remove the jumpers on JP4 and JP5 to allow PCI to automatically assign the IRQ and interrupt PIN type (refer to your motherboard system BIOS supplier for BT-946C support information).

5. If your motherboard offers no means of assigning an IRQ value to the BT-946C, you can only run DOS and Windows. Use the BT-946C's default jumper settings or select another BIOS address using JP4 and JP5. Request BT-946C support from your motherboard supplier.



PART 3

VL-BUS HOST ADAPTERS

Part 3: Contents

Section 3•1: Product Overview

Section 3•2: Electrical Interface

Section 3•3: VL-Bus Timing Descriptions

List of Figures

List of Tables

Section 3•2: Electrical Interface

3-1

Product Overview

The BusLogic BT-445C and BT-440C (BT-44X) host adapters are intelligent VL-Bus to SCSI bus master host adapter products based on BusLogic-designed MultiMaster ASIC technology. The BT-44X board provides a high-performance interconnection between the Video Electronics Standards Association's (VESA) local bus—the VL-Bus, and Small Computer System Interface (SCSI) peripheral devices. It also provides a connection to the standard PC/AT system bus. The BT-44X is designed for DOS/Windows and multitasking applications such as Windows NT, NetWare, OS/2, UNIX™ and XENIX™.

BusLogic has embedded driver support in most popular operating systems. No additional drivers are needed when your system runs with current versions of the following operating systems:

- NetWare 3.12/4.X
- Windows NT
- Interactive UNIX
- SCO UNIX
- IBM OS/2
- UNIXWare
- Solaris (for *x*86)
- Vines
- NeXTStep

Software drivers for NetWare 3.11, OS/2, SCO/UNIX and DOS are also available separately.

A BusLogic-designed bus master controller ASIC, an advanced SCSI controller chip, and a 16-bit microprocessor provide higher speed, lower power consumption, fewer parts and higher reliability.

All bus master mode data transfers, including mailboxes, Command Control Blocks (CCBs) and SCSI peripheral data are performed via the 32-bit, high speed VL-Bus. The PC/AT system bus, on the other hand, allows the host to access the on-board I/O registers, the BIOS, and the floppy controller (BT-445C only). The BT-44X board also interrupts the host via the PC/AT system bus. Hence, the board is not limited to interrupt channel 9 as imposed by the VL-Bus.

First party 32-bit DMA data transfers are performed at speeds of up to 132 MBytes/sec on the VL-Bus. The BT-44X supports single-ended SCSI drives with asynchronous data rates of up to 7 MBytes/sec and synchronous data rates of up to 10 MBytes/sec. The BT-44X uses the voltage regulator and 110 ohm resistor sets for the SCSI bus active termination.

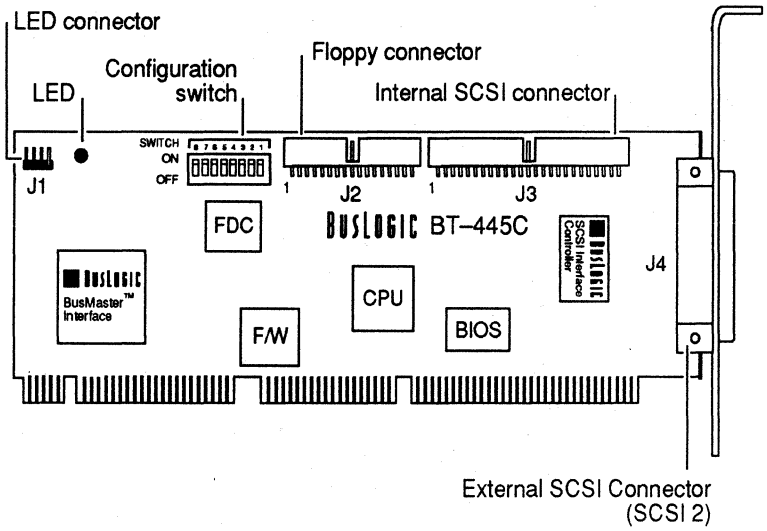


Figure 3-1. The BT-445C Host Adapter

The BT-440C is the same as the BT-445C but without floppy support.

Both internal and external 50-pin connectors are included on the BT-44X for flexibility in attaching SCSI devices to the system. An on-board floppy controller chip (BT-445C only) independently communicates with any combination of up to two 3.5" or 5.25" floppy drives.

Specifications

Dimensions:	9.5" x 3.25"
Electrical:	
Operating Voltage	5±0.25V
Operating Current	.5A Max.
Max. Ripple/Noise	100 mV
Environmental:	
Temperature	0°C to 60°C (32°F to 128°F)
Relative Humidity	10% to 95% non-condensing
Altitude	0 to 10,000 ft. operating
	0 to 15,000 ft. non-operating
Connectors:	
SCSI Internal	50-pin double-row connector
SCSI External	50-pin shielded SCSI connector
Floppy	34-pin AT-compatible ribbon style (BT-445C only)
System Interface	IBM PC/AT standard 36-pin, 62-pin and the VL-Bus 116-pin edge connector
MTBF	90,000 hours

3•2
Electrical Interface

This section provides a complete description of the name, function, and applicable logic level of all signals between the VL-Bus host adapter and the host system. It also describes the signals processed by the SCSI protocol chip and the floppy controller chip.

The VL-Bus host adapter consists of three edge connectors. Edge connectors P1A and P1C are electrically and mechanically compatible with the standard PC/AT system bus. Edge connector P2 is electrically and mechanically compatible with the VL-Bus.

The PC/AT system bus provides the necessary hardware interface to the host Central Processing Unit (CPU) to allow it to communicate with the VL-Bus host adapter via the I/O registers and interrupt channels. It also allows the system to access the VL-Bus host adapter on-board BIOS and the floppy controller. Figure 3-2 shows the location of edge connectors on the board.

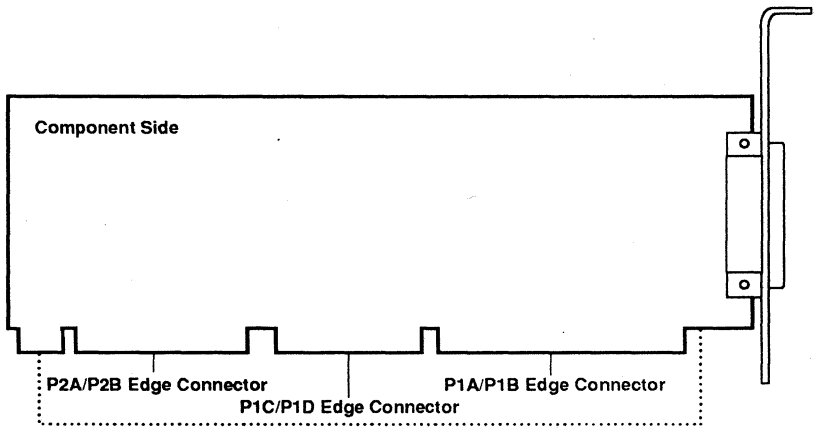


Figure 3-2. VL-Bus Host Adapter Edge Connectors

PC/AT System Bus Electrical Interface

PC/AT System Bus Pin Assignment

Table 3-1 summarizes the pin assignments for the PC/AT system bus connectors on the component side. Table 3-2 shows the pin assignments on the solder side of P1B, a 62-pin edge connector.

Table 3-1. Component Side Of P1A, Edge Connector

Signal Pin #	Signal Name	Direction
A1	-I/O CH CK	Not used
A2	D7	I/O
A3	D6	I/O
A4	D5	I/O
A5	D4	I/O
A6	D3	I/O
A7	D2	I/O
A8	D1	I/O
A9	D0	I/O
A10	I/O CH RDY	Not used
A11	AEN	Input
A12	SA19	Not used
A13	SA18	Not used
A14	SA17	Not used
A15	SA16	Input
A16	SA15	Input
A17	SA14	Input
A18	SA13	Input
A19	SA12	Input
A20	SA11	Input
A21	SA10	Input
A22	SA9	Input
A23	SA8	Input
A24	SA7	Input
A25	SA6	Input
A26	SA5	Input
A27	SA4	Input
A28	SA3	Input
A29	SA2	Input
A30	SA1	Input
A31	SA0	Input

Table 3-2. Solder Side Of P1B, Edge Connector

Signal Pin #	Signal Name	Direction
B1	GND	Ground
B2	RESET DRV	Input
B3	+5 Vdc	Power
B4	IRQ9	Output
B5	-5 Vdc	Not Used
B6	DRQ2	Output
B7	-12 Vdc	Not Used
B8	-ØWS	Not Used
B9	+12 Vdc	Not Used
B10	GND	Ground
B11	-SMEMW	Not Used
B12	-SMEMR	Input
B13	-IOW	I/O
B14	-IOR	I/O
B15	-DACK3	Not Used
B16	DRQ3	Not Used
B17	-DACK1	Not Used
B18	DRQ1	Not Used
B19	-Refresh	Not Used
B20	CLK	Not Used
B21	IRQ7	Not Used
B22	IRQ6	Output
B23	IRQ5	Not Used
B24	IRQ4	Not Used
B25	IRQ3	Not Used
B26	-DACK2	Input
B27	T/C	Input
B28	BALE	Input
B29	+5 Vdc	Power
B30	OSC	Not Used
B31	GND	Ground

PC/AT System Bus Pin Assignments

Table 3-3 summarizes pin assignments for the PC/AT system bus connectors on the component side. Table 3-4 shows the pin assignments on the solder side of P1D, a 36-pin edge connector.

Table 3-3. Component Side Of P1C, Edge Connector

Signal Pin #	Signal Name	Direction
C1	-SBHE	Not used
C2	LA23	Input
C3	LA22	Input
C4	LA21	Input
C5	LA20	Input
C6	LA19	Input
C7	LA18	Input
C8	LA17	Input
C9	-MEMR	Not used
C10	-MEMW	Not used
C11	DO8	Not used
C12	DO9	Not used
C13	D10	Not used
C14	D11	Not used
C15	D12	Not used
C16	D13	Not used
C17	D14	Not used
C18	D15	Not used

Table 3-4. Solder Side Of P1D, Edge Connector

Signal Pin #	Signal Name	Direction
D1	-MEM CS16	Not used
D2	-I/O CS 16	Not used
D3	IRQ10	Output
D4	IRQ11	Output
D5	IRQ12	Output
D6	IRQ15	Output
D7	IRQ14	Output
D8	-DACK0	Not Used
D9	DRQ0	Not Used
D10	-DACK5	Not used
D11	DRQ5	Not used
D12	-DACK6	Not used
D13	DRQ6	Not used

Table 3-4. Solder Side Of P1D, Edge Connector (Continued)

Signal Pin #	Signal Name	Direction
D14	-DACK7	Not used
D15	DRQ7	Not used
D16	+5 Vdc	Power
D17	-MASTER	Not used
D18	GND	GND

PC/AT System Bus Signal Descriptions

The following is a description of the PC/AT system bus signals. All signal lines are TTL-compatible. I/O adapters should be designed with a maximum of two low-power Shottky (LS) loads per line. Signals preceded by a hyphen (-) indicate that the signal is an active low signal.

Table 3-5. PC/AT System Bus Signal Descriptions

Signal	Definition
SA0—SA19	**Address Bits 0–19:** These positive true 20-bit address bits are used to select memory or I/O devices within the system.
AEN	**Address Enable**: This signal is used to disconnect the host CPU and other devices from the bus to allow DMA transfers to take place. When this signal is asserted, the DMA controller has control of the address bus, the data bus, the Read command lines, and the Write command lines.
BALE	**Buffered Address Latch Enable**: A19-A0 are latched on the falling edge of the Buffered Address Latch Enable (BALE) signal to indicate a valid address by the bus master. During DMA cycles the bus master forces the Buffered Address Latch Enable (BALE) high.
–MASTER	**Bus Master:** This signal is used with a DMA Request (DRQ) signal to gain control of the host system bus. The VL-Bus host adapter issues a DRQ signal to the DMA channel in cascade mode and receives a DMA Acknowledge (-DACK) signal. Upon receiving the -DACK signal, the VL-Bus host adapter controller asserts the Bus Master (-MASTER) signal, which allows it to control the system address, data, and control lines. The VL-Bus host adapter is allowed to control the system bus for no more than 15 µs so that the host can refresh the system memory at regular intervals.
–DACK2 & –DACK 5 to 7	**DMA Acknowledge:** These active low signals are used to acknowledge their respective DMA requests.
DRQ 2 & DRQ 5 to 7	**DMA Request**: The DMA Requests 2 and 5 to 7 signals are used to request a host transfer. They are active high. DRQ signals are held high until the corresponding DMA Acknowledge (DACK) signal is asserted.
D0—D15	**Data Bus Bits 0-15:** These positive true signals comprise the 16-bit, tri-state, bi-directional, data bus for commands and status communication between the VL-Bus host adapter and the PC/AT system bus. D7-D0 form the least significant byte of a 16-bit data transfer.
I/O CH RDY	**I/O Channel Ready:** This signal is pulled low by a memory or an I/O device to lengthen I/O or memory cycles.
–I/O CH CK	**I/O Channel Check:** This signal provides the active bus with parity (error) information about memory or devices on the I/O channel. When this signal is active, it indicates an uncorrectable system error. This signal is generated by the active slave device.
–I/O CS 16	This signal indicates to the system that the present data transfer is a 16-bit I/O cycle.
–IOR	**I/O Read**: Input/Output Read instructs an I/O device to place its data onto the data bus. This signal may be generated by any bus master in the system.

Table 3-5. PC/AT System Bus Signal Descriptions (Continued)

Signal	Definition
–IOW	I/O Write: Input/Output Write instructs an I/O device to store data from the data bus. This signal may be generated by any bus master in the system.
IRQ6	**Interrupt Request 6**: This signal is used to tell the host CPU that the floppy controller portion of the host adapter needs attention. A request is generated when this signal is raised from low to high. The signal is held high until the host CPU acknowledges the interrupt request.
IRQ9—IRQ12 & IRQ14—IRQ15	**Interrupt Requests 9–12, 14 and 15**: These signals tell the host CPU that an I/O device needs attention. An interrupt request is generated when an IRQ signal is raised from low to high. The signal is held high until the host CPU acknowledges the interrupt request (Interrupt Service routine).
LA23—LA17	These unlatched signals can be used to address memory within the system, extending the address space to 16Mb. These signals are valid when the Buffered Address Latch Enable (BALE) signal is asserted and are latched by the falling edge of the Buffered Address Latch Enable (BALE) signal.
–MEM CS16	**Memory Chip Select:** This signal indicates that the present data transfer is a 16-bit memory cycle derived from the slave device's decoding of LA17–LA23.
–MEMR	**Memory Read**: The memory read signal instructs the memory devices to drive data onto the data bus. This signal may be driven by any bus master in the system.
–MEMW	**Memory Write:** The memory write signal instructs the memory devices to store data from the data bus. This signal may be driven by any bus master in the system.
RESET DRV	**Reset Drive**: This active high signal is used to reset or to initialize system logic at power-up time or during a low line-voltage outage.
–SBHE	**System Bus High Enable:** This signal enables a data transfer on the upper byte (D15-D8) of the data bus. The present bus master uses this signal to condition the data bus buffers driving D15-D8 lines.
T/C	**Terminal Count:** This signal provides a pulse when the terminal count for any DMA channel is reached.

VL-Bus Electrical Interface

Edge connector P2 in Figure 3-2 is the VL-Bus connector. All bus master mode data transfers, including mailboxes, CCBs, and SCSI peripheral data are performed via the 32-bit VL-Bus. The DMA control logic on the VL-Bus host adapter controls the VL-Bus arbitration and data transfer operations. During DMA data transfers, the VL-Bus host adapter takes control of the VL-Bus and transfers data directly to and from the host memory. Both odd and even starting addresses are supported in the DMA logic.

VL-Bus Pin Assignments

Table 3-6 summarizes pin assignments for the VL-Bus connectors on the component side. Table 3-7 shows pin assignments on the solder side of P2B, a 116-pin edge connector.

Table 3-6. Component Side Of P2A, Edge Connector

Signal Pin #	Signal Name	Direction
A1	DAT01	I/O
A2	DAT03	I/O
A3	GND	Ground
A4	DAT05	I/O
A5	DAT07	I/O
A6	DAT09	I/O
A7	DAT11	I/O
A8	DAT13	I/O
A9	DAT15	I/O
A10	GND	Ground
A11	DAT17	I/O
A12	V_{cc}	Power
A13	DAT19	I/O
A14	DAT21	I/O
A15	DAT23	I/O
A16	DAT25	I/O
A17	GND	Ground
A18	DAT27	I/O
A19	DAT29	I/O
A20	DAT31	I/O
A21	ADR30	Output
A22	ADR28	Output
A23	ADR26	Output
A24	GND	Ground
A25	ADR24	Output
A26	ADR22	Output
A27	V_{cc}	Power
A28	ADR20	Output
A29	ADR18	Output
A30	ADR16	Output
A31	ADR14	Output
A32	ADR12	Output
A33	ADR10	Output
A34	ADR08	Output
A35	GND	Ground
A36	ADR06	Output
A37	ADR04	Output
A38	-WBACK	Not used

Table 3-6. Component Side Of P2A, Edge Connector (Continued)

Signal Pin #	Signal Name	Direction
A39	-BE0	Output
A40	V_{cc}	Power
A41	-BE1	Output
A42	-BE2	Output
A43	GND	Ground
A44	-BE3	Output
A45	-ADS	Output
A46	Key	Key
A47	Key	Key
A48	-LRDY	Output
A49	-LDEV<*x*>	Not used
A50	-LREQ<*x*>	Output
A51	GND	Ground
A52	-LGNT<*x*>	Input
A53	V_{cc}	Power
A54	ID2	Not used
A55	ID3	Not used
A56	ID4	Not used
A57	-LKEN	Not used
A58	-LEADS	Output

Table 3-7. Solder Side Of P2B, Edge Connector

Signal Pin #	Signal Name	Direction
B1	DAT00	I/O
B2	DAT02	I/O
B3	DAT04	I/O
B4	DAT06	I/O
B5	DAT08	I/O
B6	GND	Ground
B7	DAT10	I/O
B8	DAT12	I/O
B9	V_{cc}	Power
B10	DAT14	I/O
B11	DAT16	I/O
B12	DAT18	I/O
B13	DAT20	I/O
B14	GND	Ground

Table 3-7. Solder Side Of P2B, Edge Connector (Continued)

Signal Pin #	Signal Name	Direction
B15	DAT22	I/O
B16	DAT24	I/O
B17	DAT26	I/O
B18	DAT28	I/O
B19	DAT30	I/O
B20	V_{cc}	Power
B21	ADR31	Output
B22	GND	Ground
B23	ADR29	Output
B24	ADR27	Output
B25	ADR25	Output
B26	ADR23	Output
B27	ADR21	Output
B28	ADR19	Output
B29	GND	Ground
B30	ADR17	Output
B31	ADR15	Output
B32	V_{cc}	Power
B33	ADR13	Output
B34	ADR11	Output
B35	ADR09	Output
B36	ADR07	Output
B37	ADR05	Output
B38	GND	Ground
B39	ADR03	Ouput
B40	ADR02	Output
B41	NC	Not used
B42	-RESET	Not used
B43	-D/C	Output
B44	-M/IO	Output
B45	-W/R	Output
B46	Key	Key
B47	Key	Key
B48	-RDYRTN	Input
B49	GND	Ground
B50	IRQ9	Not used
B51	-BRDY	Input
B52	-BLAST	Output

Table 3-7. Solder Side Of P2B, Edge Connector (Continued)

Signal Pin #	Signal Name	Direction
B53	ID0	Not used
B54	ID1	Not used
B55	GND	Ground
B56	LCLK	Input
B57	V_{cc}	Power
B58	-LBS16	Not used

P3 VL-Bus Signal Descriptions

The following is a description of the VL-Bus signals. All signal lines are TTL-compatible. Signals preceded by a hyphen (-) indicate an active low signal.

Table 3-8. P3 VL-Bus Signal Descriptions

Signal	Definition
ADR02—ADR31	**Address Bus:** The address bus furnishes the physical memory or I/O port addresses to the VL-Bus target. On the motherboard, these positive true signals can either directly connect to the CPU address bus, or be tri-state buffered. They are driven by the CPU for all CPU-initiated transfers. During system I/O bus master or DMA cycles the VL-Bus controller drives system I/O bus addresses back to these signals. During VL-Bus master cycles, the VL-Bus device acting as bus master drives the address bus. If no VL-Bus target claims the transfer, the VL-Bus controller will drive the VL-Bus master address to the system I/O bus.
-ADS	**Address Data Strobe**: On the motherboard, this active low signal can either be directly connected to the CPU Address Data Strobe (-ADS) signal, or be tri-state buffered. During ISA DMA or ISA bus master transfers the VL-Bus controller acts as the active host on behalf on the ISA bus. This signal is asserted by the VL-Bus controller for one clock cycle after the address bus and status lines are valid on the VL-Bus. During VL-Bus master transfers this signal is asserted by the active VL-Bus master for one clock cycle after the address bus and status lines are valid.
-BE0— -BE3	**Byte Enables**: The byte enables indicate which byte lanes of the 32-bit data bus are involved with the current VL-Bus transfer. On the motherboard, these active low signals can either be directly connected to the CPU Byte Enables (-BE0 –BE3) signal, or tri-state buffered. A 386SX class host CPU must translate A0, A1, and -BHE to the proper byte enable format before driving the Byte Enables (-BE0 – -BE3) signal to the VL-Bus targets. The Byte Enables (-BE0 — -BE3) signal is driven by the CPU for all CPU-initiated transfers. During system I/O bus master or DMA cycles the VL-Bus controller drives these signals according to the values of the system I/O bus address lines 0, 1 and -SBHE. During VL-Bus master transfers the VL-Bus device acting as bus master drives -BE0 through -BE3 to indicate which byte lanes contain valid data.
-BLAST	**Burst Last**: This active low signal indicates that the next time the Burst Ready (-BRDY) signal is asserted the burst cycle will complete. On the motherboard this signal can either be directly connected to the CPU Burst Last (-BLAST) signal, or tri-state buffered. The 386SX and DX class host CPU must drive this signal low whenever the host controls the VL-Bus. During VL-Bus master transfers the VL-Bus device acting as bus master drives this signal. A VL-Bus master that does not support burst transfers must drive this signal low whenever it controls the VL-Bus.

Table 3-8. P3 VL-Bus Signal Descriptions (Continued)

Signal	Definition
-BRDY	**Burst Ready**: This active low signal terminates the current active burst cycle. It is asserted low for one Local CPU Clock (LCLK) period at the end of each burst transfer. It must be synchronized to LCLK so that appropriate set up and hold times to LCLK are satisfied. If the Local Ready (-LRDY) signal is asserted at the same time as this signal, -BRDY is ignored and the remainder of the current burst cycle falls back to non-burst cycles. If a VL-Bus target does not support burst cycles, this signal can remain unconnected. This signal should not be asserted if the requesting device does not support burst transfers (i.e., the Burst Last (-BLAST) signal is low). Tri-state control of this signal follows the same rules as the Local Ready (-LRDY) signal. It is shared among all VL-Bus devices; therefore, the active VL-Bus target drives it only during a burst transfer that it has claimed as its own. It must not be driven while the Address Data Strobe (-ADS) signal is asserted. Due to this restriction, the first cycle of a burst transfer must contain a minimum of one wait state. The active LBT must drive this signal high for one half LCLK cycle before tri-stating the signal. In most cases, this signal should not be driven during the first T2 period because the system cache controller may be driving it. However, if the High Speed Write configuration bit is set, the LBT may start driving this signal during the first T2 state. While no burst cycle is active on the VL-Bus, this signal is held high by a 20K pull-up resistor located on the motherboard.
DAT00—DAT31	**Data Bus:** This is a bidirectional data path between VL-Bus devices and the CPU. On the motherboard the Data Bus (DAT00—DAT31) can either be directly connected to the CPU data bus or buffered. During read transfers the active VL-Bus target drives data onto the data bus. If the read is initiated from a system I/O bus master or motherboard DMA, the data is driven onto the system I/O bus data bus by the VL-Bus controller. During write transfers the CPU, DMA slave, or bus master drives data onto the data bus. The Byte Enables (-BE0 — -BE3) signal determines which byte lane(s) of the data bus (DAT00—DAT31) are valid.
-D/C	**Data or Code Status**: The Data or Code Status (-D/C) signal indicates whether the current cycle is transferring data or code. This signal is useful to devices employing separate data and code caches. On the motherboard, the -D/C signal can either be directly connected to the CPU -D/C signal, or be tri-state buffered. During VL-Bus master transfers the VL-Bus device acting as bus master drives this signal. If the VL-Bus master does not differentiate between data and code, it must drive this signal to the default data state (high) whenever it owns the VL-Bus.
ID0—ID3	**Identifier Pins**: The identifier (ID) pins allow any VL-Bus target to identify the type and speed of the host CPU. The ID pins typically are static levels and do not change values while the system is in operation; however, the ID pins are defined as valid only during power on reset (i.e., while the Reset (-RESET) signal is asserted and should be latched at the trailing edge of -RESET. **CPU Type** — **ID<1>** — **ID<0>** Reserved — 0 — 0 386 — 0 — 1 486 — 1 — 0 Reserved — 1 — 1 **High Speed Write** — **ID<2>** 0 wait write minimum — 1 1 wait write minimum — 0 **CPU Speed** — **ID<3>** <=33 MHz — 1 >33 MHz — 0

Table 3-8. P3 VL-Bus Signal Descriptions (Continued)

Signal	Definition
	ID<4> is reserved for future use and must always be set to zero. The CPU Type Identifier bits describe the host CPU type. "386" describes a 386SX, 386DX, or compatible processor. "486" describes a 486SX, 486DX, 486DX2, or compatible processor. The High Speed Write Identifier bit is set if the VL-Bus controller is capable of handling high-speed zero wait state write transfers. Many motherboards with cache drive the CPU Ready(-RDY) and Burst Ready (-BRDY) signals during the first T2 state. Because the motherboard cache and the VL-Bus controller cannot simultaneously drive the CPU -RDY signal, the VL-Bus controller must wait until the second T2 when the cache is no longer driving the CPU -RDY and -BRDY signals. If a cache does not drive the CPU -RDY and -BRDY signals during the first T2 state, the High Speed Write bit is set because the VL-Bus controller may complete a write transfer with zero wait states. If the VL-Bus target cannot complete a write in zero wait states it may ignore this bit and default to a minimum of one wait states. Read transfers are unaffected by the High Speed Write configuration bit setting. The CPU Speed Identifier describes the maximum clock speed of the CPU. In the case of systems that can dynamically change the CPU speed (e.g., portables), the speed is defined as the maximum speed of the CPU. A system with a clock speed of 33.3 MHz is considered 33 MHz.
IRQ9	**Interrupt Request Line 9:** The IRQ9 signal is a high-asserted, level-triggered interrupt that is electrically connected to the IRQ9 signal on the ISA bus. It is present on the VL-Bus connector for "standalone" VL-Bus device boards that do not have any system I/O bus signals available. Normally, the VL-Bus device should connect to the interrupt lines on the system I/O bus. For Micro Channel systems, this signal is connected to the Micro Channel -IRQ 09 through an inverter on the motherboard.
-LBS16	**Local Bus Size 16**: This active low signal forces the CPU or LBM to run multiple 16-bit transfers to a VL-Bus target that cannot accept 32 bits of data in a single clock cycle. This signal is shared among all VL-Bus targets and must only be asserted by the active VL-Bus target. It must be asserted one clock cycle before the Local Ready (-LRDY) signal and must be held until the Ready Return (-RDYRTN) signal is sampled as active. It is then driven high one clock period before being released. While the VL-Bus is inactive, this signal is held high by a 20K pull-up resistor located on the motherboard.
LCLK	**Local CPU Clock:** This signal is a 1x clock that follows the same phase as a 486-type CPU. It is always driven by the system logic or by the VL-Bus controller to all VL-Bus masters and targets. Its frequency range is up to 66 MHz. It is allowed to dynamically change frequencies if the system logic supports a dynamically changing clock. In the case of a 386 or other CPUs that use a clock running a 2x, the main clock must be divided down to a 1x clock. The rising edge of the clock signifies the change of CPU states.
–LDEV <*x*>	**Local Device:** This signal is output by the VL-Bus target and signals the VL-Bus controller that the current cycle is a VL-Bus cycle. Each slot or device has its own Local Device (-LDEV) signal associated with that slot or device. The VL-Bus controller samples this signal on the rising edge of the Local CPU Clock (LCLK) signal one cycle after -ADS or two LCLK cycles after -ADS. If the system I/O bus controller detects -LDEV<*x*> asserted, the current cycle does not start a system I/O bus cycle. The VL-Bus controller may optionally start a VL-Bus transfer even before sampling -LDEV<*x*> asserted if it knows a cycle is owned by a LBT (for example via registers intern to the VL-Bus controller). For cache-hit and system DRAM cycles, -LDEV is ignored. Because there is one -LDEV<*x*> signal per slot, all VL-Bus devices must drive -LDEV<*x*> to valid TTL levels at all times.

Table 3-8. P3 VL-Bus Signal Descriptions (Continued)

Signal	Definition
-LEADS	**Local External Address Data Strobe**: The VL-Bus controller or active VL-Bus master asserts this signal whenever an address is present on the VL-Bus that performs a CPU cache invalidation cycle. A VL-Bus master must drive this signal while it owns the bus. A VL-Bus master with an internal cache may use this signal to invalidate its cache. This signal is not active for CPU writes, but the LBM can use the Address Data Strobe (-ADS) signal to invalidate CPU write transfers.
-LGNT<*x*>	**Local Bus Grant:** This signal is used in conjunction with the Local Request (-LREQ<*x*>) signal to establish a VL-Bus bus arbitration protocol. When the VL-Bus device asserts -LREQ<*x*>, the VL-Bus controller responds by asserting the Local Bus Grant (-LGNT<*x*>) signal for that slot. The active VL-Bus master then has control of the VL-Bus and may own the bus until it no longer needs the bus or the VL-Bus controller removes -LGNT<*x*> to preempt the active VL-Bus master. There is one pair of -LREQ and -LGNT signals per slot.
-LKEN	**Local Cache Enable:** This active low signal is asserted if the current VL-Bus transfer is cacheable. It is always driver by the VL-Bus controller. If it is asserted one clock before -LRDY and held until -RDYRTN is asserted during the last read in a cache line, the line is placed in the CPU cache. The system designer must determine how the VL-Bus controller decides which transfers are cacheable.
–LRDY	**Local Ready:** This active low signal begins the handshake that terminates the current active bus cycle. It is shared among all VL-Bus devices. The active LBT drives this signal only during the time of the cycle that it has claimed as its own. While the VL-Bus is inactive, this signal is held high by a 20K pull-up resistor located on the motherboard. Because the VL-Bus is normally a not-ready bus, the CPU must wait until this signal is asserted low to terminate an active VL-Bus cycle. This signal must be asserted low for one LCLK period. It is then driven high one clock period before being released. It is synchronized to LCLK so appropriate setup and hold times to LCLK must be satisfied. In most cases, -LRDY should not be driven the first T2 period because the system cache controller may be driving it. However, if the High Speed Write configuration bit is set the LBT may start driving this signal during the first T2 state.
–LREQ<*x*>	**Local Request:** This active low signal is used in conjunction with the Local Bus Grant (-LGNT<*x*>) signal. It is used by a VL-Bus device to gain control of the VL-Bus and become an active LBM. There is one pair of -LREQ and -LGNT signals per slot. LBTs that are never a bus master should leave this signal not connected. The motherboard pulls this signal high using a 20K pull-up resistor. When the VL-Bus device asserts this signal, the VL-Bus controller responds by asserting the -LGNT<*x*> signal for that slot. The VL-Bus device then has control of the VL-Bus and may hold the bus until the VL-Bus controller removes -LGNT<*x*>
–M/IO	**Memory or I/O Status**: This CPU output indicates the type of access currently executing on the VL-Bus. On the motherboard, the -M/IO signal can either be directly connected to the CPU -M/IO signal, or tri-state buffered. A memory cycle is indicated by -M/IO high, while an I/O cycle is indicated by -M/IO low. This signal is driven by the CPU for all CPU-initiated transfers. During system I/O bus master or DMA cycles the VL-Bus controller drives this signal according to the values of system I/O bus signals -MEMR, -MEMW, -IOR, and -IOW. During VL-Bus master transfers, the VL-Bus device acting as bus master drives this signal.

Table 3-8. P3 VL-Bus Signal Descriptions (Continued)

Signal	Definition
–RDYRTN	**Ready Return:** This signal establishes a handshake so that the VL-Bus target knows when the cycle has ended. Using this signal is equivalent to the Ready (-RDY) signal that is tied directly to the CPU or cache controller. It is always driven by the VL-Bus controller to all VL-Bus masters and targets. For LCLK speeds up to 33 MHz, this signal is typically asserted in the same LCLK cycle as -LRDY is asserted. At higher LCLK speeds it may trail the -LRDY signal by one LCLK cycle due to signal resynchronization. During DMA or system I/O bus master signals, the VL-Bus controller asserts this signal for one LCLK cycle when the DMA or system I/O bus master command ends.
-RESET	**System Reset:** This active low signal is a master reset that is asserted after system power up and before any valid CPU cycles take place. It is always driver by the system logic or the VL-Bus controller to all VL-Bus masters and targets. It places all devices at a known state before execution begins. Unlike the 386 CPU reset, there is no guaranteed relationship between the rising or falling edges of this signal and the phase of LCLK.
-WBACK	**Write Back:** This VL-Bus controller output is used to maintain cache coherency in systems that have a cache structure that requires this function. An example of this is a system with a CPU containing a write back cache. The VL-Bus controller may assert WBACK # at any time after an ASD# is issued and before or coincident with the first ready (either RDYRTN# or BRDY#) of that assert. When an active VL-Master samples WBACK# asserted, it must immediately abort the bus cycle and flat all address, data, and control signals that it drives as master. When WBACK# is sample inactive, the VL-Master restarts the bus cycle with a new ADS#. If a ready was returned at the same time as the WBACK# was sampled active, the ready (as well as the data on a read) should be ignored. WBACK# may be generated on either a read or a write and is synchronous to LCLK. It should always be sampled on a rising clock edge, and not be used asynchronously.
–W/R	**Write or Read Status:** This CPU output indicates the type of access currently executing on the VL-Bus. On the motherboard, this signal can either be directly connected to the CPU -W/R signal, or tri-state buffered. A write access is indicated by -W/R high, while a read access is indicated by -W/R low. This signal is driven by the CPU for all CPU-initiated transfers. During system I/O bus master or DMA cycles the VL-Bus controller drives this signal according to the values of system I/O bus signals -MEMR, -MEMW, -IOR, and -IOW. During VL-Bus transfers the VL-Bus device acting as bus master drives this signal.

Note: Every VL-Bus device must connect to all power and ground pins. Power must be drawn equally from all power pins. To ensure future compatibility, reserved pins must not be connected to any signal on either the VL-Bus controller (i.e., the motherboard) or any VL-Bus add-in boards.

3-3

VL-Bus Timing Descriptions

General Overview of a VL-Bus Transfer

A VL-Bus CPU transfer begins when valid address and status information is placed on ADR<31..02>, M/IO#, W/R#, D/C# and BE<3..0>#. ADS# is strobed to begin the transfer. The VL-Bus target has 20ns to recognize the address to which it should respond and assert LDEV#. Depending on the host CPU speed and VL-Bus controller design, LDEV# is generally sampled at either the end of the first T2 or second T2 for CPU or Local Bus Master initiated transfer. If LDEV# is asserted, the system I/O bus controller does not start a system I/O bus cycle.

The VL-Bus target begins to drive LRDY# (asserted or negated, depending on whether additional wait states are needed) after ADS# is negated. When the VL-Bus target has completed the transfer, it asserts LRDY# for one LCLK cycle and then negates LRDY# for one-half LCLK cycle prior to the VL-Bus target releasing LRDY#. When the VL-Bus controller detects LRDY# asserted, it may immediately assert RDYRTN# or it may resynchronize LRDY# and assert RDYRTN# on the next LCLK cycle. If the current transfer is a read, the VL-Bus target must hold the read data on the data bus until the LCLK which RDYRTN# is sampled asserted.

RDYRTN# is equivalent to the 486 CPU RDY#. Slower systems (i.e., 33 MHz or less) do not require LRDY# to be resynchronized. Faster systems may resynchronize LRDY# before asserting RDYRTN#, causing CPU RDY# to appear one clock cycle after LRDY#. During a DMA or system I/O bus master read cycle, the end of the cycle is asynchronous to the CPU clock; therefore RDYRTN# (synchronized to LCLK) may appear any number of clock cycles after LRDY#. The VL-Bus target must always watch RDYRTN# during read operations, which it terminates with LRDY# and hold data on the data bus until RDYRTN# is sampled asserted.

Arbitration Timing

Figure 3-4 shows the timing a VL-Bus Master and VL-Bus controller must meet. The preempt acknowledge allows for up to 5 usec delay; the active VL-Bus master should complete the current transfer (or set of transfers if locked cycles) and then relinquish control of the VL-Bus back to the VL-Bus controller. A bus master that does not support preempting may request control of the VL-Bus but must release the bus within 5 usec. The normal maximum latency between a VL-Master asserting LREQ# and the system controller granting the bus by asserting LGNT# is 20uS. At times, this latency will be exceeded, specifically if an ISA master holds the bus for long periods; many wait states are added to ISA bus accesses by the CPU, etc. 20uS is a good number to be used in determining FIFO size, but a design should be aware that it could be exceeded. The card could presumably do a retry in these cases. The system controller should be designed such that a VL-Master is served within 20uS the vast majority of the time, as far as it has control.

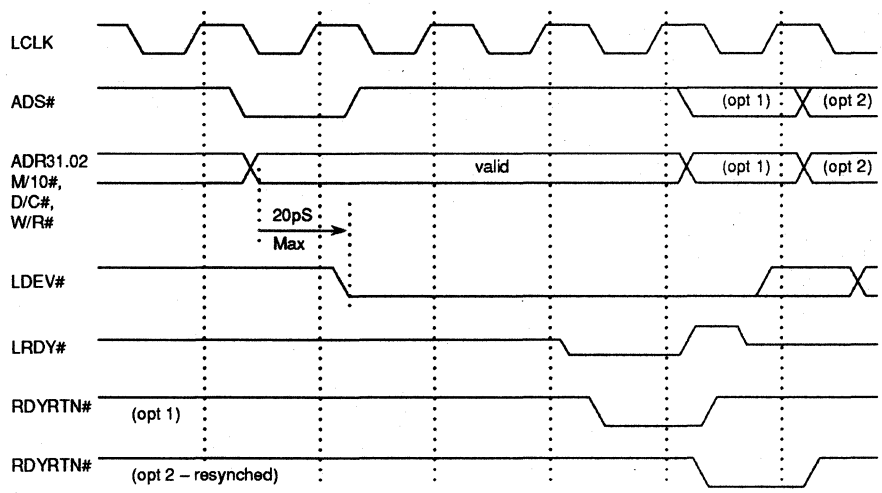
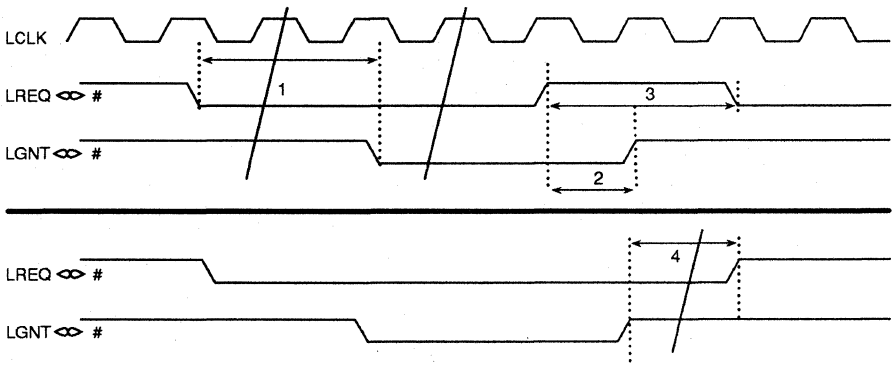
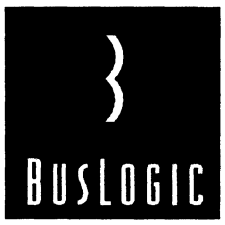


Figure 3-3. Simplified Timing Diagram



No.	Description	Min.	Max.	Units
1	LREQ# asserted to LGNT# asserted	1		LCLK
2	LREQ# negated to LGNT# negated	1	2	LCLK
3	LREQ# negated to LREQ# asserted	2	–	LCLK
4	LGNT# negated to LREQ# negated	–	5	usec

Figure 3-4. LBM Arbitration Timing



PART 4

EISA HOST ADAPTERS

Part 4: Contents

List of Figures

Section 4•1: Product Overview

Section 4•2: Electrical Interface

List of Tables

Section 4•2: Electrical Interface

Section 4•3: EISA Bus Master Timing Diagrams

4•1

Product Overview

The BusLogic BT-746/747C (BT-74X) host adapters are intelligent PC/AT to SCSI bus master host adapter products based on BusLogic-designed MultiMaster ASIC technology. Designed for DOS/Windows and multitasking applications such as Windows NT, NetWare, OS/2, UNIX™ and XENIX™, the intelligent BT-74X provides a high-performance interconnection between the EISA (Extended Industry Standard Architecture) bus and Small Computer System Interface (SCSI) peripheral devices.

BusLogic has embedded driver support in most popular operating systems. No additional drivers are needed when your system runs with the current versions of the following operating systems:

- NetWare 3.12/4.X
- Windows NT
- Interactive UNIX
- SCO UNIX
- IBM OS/2
- UNIXWare
- Solaris (for *x*86)
- Vines
- NextStep

Software drivers for NetWare 3.11, OS/2, SCO/UNIX and DOS are also available separately.

A BusLogic-designed bus master controller ASIC, an advanced SCSI controller chip and a 16-bit microprocessor chip provide higher speed, lower power consumption, fewer parts and higher reliability.

The BT-74X supports a full 32-bit address path, and can access up to four Gigabytes of system memory. The total memory supported is thus limited only by the packaging constraints of the individual product, rather than by the system architecture. Bus master 8-, 16-, or 32-bit data transfers are performed at speeds up to 33 MBytes/sec on the EISA bus.

The BT-74X supports single-ended SCSI drives with asynchronous data rates of up to 7 MBytes/sec and synchronous data rates of up to 10 MBytes/sec with the proper termination and cabling. The BT-7X uses an integrated controller to provide active termination on the SCSI bus.

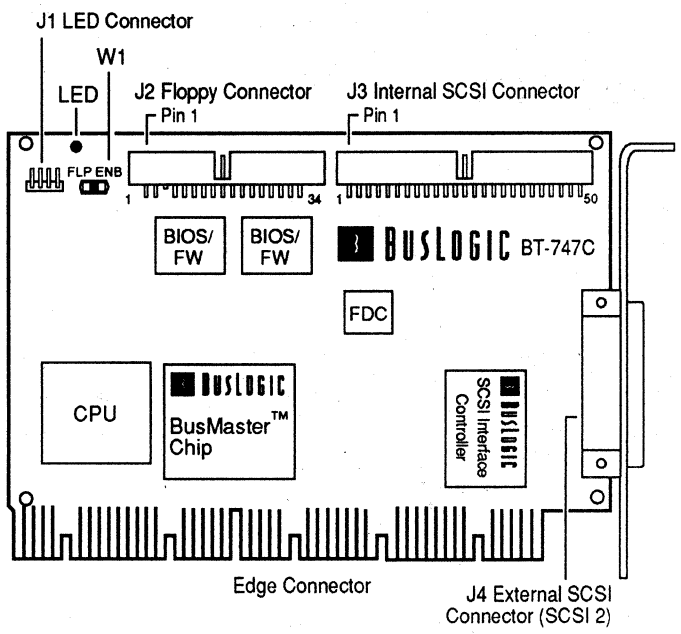


Figure 4-1. BT-747C Host Adapter

The BT-746C is the same as the BT-747C but without floppy support.

Both internal and external 50-pin connectors are included on the board for flexibility in attaching SCSI devices to the system. An on-board floppy controller chip on the BT-747C independently communicates with any combination of up to two 3.5" or 5.25" floppy drives. BT-74X adapters can also support the newly emerging 2.88 MB floppies.

Specifications

Dimensions:	6.25" x 4.5"
Electrical:	
Operating Voltage	5±0.25V
Operating Current	.5A Max.
Max. Ripple/Noise	100 mV
Environmental:	
Temperature	0°C to 60°C (32°F to 128°F)
Relative Humidity	10% to 95% non-condensing
Altitude	0 to 10,000 ft. operating
	0 to 15,000 ft. non-operating
Interface Connections:	
SCSI Internal	50-pin double-row connector
SCSI External	50-pin shielded SCSI connector
Floppy	34-pin AT-compatible ribbon style (BT-747C only)
System Interface	EISA standard edge connector
MTBF	90,000 hours

4•2

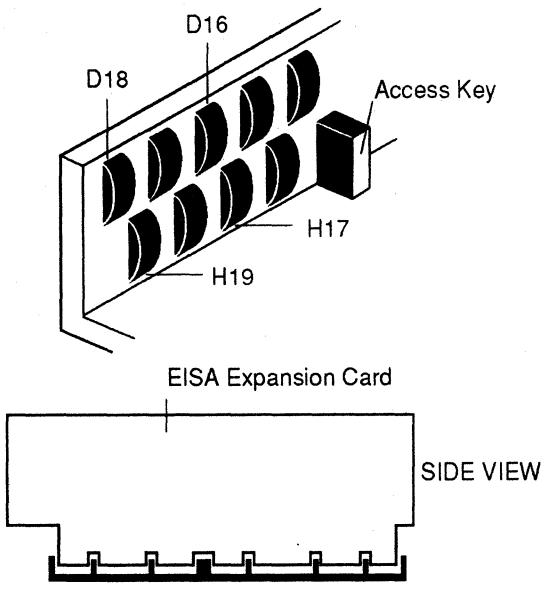
Electrical Interface

This section provides the user with a complete description of the name, function, and applicable logic level of all signals between the BT-74X and the EISA host system. It also describes the signals processed by the SCSI protocol chip and the floppy controller chip.

EISA System Bus Electrical Interface

The BT-74X is electrically and mechanically compatible with the Input/Output (I/O) bus used in EISA computers. Physically, this Input/Output bus is contained on two card edge connectors. The bus master control logic on the BT-74X controls the EISA system bus arbitration and data transfer operations. During bus master data transfers, the BT-74X takes control of the system bus and transfers data directly to and from the main system memory. Both odd and even starting addresses are supported by the BT-74X.

The EISA system I/O bus provides the necessary hardware interface to the host Central Processing Unit (CPU) to allow it to communicate with the BT-74X. Figure 4-2 illustrates the interface between the EISA connector on the system bus and the BT-74X. Figure 4-3 identifies the positions of connector rows on the BT-74X's board edge.



Access Key notches on the bottom of the BT-74X Host Adapter card match the Access Keys in the EISA connector.

Figure 4-2. Interface between the EISA Connector on the System Bus and the BT-74X

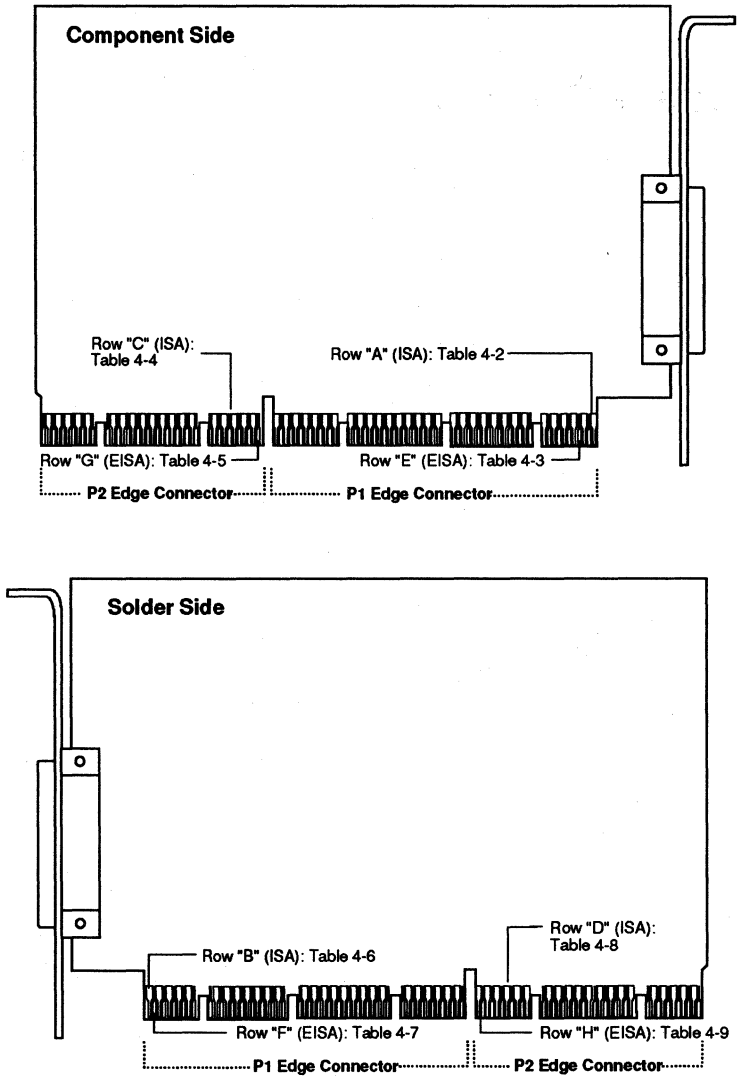


Figure 4-3. Connector Rows on the BT-74X's Board Edge

Summary of EISA Signals

The following tables show the additional signals added to the ISA connectors to provide support for the EISA bus.

Table 4-1. EISA Signals

Bus Pins	Signal Name	Description
16	D<31:16>	Data lines
8	- LA<31:24>	Address lines
15	LA<16:2>	Address lines
4	- BE<3:0>	Byte enables
1	- LOCK	Bus lock
1	- EX32	32-bit EISA slave indicator
1	- EX16	16-bit EISA slave indicator
1	- START	EISA start of cycle control
1	- CMD	EISA end of cycle control
1	M-IO	EISA memory or I/O indicator
1	W-R	EISA write or read indicator
1	EXRDY	EISA ready indicator
1	- MREQx	Slot specific bus request
1	- MAKx	Slot specific bus grant
1	- SLBURST	Burst cycle indicator from slave
1	- MSBURST	Burst cycle control from master
55		Total additional pins on EISA connector

Tables 4-2 through 4-5 summarize pin assignments for the I/O channel connectors on both sides of P1, a 62-pin edge connector.

Table 4-2. P1 Component Side Row "A" Edge Connector (ISA)

Signal Pin #	Signal Name	Direction
A1	- IOCHK	Input
A2	D7	I/O
A3	D6	I/O
A4	D5	I/O
A5	D4	I/O
A6	D3	I/O
A7	D2	I/O
A8	D1	I/O
A9	D0	I/O
A10	IOCHRDY	Not Used
A11	AENx	Input
A12	SA19	Not Used
A13	SA18	Not Used
A14	SA17	Not Used
A15	SA16	Not Used
A16	SA15	Not Used
A17	SA14	Not Used
A18	SA13	I/O
A19	SA12	I/O
A20	SA11	I/O
A21	SA10	I/O
A22	SA9	I/O
A23	SA8	I/O
A24	SA7	I/O
A25	SA6	I/O
A26	SA5	I/O
A27	SA4	I/O
A28	SA3	I/O
A29	SA2	I/O
A30	SA1	I/O
A31	SA0	I/O

Table 4-3. P1 Component Side Row "E" Edge Connector (EISA)

Signal Pin #	Signal Name	Direction
E1	- CMD	I/O
E2	- START	I/O
E3	EXRDY	I/O
E4	- EX32	I/O
E5	GND	Ground
E6	Access Key	Access Key
E7	- EX16	Not Used
E8	- SLBURST	Input
E9	- MSBURST	Output
E10	W-R	I/O
E11	GND	Ground
E12	ENB_RQT	Output
E13	RESERVED	Not Used
E14	RESERVED	Not Used
E15	GND	Ground
E16	Access Key	Access Key
E17	- BE<1>	I/O
E18	- LA<31>	I/O
E19	GND	Ground
E20	- LA<30>	I/O
E21	- LA<28>	I/O
E22	- LA<27>	I/O
E23	- LA<25>	I/O
E24	GND	Ground
E25	Access Key	Access Key
E26	LA<15>	I/O
E27	LA<13>	I/O
E28	LA<12>	I/O
E29	LA<11>	I/O
E30	GND	Ground
E31	LA<9>	I/O

Table 4-4. P2 Component Side Row "C" Edge Connector (ISA)

Signal Pxin #	Signal Name	Direction
C1	- SBHE	Not Used
C2	LA23	I/O
C3	LA22	I/O
C4	LA21	I/O
C5	LA20	I/O
C6	LA19	I/O
C7	LA18	I/O
C8	LA17	I/O
C9	- MEMR	Input
C10	- MEMW	Not Used
C11	D8	I/O
C12	D9	I/O
C13	D10	I/O
C14	D11	I/O
C15	D12	I/O
C16	D13	I/O
C17	D14	I/O
C18	D15	I/O

Table 4-5. P2 Component Side Row "G" Edge Connector (EISA)

Signal Pin #	Signal Name	Direction
G1	LA<7>	I/O
G2	GND	Ground
G3	LA<4>	I/O
G4	LA<3>	I/O
G5	GND	Ground
G6	Access Key	Access Key
G7	D<17>	I/O
G8	D<19>	I/O
G9	D<20>	I/O
G10	D<22>	I/O
G11	GND	Ground
G12	D<25>	I/O
G13	D<26>	I/O
G14	D<28>	I/O
G15	Access Key	Access Key
G16	GND	Ground

Table 4-5. P2 Component Side Row "G" Edge Connector (EISA) (Continued)

Signal Pin #	Signal Name	Direction
G17	D<30>	I/O
G18	D<31>	I/O
G19	- MREQx	Output

Tables 4-6 through 4-9 summarize pin assignments for the I/O channel connectors on both sides of P2, a 36-pin edge connector.

Table 4-6. P1 Solder Side Row "B" Edge Connector (ISA)

Signal Pin #	Signal Name	Direction
B1	GND	Ground
B2	RESET DRV	Input
B3	+5 Vdc	Power
B4	IRQ9	Output
B5	- 5 Vdc	Not Used
B6	DRQ2	Output
B7	-12 Vdc	Not Used
B8	- ØWS	Not Used
B9	+12 Vdc	Not Used
B10	GND	Ground
B11	- SMEMW	Not Used
B12	- SMEMR	Input
B13	- IOW	I/O
B14	- IOR	I/O
B15	- DACK3	Not Used
B16	DRQ3	Not Used
B17	- DACK1	Not Used
B18	DRQ1	Not Used
B19	- Refresh	Not Used
B20	BCLK	Input
B21	IRQ7	Not Used
B22	IRQ6	Output
B23	IRQ5	Not Used
B24	IRQ4	Not Used
B25	IRQ3	Not Used
B26	- DACK2	Input
B27	T/C	Input
B28	BALE	Not Used
B29	+5 Vdc	Power
B30	OSC	Not Used
B31	GND	Ground

Table 4-7. P1 Solder Side Row "F" Edge Connector (EISA)

Signal Pin #	Signal Name	Direction
F1	GND	Ground
F2	+ 5 Vdc	Power
F3	+ 5 Vdc	Power
F4	RESERVED	Not Used
F5	RESERVED	Not Used
F6	Access Key	Access Key
F7	RESERVED	Not Used
F8	RESERVED	Not Used
F9	+ 12 Vdc	Not Used
F10	M-10	I/O
F11	- LOCK	I/O
F12	RESERVED	Not Used
F13	GND	Ground
F14	RESERVED	Not Used
F15	- BE<3>	I/O
F16	Access Key	Access Key
F17	- BE<2>	I/O
F18	- BE<0>	I/O
F19	GND	Ground
F20	+ 5 Vdc	Power
F21	- LA<29>	I/O
F22	GND	Ground
F23	- LA<26>	I/O
F24	- LA<24>	I/O
F25	Access Key	Access Key
F26	LA<16>	I/O
F27	LA<14>	I/O
F28	+ 5 Vdc	Power
F29	+ 5 Vdc	Power
F30	GND	Ground
F31	LA<10>	I/O

Table 4-8. P2 Solder Side Row "D" Edge Connector (ISA)

Signal Pin #	Signal Name	Direction
D1	- MEM CS16	Not Used
D2	- I/O CS 16	Not Used
D3	IRQ10	Output
D4	IRQ11	Output
D5	IRQ12	Output
D6	IRQ15	Output
D7	IRQ14	Output
D8	- DACK0	Not Used
D9	DRQ0	Not Used
D10	- DACK5	Not Used
D11	DRQ5	Not Used
D12	- DACK6	Not Used
D13	DRQ6	Not Used
D14	- DACK7	Not Used
D15	DRQ7	Not Used
D16	+5 Vdc	Power
D17	- MASTER16	Not Used
D18	GND	GND

Table 4-9. P2 Solder Side Row "H" Edge Connector (EISA)

Signal Pin #	Signal Name	Direction
H1	LA<8>	I/O
H2	LA<6>	I/O
H3	LA<5>	I/O
H4	+ 5 Vdc	Power
H5	LA<2>	I/O
H6	Access Key	Access Key
H7	D<16>	I/O
H8	D<18>	I/O
H9	GND	Ground
H10	D<21>	I/O
H11	D<23>	I/O
H12	D<24>	I/O
H13	GND	Ground
H14	D<27>	I/O
H15	Access Key	Access Key
H16	D<29>	I/O

Table 4-9. P2 Solder Side Row "H" Edge Connector (EISA) (Continued)

Signal Pin #	Signal Name	Direction
H17	+ 5 Vdc	Power
H18	+ 5 Vdc	Power
H19	- MAKx	Input

P1 and P2 Input/Output Signal Descriptions

This section describes signals from each connector of the EISA bus signals. I/O adapters should be designed with a maximum of two low-power Shottky (LS) loads per line. Signals preceded by a hyphen (-) indicate an active low signal.

Table 4-10. P1 & P2 Signal Descriptions

Signal	Definition
AEN(x)	**Address Enable (x)**: When this slot-specific (the "x" refers to the slot number) signal is deasserted, it indicates that an I/O slave may respond to addresses and I/O commands on the bus. The **AEN(x)** signal is asserted during DMA cycles to prevent I/O slaves from misinterpreting DMA cycles as valid I/O cycles. The system board must deassert the **AEN(x)** signal when the Start signal is asserted for an I/O access. The **AEN(x)** signal must remain deasserted until after the Command signal is asserted. The **AEN(x)** signal is also used to disable I/O accesses to all other option slots during accesses to a particular slot's slot-specific I/O address range.
BALE	**Buffered Address Latch Enable**: This signal indicates that a valid address is present on the Latchable Address Lines 2-31. The Latchable Address Lines 2-31 or any decodes developed from them by ISA devices are latched (with transparent latches) on the trailing edge of this signal if the address is needed for the whole cycle. This signal is always asserted during a DMA or 16-bit ISA bus master operation. EISA devices should not use this signal to latch addresses; the trailing edge of the Start command or the leading edge of the Command signal should be used.
-BE0 — -BE3	**Byte Enable 0-3:** These signals are the byte enable signals that identify the specific bytes addressed in a double word. They are pipelined from one cycle to the next and must be latched by the addressed slave if required for the whole cycle. The timing of these signals varies depending on the cycle type. During normal cycles, they go valid before the **BALE** signal goes active and remain valid as long as the Latchable Address Lines 3-31 remain valid. During DMA or 16-bit ISA bus master cycles, they go valid at least 1/2 bus clock cycle before the Command signal or the ISA command signals go active.
BCLK	**Bus Clock:** This signal is provided to synchronize events with the main system clock. It operates at a frequency between 8.333 MHz and 6 MHz with a normal duty cycle of 50 percent. It is driven only by the system board. It's period is sometimes extended for synchronization to the main CPU or other system board devices. For example, the Compressed cycle type extends each bus clock period by holding this signal low for half a cycle beyond the normal transition to high.
-CMD	**Command:** This signal provides timing control within the cycle. The system board asserts it on the rising edge of the **BCLK** signal, simultaneously with negation of the Start signal. The system board holds this signal asserted until the end of the cycle. The end of the cycle normally is synchronized with the rising edge of the **BCLK** signal, but in certain cases is asynchronous. A bus master does not drive this signal.

Table 4-10. P1 & P2 Signal Descriptions (Continued)

Signal	Definition
-DAK2	**DMA Acknowledge 2:** The system board asserts a DMA channel's **-DAK2** signal to indicate that the channel has been granted the bus. A DMA device is selected if it decodes the **-DAK2** signal with the I/O Read Cycle signal or the IO Write Cycle signal asserted. The **-DAK2** signal can also be used to acknowledge grant of bus access to a 16-bit ISA bus master. The bus master must assert the 16-Bit Bus Master Transfer signal after sampling the **-DAK2** signal asserted. Address and cycle control signals must be floated and the 16-Bit Bus Master. The transfer signal must be negated before the system board negates the **-DAK2** signal. For EISA block or demand mode DMA transfers, the **-DAK2** signal remains asserted until the transfer is completed or until the centralized arbitration controller preempts the DMA process. The preemption occurs after another device requests the bus and 4 µs elapse.
DRQ2	**DMA Request 2:** These signals are used to request a DMA service from the DMA subsystem or for a 16-bit ISA bus master to request access to the system bus. The request is made when the **DRQ2** signal is asserted. The system board allows the **DRQ2** signal to be asserted asynchronously. The requesting device must hold the **DRQ2** signal asserted until the system board asserts the appropriate **DRQ2** signal. For demand mode DMA memory-read I/O-write cycles, the **DRQ2** signal is sampled on the rising edge of the **BCLK** signal, one bus clock cycles from the end of the cycle (the rising edge of the I/O Write Cycle signal). For demand mode DMA memory-write I/O-read cycles, the **DRQ2** signal is sampled on the rising edge of the **BCLK** signal, 1.5 bus clock cycles from the end of the cycle (the rising edge of the I/O Read Cycle signal). For demand mode burst DMA, the **DRQ2** signal is sampled each cycle on the rising edge of the **BCLK** signal. For 16-bit ISA bus masters, the **DRQ2** signal is sampled on the rising edge of the **BCLK** signal, two bus clock cycles before the system board negates the **DRQ2** signal. The trailing edge of the **DRQ2** signal must meet the set-up and hold time to the sampling point for proper system operation.
D24 — D31	**Data Bits 24-31:** These signals are the highest-order 8 bits of the 32-bit EISA data bus. A 32-bit device uses these signals to transfer the fourth (highest) byte of a double word when the **-BE3** signal is asserted.
D16 — D23	**Data Bits 16-23:** These signals are the second highest-order 8 bits of the 32-bit EISA data bus. A 32-bit device uses these signals to transfer the third (second highest) byte of a double word when the **-BE2** signal is asserted.
D8 — D15	**Data Bits 8-15:** These signals are the high 8 bits of the 16-bit data bus. These signals are used by 16-bit devices to transfer the high half of a data word when the System Bus High Enable signal, the **-BE3** signal or the **-BE1** signal is asserted. The Data Bits 8-15 signals are used by 32-bit devices to transfer the second (third highest) byte of a double word when the Byte Enable 1 signal is asserted.
D0 — D7	**Data Bits 0-7:** These signals are the low 8 bits of the data bus. These signals are used by 8-bit devices to transfer data. These signals are used by 16-bit devices to transfer the lower half of a data word when the System Address 0 signal is deasserted, or when the **-BE2** signal or the **-BE0** signal is asserted. The Data Bits 0-7 signals are used by 32-bit devices to transfer the first (lowest byte of a double word) when the **-BE0** signal is asserted.

Table 4-10. P1 & P2 Signal Descriptions (Continued)

Signal	Definition
EXRDY	**EISA Ready:** EISA I/O and memory slaves deassert this signal to request Wait state timing (each Wait state is one bus clock cycle). The system board samples the **EXRDY** signal on each falling edge of the **BCLK** signal after it asserts the Command signal. The system board holds the Command signal asserted during the entire period that the **EXRDY** signal is deasserted, and at least one half bus clock cycle after sampling the **EXRDY** signal asserted. The **EXRDY** signal must be driven with an open-collector type buffer (a system board pull-up resistor provides the asserting drive current). The EISA slave should deassert the **EXRDY** signal during the Start signal or on the rising edge of the **BCLK** signal at the end of the Start signal if Wail states are to be added. The slave *must* allow the **EXRDY** signal to float high (asserted) synchronously with the Buffered Address Latch Enable signal's falling edge and must not hold the **EXRDY** signal asserted longer than 2.5 µs. The **EXRDY** signal should *never* be driven high.
-EX32	**EISA 32-Bit Transfer:** A memory or I/O slave asserts this signal to indicate that it supports 32-bit (double word) transfers. A two bus clock cycle is executed when a slave asserts this signal during a memory access. The slave asserts this signal after decoding a valid address on the Latchable Address Lines 2-31 and the Memory-Input Output signal. This signal should not be latched by the slave. Both 16- and 32-bit EISA bus masters must monitor this signal at the trailing edge of the Start signal to determine if the slave supports 32- (and 16-) bit EISA transfers (asserted), or if the system board is performing data size translation (deasserted). If data size translation is being performed and the master is a 32-bit master, then the system board asserts this signal to indicate the completion of the translation.
-EX16	**EISA 16-Bit Transfer:** An EISA memory or I/O slave asserts this signal to indicate that it supports 16-bit (word) transfers. A 16-bit EISA bus master samples this signal asserted to confirm a 16-bit EISA slave. An EISA cycle (two bus clock cycles) is executed when a slave asserts this signal during a memory access by the system board or a 16-bit EISA bus master. The slave asserts this signal after decoding a valid address on the Latchable Address Lines 2-31 and the Memory-Input Output signal. This signal should not be latched by the slave. This signal must be monitored by 16-bit EISA bus masters to determine if the slave supports 16-bit EISA transfers (asserted), or if the system board is performing data size translation (deasserted). If data size translation is being performed (ISA cycles) and the master is a 16-bit master (indicated by the master asserting the 16-Bit Bus Master Transfer signal), then the system board asserts this signal to indicate completion of the translation.
-IOCHK	**I/O Check:** An EISA or ISA expansion board can assert this signal to indicate to the main CPU that a serious error has occurred. Parity errors and uncorrectable system errors exemplify problems that might cause an expansion board to assert this signal.
-IORC	**I/O Read Cycle:** A DMA device can drive data on the data bus after sampling this signal asserted. An ISA I/O slave drives data onto the bus while this signal is asserted and the **AEN(x)** signal is deasserted. The device must hold the data valid until sampling the **-IORC** signal negated.
-IOWC	**I/O Write Cycle:** A DMA device can latch data from the data bus when this signal is asserted. An ISA I/O slave latches data from the data bus when this signal is asserted and the **AEN(x)** signal is deasserted (low). The main CPU or bus master must drive valid data on the bus before asserting this signal.

Table 4-10. P1 & P2 Signal Descriptions (Continued)

Signal	Definition
IRQ14-15, IRQ9-12, & IRQ6	**Interrupt Request**: These signals are used to interrupt the CPU to request some service. In compatible mode, the interrupt is recognized when the **IRQ** (x) signal goes from a low to a high and remains there until the appropriate interrupt service routine is executed. If programmed to level-sensitive mode, the interrupt is recognized when the **IRQ**(x) signal is asserted (low). Another interrupt is generated at the end of the interrupt service routine if the **IRQ** (x) signal is still held low, allowing a single line to be shared by more than one device. The BT-74X asserts an interrupt request in compatible mode.
LA2 — LA16	**Latchable Address Lines 2-16:** These signals are a part of the latchable address bus. The Latchable Address Lines 2-31 are pipelined from one cycle to the next and must be latched by the addressed slave if required for the whole cycle. The Latchable Address Lines 2-31 are presented early enough in the cycle decode to support 1.5 or 2 bus clock cycle memory accesses. During standard cycles, they go valid before the Start signal is asserted and remain valid at least 1/2 bus clock cycle after the Command signal or the ISA command signals are asserted. During DMA or 16-bit ISA bus master cycles, the Latchable Address Lines 2-31 are valid at least one bus clock cycle before the Command signal or ISA command signals are asserted. The Latchable Address Lines 2-31 can be driven by an expansion board acting as a bus master. An EISA slave may latch the entire address (the Latchable Address Lines 2-31 and the Byte Enable 0-3 signals) and status signals (the Memory -Input Output signal and the Write/-Read signal) on the trailing edge of the Start signal or leading edge of the Command signal.
LA17 — LA23	**Latchable Address Lines 17-23:** These signals are a part of the 32-bit latchable address bus. They have the same characteristics as the Latchable Address Lines 2-16 except that they are wired to the 16-bit portion of the ISA connector. An ISA slave can latch the Latchable Address Lines 17-23 with the trailing edge of the **BALE** signal.
-LA24 — -LA31	**Latchable Address Lines 24-31:** These signals are the highest byte of the 32-bit latchable address bus. They have the same characteristics as the Latchable Address Lines 2-16 except that they use inverted logic. A high on a Latchable Address Line 24-31 must be interpreted as an address bit of "0". A low must be interpreted as an address bit of "1". (When the notation -LA2 — -LA31 is used, only LA24 — LA31 are active low, the next are active high.)
-LOCK	**Lock:** The main CPU or a bus master may assert this signal to guarantee exclusive memory access during the time this signal is asserted. A bus master may also assert this signal to guarantee exclusive I/O access during the time this signal is asserted. Assertion of this signal allows bit test-and-set operations (as used for semaphores) to be executed as a unit, with the bus lock preventing multiple devices from simultaneously modifying the semaphore bit.
- MAK(x)	**Master (x) Acknowledge:** This signal is a slot-specific signal that is asserted by the system board to grant bus access to an EISA bus master. The "x" refers to the slot number. The **- MAK(x)** signal is asserted from the rising edge of the **BCLK** signal and the bus master can begin driving the Latchable Address Lines 2-31, the Byte Enable 0-3 signals, the Master Burst Cycle signal, the Start signal, the Memory -Input Output signal, and the Write/-Read signal on the next falling edge of the **BCLK** signal. The system board negates the **-MAK(x)** signal on the rising edge of the **BCLK** signal after sampling the Master (x) Request signal negated. The system board can also negate this signal to indicate to an active bus master that another device has requested the bus. The bus master must negate the Master (x) Request signal to release the bus within 64 bus clock cycles (8 µs) of sampling the **- MAK(x)** signal negated.

Table 4-10. P1 & P2 Signal Descriptions (Continued)

Signal	Definition
-MASTER16	**16-Bit Bus Master Transfer:** A bus master asserts the **-MASTER16** signal to indicate 16-bit data size. A bus master can assert this signal after the system board asserts the **-DAK2** signal or the- **MAK(x)** signal. The 16-bit EISA bus master negates this signal after completing the last transfer. An ISA master negates the **-MASTER16** signal immediately when the system board negates the **-DAK2** signal. A 32-bit bus master can assert this signal during the Start signal to disable automatic 32-to-16-bit data size translation for 16-bit EISA memory burst slaves. It can then perform 16-bit Burst cycles to a 16-bit EISA slave.
-MREQ(x)	**Master (x) Request:** This signal is a slot-specific signal used by EISA bus masters to request bus access. The "x" refers to the slot number. Bus masters requiring use of the bus must assert this signal until the system board grants bus access by asserting the **- MAK(x)** signal. The requesting device must hold the **-MREQ(x)** signal asserted until the system board asserts the appropriate **- MAK(x)** signal. The system board samples the **-MREQ(x)** signal on the rising edge of the **BCLK** signal. If the **-MREQ(x)** signal is sampled asserted, the arbitration controller performs the arbitration and the system board asserts the **- MAK(x)** signal when the bus becomes available. The bus master can begin driving the bus with address and other signals on the falling edge of the **BCLK** signal when the **- MAK(x)** signal is sampled asserted. When a bus master completes a transfer, it can release the bus by negating the **- MAK(x)** signal on the falling edge of the **BCLK** signal. If no bus cycle is in progress when **- MAK(x)** signal is negated, the bus master must float the Latchable Address Lines 2-31, the Byte Enable 0-3 signals, the Master Burst Cycle signal, the Lock signal, the Data Bits 0-31 signals, the Start signal, the Memory -Input Output signal, and the Write/-Read signal on or before the rising edge of the **BCLK** signal after the **- MAK(x)** signal is negated. If a cycle is in progress when the **- MAK(x)** signal is negated, then the Latchable Address Lines 2-31, the Byte Enable 0-3 signals, the Master Burst Cycle signal, the Lock signal, the Start signal, the Memory -Input Output signal, and the Write/-Read signal must be floated by the rising edge of the **BCLK** signal at the end of the cycle. The Data Bits 0-31 signals must be floated on (the EISA Ready signal's termination) on before (the EISA 32-Bit Transfer signal's or the EISA 16-Bit Transfer signal's termination) the falling edge of the **BCLK** signal after the end of the cycle. Cycle completion is indicated by the memory or I/O slave asserting the EISA Ready signal or the system board asserting the EISA 16-Bit Transfer signal or the EISA 32-Bit Transfer signal after completing bus conversions. A bus master must wait at least two bus clock cycles after releasing the bus before reasserting its **- MAK(x)** signal. The trailing edge of the **- MAK(x)** signal must meet the set-up and hold time to the sampling point for proper system operation.
-MSBURST	**Master Burst Cycle:** An EISA CPU or bus master asserts this signal to indicate to the slave (typically, main memory) that the CPU or bus master can provide Burst cycles. This signal is asserted with the Latchable Address Lines 2-31 for the second and all subsequent cycles of the burst and is sampled on the rising edge of the **BCLK** signal by the slave.
M-IO	**Memory -Input Output:** The main CPU or an EISA bus master asserts this signal to indicate the type of cycle in progress as a memory cycle (high) or I/O cycle (low). This signal is pipelined from one cycle to the next and is latched by the addressed slave if needed for the whole cycle. It should be included in all decodes by EISA slaves. It must not be used in decoding the signals -M16 or -IO16.

Table 4-10. P1 & P2 Signal Descriptions (Continued)

Signal	Definition
RESDRV	**Reset Hardware:** Assertion of this signal causes a hardware reset of ISA and EISA expansion boards. This signal is asserted by the reset controller during power up or after a bus time-out. This signal has a minimum pulse width equivalent to 9 bus clock cycles (the minimum time between two ISA I/O write cycles). All devices that can prevent operation of the CPU, memory or system board I/O must use this signal for hardware reset. Slaves that insert Wait states based on internal state machines, devices that require software initialization, and DMA devices are examples of hardware that reset after sampling the **RESDRV** signal asserted.
SA0 — SA13	**System Address Lines 0-13:** These signals address memory or I/O within the system. They form the low-order 20 bits of the 32-bit address. On normal cycles these signals are driven onto the bus while the Buffered Address Latch Enable signal is high and are latched by the system board on the trailing edge of the Buffered Address Latch Enable signal. The System Address Lines 0-13 are valid throughout the bus command cycle.
-SBHE	**System Bus High Enable**: This signal enables a data transfer on the upper byte (D8-D15) of the data bus. The present bus master uses this signal to condition the data bus buffers driving D8-D15.
-SLBURST	**Slave Burst Cycle:** A slave (typically, main memory) indicates its support of Burst cycles by asserting this signal. The slave develops the **-SLBURST** signal from the Latchable Address Lines 10-31 and the **M-IO** signal and produces the **-SLBURST** signal regardless of the state of the **-MSBURST**. The **-SLBURST** signal is sampled on the rising edge of the **BCLK** signal by the main CPU, DMA controller or bus master.
-START	**Start:** This signal provides timing control at the start of a cycle. The CPU or bus master asserts this signal after the Latchable Address Lines 2-31 and the **M-IO** signal become valid and negates the Start command on a rising edge of the **BCLK** signal after one bus clock cycle. The Byte Enable 0-3 signals and the Write/-Read signal may not be valid at the leading edge of the Start signal.
TC	**Terminal Count**: This signal provides a pulse when the terminal count for a DMA channel is reached.
W-R	**Write/-Read:** This status signal identifies the cycle as a write (high) or read (low). It becomes valid after assertion of the **-START** signal (and before assertion of the Command signal). It remains valid as long as the Latchable Address Lines 2-31 are valid. It is driven from the same edge of the **BCLK** signal that activates the Start signal.

4•3

EISA Bus Master Timing Diagrams

Table 4-11. EISA Bus Timing Specifications

Name	Parameter	Min.	Max.	Unit
T1	BCLK high time	55		ns
T2	BCLK low time	55		ns
T3	BLCK period	120	250	ns
T4	- MREQ delay from BCLK falling	0	35	ns
T5	- MACK setup to BCLK falling	10		ns
T6	LA[0-31], M-IO delay from BCLK falling	2	40	ns
T7	LA[0-31], M-IO set up to -START low	20		ns
T8	LA[0-31], M-IO, -BE[0-3], W-R float delay	2	50	ns
T9	- BE[0-3],W-R, - LOCK valid from BCLK falling		40	ns
T10	- START delay from BCLK falling	0	25	ns
T11	- CMD delay from BCLK rising	0	25	ns
T12	- EX32, - EX16 set up to BCLK rising	25		
T13	- EX32, - EX16 hold from BCLK rising	50		ns
T14	- EXRDY set up to BCLK falling	15		ns
T15	- EXRDY hold from BCLK falling strobe high	2		ns
T16	READ DATA set up to BCLK rising	15		ns
T17	READ DATA hold from BCLK	2		ns
T18	WRITE DATA delay from BCLK falling	2	30	ns
T19	WRITE DATA float delay from BCLK falling	2	40	ns
T20	- EX32, - EX16 set up to BCLK falling (host translation)	30		ns
T21	- EX32, - EX16 hold from BCLK falling (host translation)	2		ns
T22	- EX32, - EX16 set up to BCLK rising (host translation)	15		ns
T23	- EX32, - EX16 hold from BCLK rising (host translation)	20		ns

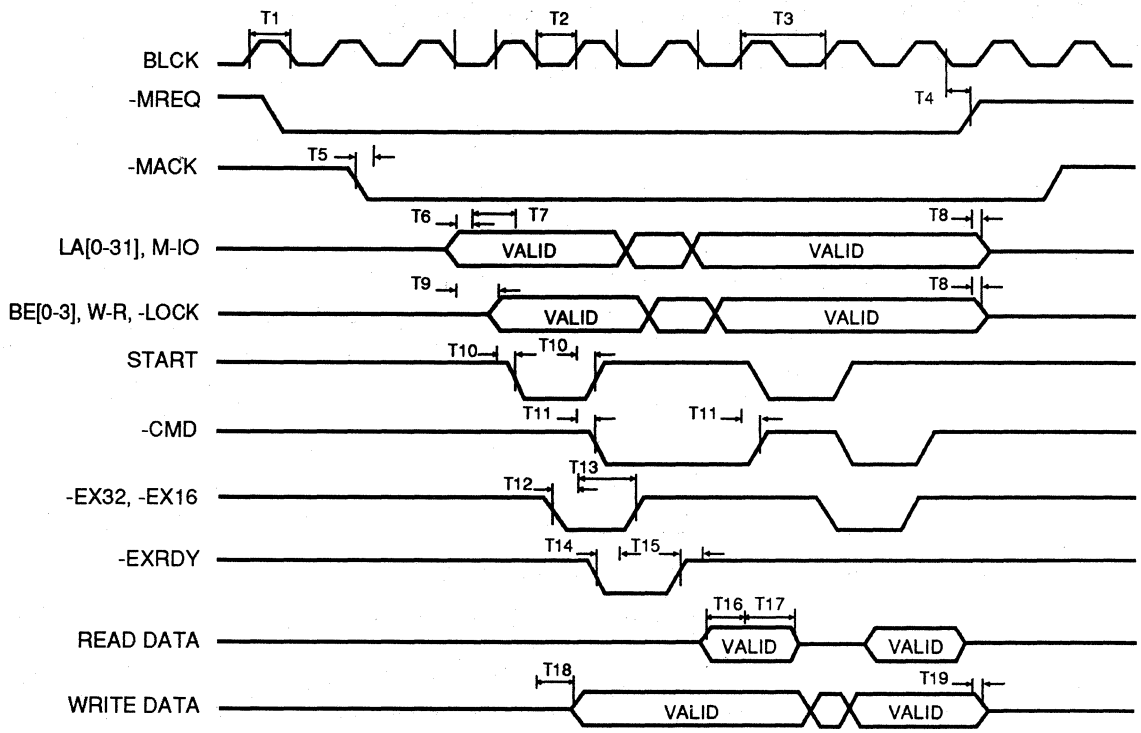
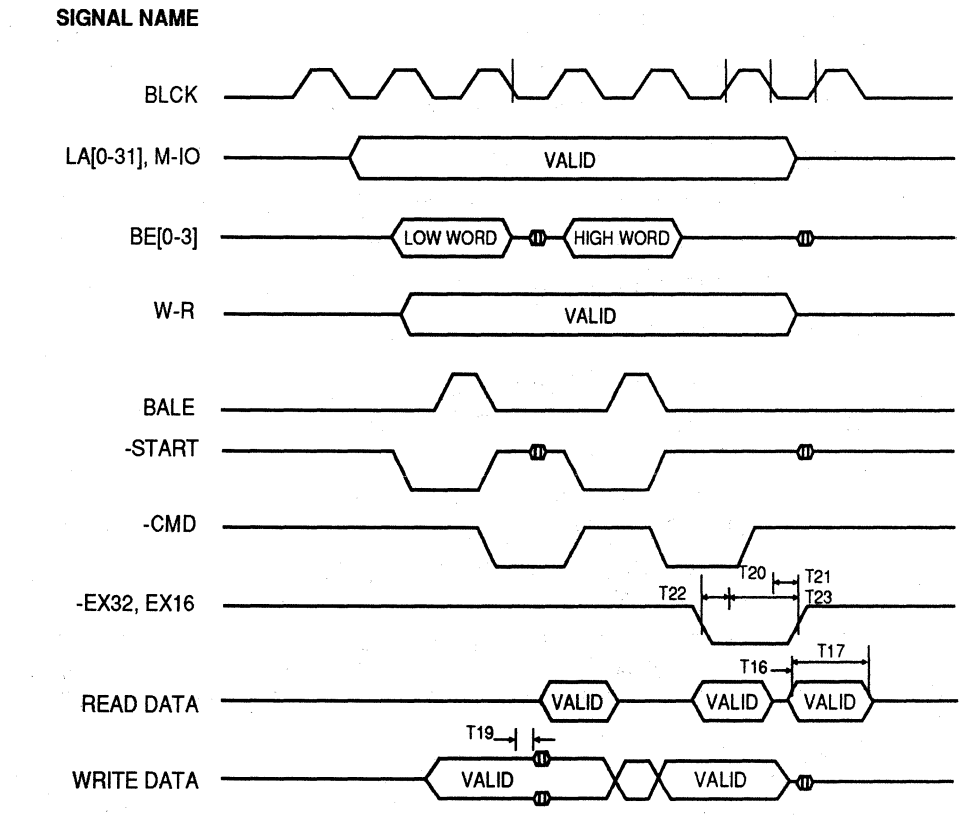


Figure 4-4. EISA Read Cycle Timing Characteristics

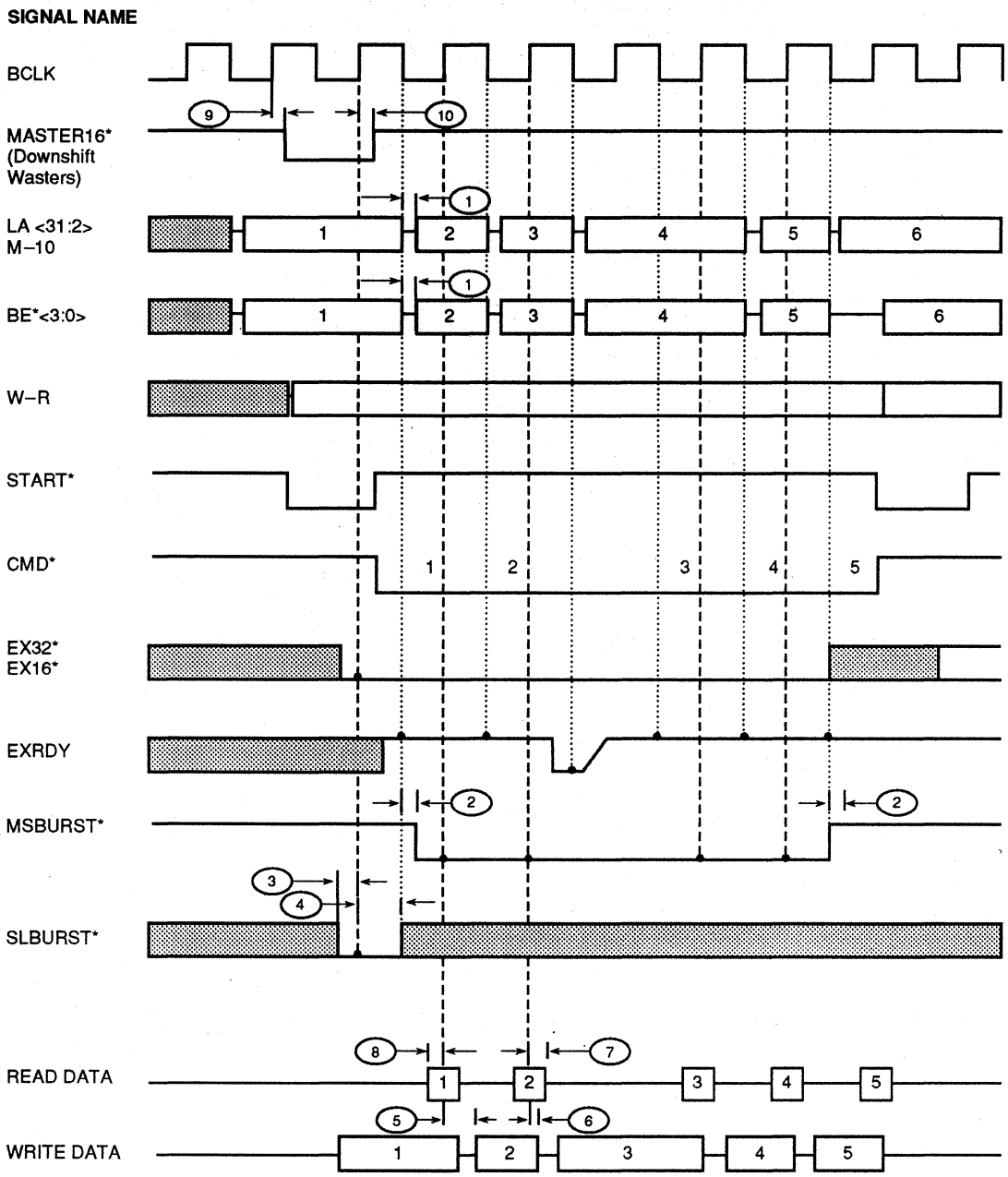


Note:
1. ⓓ indicates where control transfers from master to system or from system to master.

Figure 4-5. EISA Write Cycle Timing Characteristics

Table 4-12. EISA Burst Cycle Parameters

	16- or 32-bit EISA master timing, Burst:		
	Description	**min.**	**max.**
1	LA <31:2>, BE* <3:0> delay from BCLK falling	2.0	44.0
2	MSBURST* delay from BCLK falling	2.0	35.0
3	SLBURST* setup to BCLK rising	15.0	
4	SLBURST* held from BCLK rising	55.0	
5	Data delay from BCLK rising (write) (16-bit) Data delay from BCLK rising (write) (32-bit or downshift)	5.0 5.0	38.0 40.0
6	Data hold from BCLK rising (write)	5.0	
7	Data held after BCLK rising (read)	5.0	
8	Data setup to BCLK rising (read) (16-bit) Data setup to BCLK rising (read) (32-bit)	13.0 15.0	
9	MASTER16* asserted delay from BCLK rising (downshift)		50.0
10	MASTER16* float delay from BCLK rising (downshift)	3.0	40.0



1) EISA Standard Access (Start of Burst)
2) EISA Burst Access
3) EISA Burst Access with One Wait State
4-5) EISA Burst Access
6) EISA Standard Access

Figure 4-6. EISA 32-bit Burst Cycle Timing Characteristics

PART 5

ISA HOST ADAPTERS

List of Tables

Section 5•2: Electrical Interface

Section 5•3: ISA Bus Master Timing Diagrams

5•1

Product Overview

The BusLogic BT-540CF and BT-545C (BT-54X) host adapters are intelligent PC/AT to SCSI bus master host adapter products based on BusLogic-designed MultiMaster ASIC technology. Designed for DOS/Windows and multitasking applications such as Windows NT, NetWare, OS/2, UNIX™ and XENIX™, the intelligent BT-54X provides a high-performance interconnection between the PC/AT bus and Small Computer System Interface (SCSI) peripheral devices.

BusLogic has embedded driver support in most popular operating systems. No additional drivers are needed when your system runs with the current versions of the following operating systems:

- NetWare 3.12/4.X
- Windows NT
- Interactive UNIX
- SCO UNIX
- IBM OS/2
- UNIXWare
- Solaris (for *x*86)
- Vines
- NeXTStep

Software drivers for NetWare 3.11, OS/2, SCO/UNIX and DOS are also available separately.

A BusLogic-designed bus master controller ASIC, an advanced SCSI controller chip, and a 16-bit microprocessor provide higher speed, lower power consumption, fewer parts and higher reliability. First party 16-bit DMA data transfers are performed at speeds of up to 10 MBytes/sec on the PC/AT bus:

- The BT-545C supports single-ended SCSI drives with asynchronous data rates of up to 7 MBytes/sec and synchronous data rates of up to 10 MBytes/sec. The BT-545C uses integrated active terminators.
- The BT-540CF offers the same high performance as the BT-545C. It has a Centronics external SCSI connector, but without floppy support.

Both internal and external 50-pin connectors are included on the BT-54X for flexibility in attaching SCSI devices to the system. An on-board floppy controller chip on the BT-545C independently communicates with any combination of up to two 3.5" or 5.25" floppy drives. BT-54X adapters can also support the newly emerging 2.88 MB floppies.

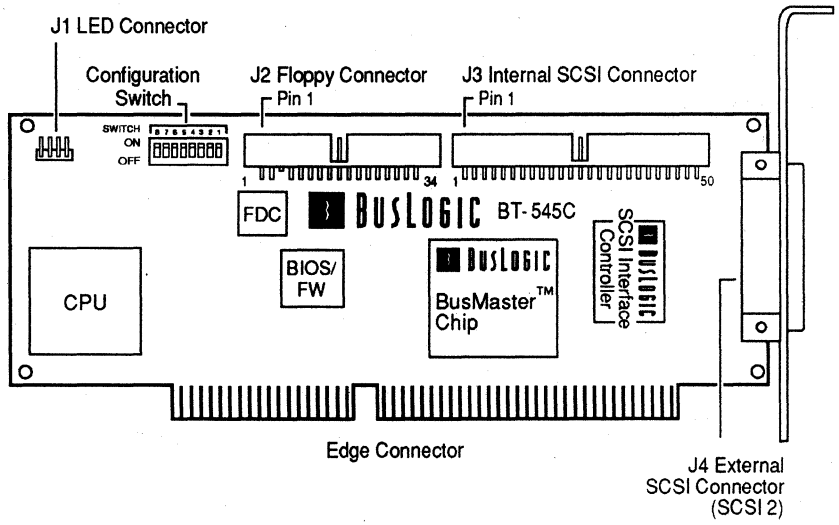


Figure 5-1. The BT-545C Host Adapter

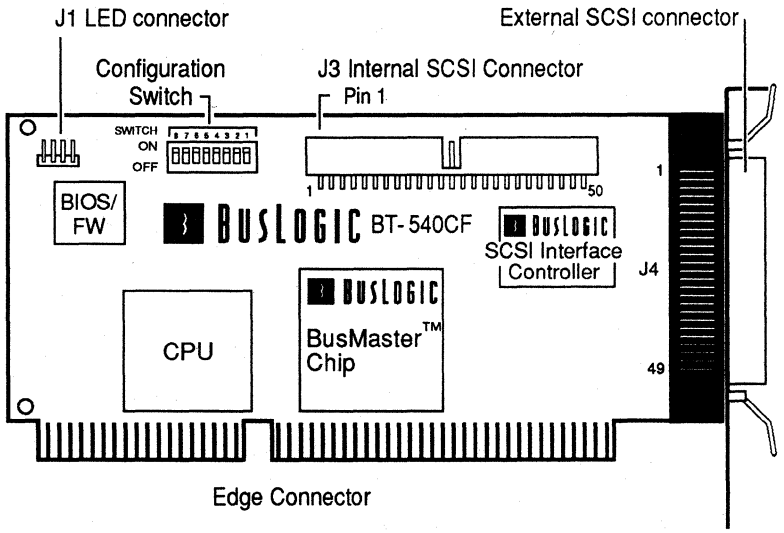


Figure 5-2. The BT-540CF Host Adapter

Specifications

Electrical:	
Operating Voltage	5±0.25V
Operating Current	.5 A Max.
Max. Ripple/Noise	100 mV
Environmental:	
Temperature	0°C to 60°C (32°F to 128°F)
Relative Humidity	10% to 95% non-condensing
Altitude	0 to 10,000 ft. operating
	0 to 15,000 ft. non-operating
Dimensions:	
BT-540CF	6.5" x 3.5"
BT-545C	8.0" x 3.25"
Connectors:	
SCSI Internal	50-pin double-row connector
SCSI External	50-pin shielded SCSI connector
Floppy	34-pin AT-compatible ribbon style (BT-545C only)
System Interface	IBM PC/AT standard 36-pin and 62-pin edge connector
MTBF	90,000 hours

5•2

Electrical Interface

This section provides a complete description of the name, function, and applicable logic level of all signals between the BT-54X and the PC/AT host system.

PC/AT Bus Electrical Interface

The BT-54X is electrically and mechanically compatible with the Input/Output (I/O) bus used in the IBM PC/AT computer. Physically, this I/O bus is contained on two card edge connectors. The DMA control logic on the BT-54X controls the PC/AT bus arbitration and data transfer operations. During DMA data transfers, the BT-54X takes control of the system bus and transfers data directly to and from the main system memory. Both odd and even starting addresses are supported in the DMA logic.

The PC/AT's I/O bus provides the necessary hardware interface to the host Central Processing Unit (CPU) to allow it to communicate with the BT-54X. Figure 5-3 shows the location of the edge connectors on the board.

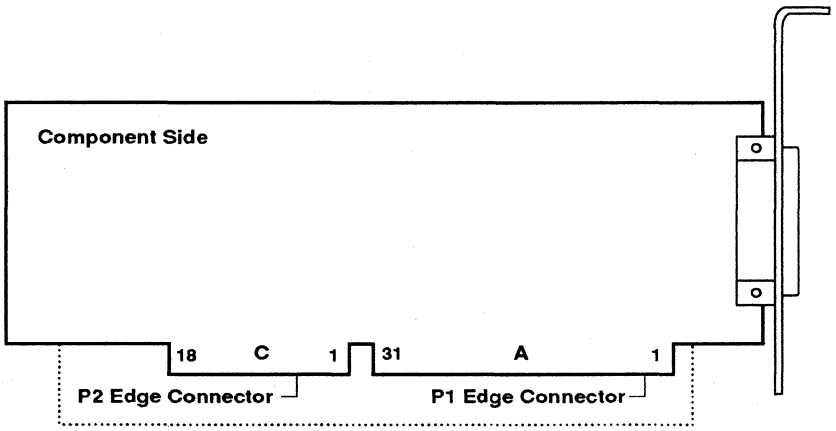


Figure 5-3. BT-54X Edge Connectors

P1 Input/Output Channel Pin Assignment

Table 5-1 summarizes the pin assignments for the I/O channel connectors on the component side. Table 5-2 shows the pin assignments on the solder side of P1, a 62-pin edge connector.

Table 5-1. Component Side Of P1 (Row A), Edge Connector

Signal Pin #	Signal Name	Direction
A1	-I/O CH CK	Not Used
A2	D7	I/O
A3	D6	I/O
A4	D5	I/O
A5	D4	I/O
A6	D3	I/O
A7	D2	I/O
A8	D1	I/O
A9	D0	I/O
A10	I/O CH RDY	Input
A11	-AEN	Input
A12	SA19	I/O
A13	SA18	I/O
A14	SA17	I/O
A15	SA16	I/O
A16	SA15	I/O
A17	SA14	I/O
A18	SA13	I/O
A19	SA12	I/O
A20	SA11	I/O
A21	SA10	I/O
A22	SA9	I/O
A23	SA8	I/O
A24	SA7	I/O
A25	SA6	I/O
A26	SA5	I/O
A27	SA4	I/O
A28	SA3	I/O
A29	SA2	I/O
A30	SA1	I/O
A31	SA0	I/O

Table 5-2. Solder Side Of P1, Edge Connector

Signal Pin #	Signal Name	Direction
B1	GND	Ground
B2	RESET DRV	Input
B3	+5 Vdc	Power
B4	IRQ9	Output
B5	-5 Vdc	Not Used
B6	DRQ2	Output
B7	-12 Vdc	Not Used
B8	-NOWS	Not Used
B9	+12 Vdc	Not Used
B10	GND	Ground
B11	-SMEMW	Not Used
B12	-SMEMR	Not Used
B13	-IOW	I/O
B14	-IOR	I/O
B15	-DACK3	Not Used
B16	DRQ3	Not Used
B17	-DACK1	Not Used
B18	DRQ1	Not Used
B19	-Refresh	Not Used
B20	CLK	Not Used
B21	IRQ7	Not Used
B22	IRQ6	Output
B23	IRQ5	Not Used
B24	IRQ4	Not Used
B25	IRQ3	Not Used
B26	-DACK2	Input
B27	T/C	Input
B28	BALE	Input
B29	+5 Vdc	Power
B30	OSC	Not Used
B31	GND	Ground

P2 Input/Output Channel Pin Assignments

Table 5-3 summarize pin assignments for the I/O channel connectors on the component side. Table 5-4 shows the pin assignments on the solder side of P2, a 36-pin edge connector.

Table 5-3. Component Side Of P2 (Row C), Edge Connector

Signal Pin #	Signal Name	Direction
C1	-SBHE	I/O
C2	LA23	Output
C3	LA22	Output
C4	LA21	Output
C5	LA20	Output
C6	LA19	Output
C7	LA18	Output
C8	LA17	Output
C9	-MEMR	Output
C10	-MEMW	Output
C11	DO8	I/O
C12	DO9	I/O
C13	D10	I/O
C14	D11	I/O
C15	D12	I/O
C16	D13	I/O
C17	D14	I/O
C18	D15	I/O

Table 5-4. Solder Side Of P2, Edge Connector

Signal Pin #	Signal Name	Direction
D1	-MEM CS16	Input
D2	-I/O CS 16	Input
D3	IRQ10	Output
D4	IRQ11	Output
D5	IRQ12	Output
D6	IRQ15	Output
D7	IRQ14	Output
D8	-DACK0	Not Used
D9	DRQ0	Not Used
D10	-DACK5	Input
D11	DRQ5	Output
D12	-DACK6	Input
D13	DRQ6	Output

Table 5-4. Solder Side Of P2, Edge Connector (Continued)

Signal Pin #	Signal Name	Direction
D14	-DACK7	Input
D15	DRQ7	Output
D16	+5 Vdc	Power
D17	-MASTER	Output
D18	GND	GND

P1 and P2 Input/Output Channel Signal Descriptions

The following is a description of the PC/AT bus signals. All signal lines are TTL-compatible. I/O adapters should be designed with a maximum of two low-power Shottky (LS) loads per line. Signals preceded by a hyphen (-) indicate that the signal is in an active low signal.

Table 5-5. P1 & P2 I/O Channel Signal Descriptions

Signal	Definition
SA0—SA19	**Address Bits 0–19:** These positive true 20-bit address bits are used to select memory or I/O devices within the system.
AEN	**Address Enable**: This signal is used to disconnect the host CPU and other devices from the bus to allow DMA transfers to take place. When this signal is asserted, the DMA controller has control of the address bus, the data bus, the Read command lines, and the Write command lines.
BALE	**Buffered Address Latch Enable**: A19-A0 are latched on the falling edge of the Buffered Address Latch Enable (BALE) signal to indicate a valid address by the bus master. During DMA cycles the bus master forces the Buffered Address Latch Enable (BALE) high.
–MASTER	**Bus Master:** This signal is used with a DMA Request (DRQ) signal to gain control of the host system bus. The BT-54X issues a DRQ signal to the DMA channel in cascade mode and receives a DMA Acknowledge (-DACK) signal. Upon receiving the -DACK signal, the BT-54X controller asserts the Bus Master (-MASTER) signal, which allows it to control the system address, data, and control lines. The BT-54X is allowed to control the system bus for no more than 15 µs so that the host can refresh the system memory at regular intervals.
–DACK2 & –DACK 5 to 7	**DMA Acknowledge:** These active low signals are used to acknowledge their respective DMA requests.
DRQ 2 & DRQ 5 to 7	**DMA Request**: The DMA Requests 2 and 5 to 7 signals are used to request a host transfer. They are active high. DRQ signals are held high until the corresponding DMA Acknowledge (DACK) signal is asserted.
D0—D15	**Data Bus Bits 0-15:** These positive true signals comprise the 16-bit, tri-state, bi-directional, data bus for commands and status communication between the BT-54X and the PC/AT host. D7-D0 form the least significant byte of a 16-bit data transfer.
I/O CH RDY	**I/O Channel Ready:** This signal is pulled low by a memory or an I/O device to lengthen I/O or memory cycles.
–I/O CH CK	**I/O Channel Check:** This signal provides the active bus with parity (error) information about memory or devices on the I/O channel. When this signal is active, it indicates an uncorrectable system error. This signal is generated by the active slave device.

Table 5-5. P1 & P2 I/O Channel Signal Descriptions (Continued)

Signal	Definition
–I/O CS 16	This signal indicates to the system that the present data transfer is a 16-bit I/O cycle.
–IOR	**I/O Read**: Input/Output Read instructs an I/O device to place its data onto the data bus. This signal may be generated by any bus master in the system.
–IOW	**I/O Write:** Input/Output Write instructs an I/O device to store data from the data bus. This signal may be generated by any bus master in the system.
IRQ6	**Interrupt Request 6**: This signal is used to tell the host CPU that the floppy controller portion of the host adapter needs attention. A request is generated when this signal is raised from low to high. The signal is held high until the host CPU acknowledges the interrupt request.
IRQ9—IRQ12 & IRQ14—IRQ15	**Interrupt Requests 9–12, 14 and 15**: These signals tell the host CPU that an I/O device needs attention. An interrupt request is generated when an IRQ signal is raised from low to high. The signal is held high until the host CPU acknowledges the interrupt request (Interrupt Service routine).
LA23—LA17	These unlatched signals can be used to address memory within the system, extending the address space to 16Mb. These signals are valid when the Buffered Address Latch Enable (BALE) signal is asserted and are latched by the falling edge of the Buffered Address Latch Enable (BALE) signal.
–MEM CS16	**Memory Chip Select:** This signal indicates that the present data transfer is a 16-bit memory cycle derived from the slave device's decoding of LA17–LA23.
–MEMR	**Memory Read**: The memory read signal instructs the memory devices to drive data onto the data bus. This signal may be driven by any bus master in the system.
–MEMW	**Memory Write:** The memory write signal instructs the memory devices to store data from the data bus. This signal may be driven by any bus master in the system.
RESET DRV	**Reset Drive**: This active high signal is used to reset or to initialize system logic at power-up time or during a low line-voltage outage.
–SBHE	**System Bus High Enable:** This signal enables a data transfer on the upper byte (D15-D8) of the data bus. The present bus master uses this signal to condition the data bus buffers driving D15-D8 lines.
T/C	**Terminal Count:** This signal provides a pulse when the terminal count for any DMA channel is reached.

5•3

ISA Bus Master Timing Diagrams

Table 5-6. ISA Bus Timing Specifications

Name	Parameter	Min.	Typ.	Max.	Unit
T1	(drq [5-7]) high to (-dack [5-7]) low	0			ns
T2	Bus tri-state to (drq [5-7]) low	2CP-10			ns
T3	(-dack [5-7]) low to bus drivers on	0		4CP+15	ns
T4	(-dack [5-7]) low to (-master) low	0		4CP+15	ns
T5	(access size) returned from (A0-A19) valid	0		3CP	ns
T6	(A0-A19) hold from (access strobe) high	1CP			ns
T7	(access strobe) to (access strobe)	8CP-10			ns
T8	(A0-A19) valid to (access strobe)	4CP-10			ns
T9	(access strobe) pulse width	4CP-10			ns
T10	(access strobe) high to next (access strobe) low	4CP-10			ns
T11	(access strobe) low from (-dack [5-7]) low	14CP			ns
T12	(write data) setup to (access strobe) low	0			ns
T13	(write data) hold from (access strobe) high	1CP			ns
T14	(read data) valid to (access strobe) high	25			ns
T15	(read data) hold from (access strobe) high	0			ns
T16	(IO ch rdy) low from unextended and of (access strobe)			2CP	ns
T17	(IO ch rdy) high to (access strobe) high			5CP	ns

Note:

1. CP is 25 ns.
2. Default pulse width of (access strobe) is programmable in register 18.
3. The addressed slave must deassert (IO ch rdy) within parameter T16 time of the end of the normal transfer cycle.
4. (access strobe) is one of the following pins: -MEMR, -MEMW, -IOR, or -IOW.
5. (access size) is one of the following pins: -I/O C516 or -MEMC516.

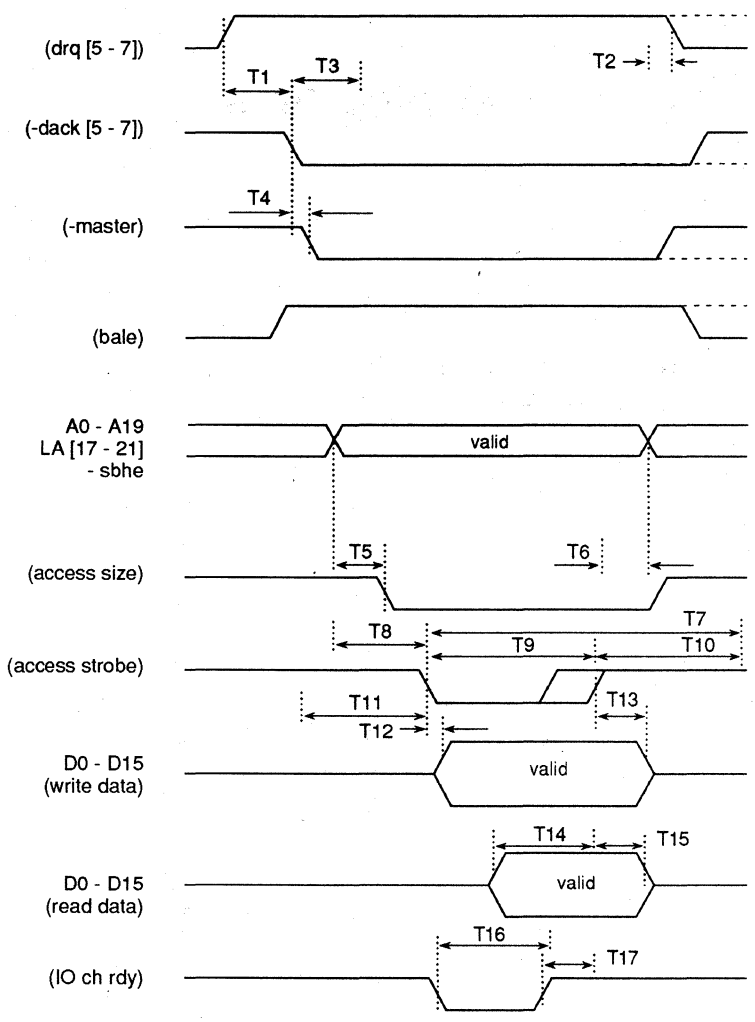


Figure 5-4. ISA Bus Master Timing Characteristics

Index

Numerics

A

B

C

D

E

F

H

I

L

M

O

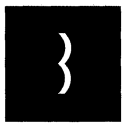
P

R

S

T

V



BUSLOGIC

BusLogic Inc.
4151 Burton Drive
Santa Clara, CA 95054

P/N 3002593 REV. A

Printed on Recycled Paper