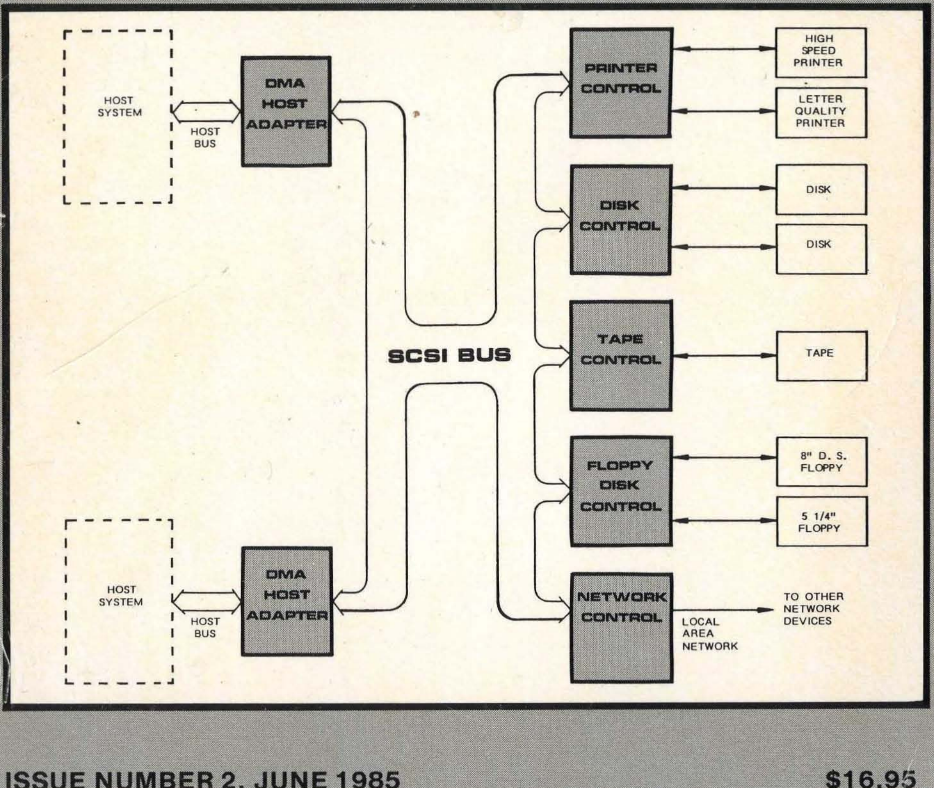
ADSI ADAPTIVE DATA SYSTEMS INC.

SCSI GUIDEBOOK



ISSUE NUMBER 2, JUNE 1985 $16.95

SCSI GUIDEBOOK

ISSUE NUMBER 2, JUNE 1985

PRINTED USA

TABLE OF CONTENTS

FORWARD vii

1.0 INTRODUCTION 1

1.1 ISSUE NUMBER 2 OF THE SCSI GUIDEBOOK 1

1.2 HISTORY OF SCSI 2

2.0 BASIC SCSI CONCEPTS 4

2.1 THE SCSI BUS VS. THE SYSTEM BUS 4
2.1.1 SIMILARITIES 4
2.1.2 DIFFERENCES 6

2.2 LEVELS OF SCSI USAGE 8
2.2.1 SINGLE INITIATOR / SINGLE TARGET 8
2.2.2 SINGLE INITIATOR / MULTIPLE TARGET 8
2.2.3 MULTIPLE INITIATOR / MULTIPLE TARGET 9

2.3 SCSI SPECIFICATION SUMMARY 10
2.3.1 BUS SIGNALS 11
2.3.2 BUS PROTOCOL 15
2.3.3 COMMAND / STATUS SET and MESSAGE SYSTEM 16

2.4 HOST MEMORY / HOST BUS ADAPTER / SCSI FORMATTER RELATION 18
2.4.1 SCSI I/O SYSTEM READ DATA EXAMPLE 20
2.4.2 I/O CHANNEL CONCEPT 22
2.4.3 I/O CHANNEL OPERATION EXAMPLES 24
2.4.3.1 EXAMPLE #1 24
2.4.3.2 EXAMPLE #2 50

3.0 SCSI PERIPHERALS 64
3.1 HARDWARE FEATURES 65
3.1.1 ARBITRATION & RESELECTION CAPABILITY 65
3.1.2 DATA BUS PARITY 65
3.1.3 SINGLE / DIFFERENTIAL BUS DRIVE 65
3.1.4 DATA TRANSFER RATE 67
3.1.5 FORMATTER BUFFER CAPACITY 69

3.2 FIRMWARE FEATURES 70
3.2.1 SCSI MESSAGE SYSTEM 71
3.2.2 SCSI COMMAND DECODE & EXECUTION 78
3.2.3 SCSI BUS ERROR RECOVERY 78
3.2.4 MULTITASKING 79

4.0 SCSI PERIPHERAL FORMATTERS 80

4.1 HARD AND FLEXIBLE DISK FORMATTERS 81
4.1.1 COMMAND SET 82
4.1.2 PERFORMANCE CONSIDERATIONS 91
4.1.3 FORMATTER / DISK INTERFACE 94

4.2 TAPE FORMATTERS 97
4.2.1 STREAMING VS. START/STOP 98
4.2.2 COMMAND SET 98
4.2.3 BUFFERS 102

4.3 PRINTER FORMATTERS 104

4.4 PROCESSOR DEVICES AND HOST ADAPTERS 107
4.4.1 HOST ADAPTER FUNCTIONS AND FEATURES 107
4.4.2 HOST ADAPTER IMPLEMENTATIONS 108
4.4.3 COMMAND SET 112

4.5 WRITE-ONCE READ-MULTIPLE AND READ-ONLY FORMATTERS 114
4.5.1 WRITE-ONCE READ-MULTIPLE DEVICES 114
4.5.2 READ-ONLY DIRECT ACCESS DEVICES 117

APPENDICES

A: COMMAND DESCRIPTOR BLOCK DEFINITION
A.1 GROUP 0 COMMON COMMANDS AND DIRECT-ACCESS DEVICE COMMANDS A-1
A.1.5 REASSIGN BLOCKS Command A-9
A.1.6 REAQD Command.......... A-11
A.1.7 WRITE Command.......... A-12
A.1.8 SEEK Command.......... A-13
A.1.9 INQUIRY Command A-14
A.1.10 MODE SELECT Command A-16
A.1.11 RESERVE Command A-18
A.1.12 RELEASE Command A-21
A.1.13 COPY Command A-22
A.1.14 MODE SENSE Command A-30
A.1.15 START/STOP UNIT Command A-32
A.1.16 RECEIVE DIAGNOSTIC Command A-33
A.1.17 SEND DIAGNOSTIC Command A-34
A.1.18 PREVENT/ALLOW MEDIUM REMOVAL Command A-35
A.2 GROUP 1 COMMON COMMANDS AND DIRECT-ACCESS DEVICE COMMANDS A-25
A.3 GROUP 0 COMMANDS FOR SEQUENTIAL ACCESS DEVICES .. A-35
A.4 GROUP 0 COMMANDS FOR PRINTER DEVICES A-73
A.4.1 FORMAT Command A-73
A.4.2 PRINT Command.......... A-74
A.4.3 SLEW AND PRINT Command A-75
A.4.4 FLUSH BUFFER Command A-76
A.4.5 RECOVER BUFFERED DATA Command A-77
A.4.6 MODE SELECT Command A-78
A.4.7 RESERVE UNIT and RELEASE UNIT Commands A-79
A.4.8 MODE SENSE Command A-81
A.4.9 STOP PRINT Command A-83
A.5 GROUP 0 COMMANDS FOR PROCESSOR DEVICES A-84
A.6 GROUP 0 COMMANDS FOR WRITE-ONCE READ-MULTIPLE DEVICES A-86
A.6.1 READ Command A-86
A.6.2 WRITE Command A-87
A.6.3 MODE SELECT Command A-89
A.6.4 MODE SENSE Command A-91
A.7 GROUP 1 COMMANDS FOR WRITE-ONCE READ-MULTIPLE DEVICES A-92
A.8 GROUP 0 COMMANDS FOR READ-ONLY DIRECT-ACCESS DEVICES A-97
A.9 GROUP 1 COMMANDS FOR READ-ONLY DIRECT-ACCESS DEVICES A-97
B: RETURN STATUS BLOCK AND SENSE DATA BLOCK DEFINITION
B.1 STATUS BYTE B-1
B.2 SENSE DATA B-3
C: GLOSSARY OF TERMS

LIST OF FIGURES

1-1: Small Computer System Interface Architecture 3

2-1: I/O Bus Connection 5
2-2: Interface Connection 6
2-3: Single Initiator / Multiple Target Configuration ... 9
2-4: Multiple Initiator / Multiple Target Configuration . 10
2-5: Bus Phase Diagram 12
2-6: SCSI Bus Signal Definition 13
2-7: SCSI Signal Sources 14
2-8: Information Transfer Phases 15
2-9: SCSI Device Types 16
2-10: Host Memory / Host Adapter / SCSI Formatter 19
2-11: Logical Thread 21
2-12: I/O System Configuration 25

Example #1 Figures
2-13: Example #1 Time Line Diagram 26
2-14: I/O Channel Performance Evaluation 27
2-15: System Snapshot t = 1.0ms 29
2-16: System Snapshot t = 1.5ms 30
2-17: System Snapshot t = 2.0ms 31
2-18: System Snapshot t = 3.0ms 33
2-19: System Snapshot t = 3.5ms 35
2-20: System Snapshot t = 35.0ms 36
2-21: System Snapshot t = 41.5ms 38
2-22: System Snapshot t = 43.0ms 40
2-23: System Snapshot t = 46.0ms 43
2-24: System Snapshot t = 47.5ms 45
2-25: System Snapshot t = 65.5ms 47
2-26: System Snapshot t = 346.0ms 49

Example #2 Figures
2-27: I/O Channel Performance Evaluation 50
2-28: Example #2 Time Line Diagram 51
2-29: System Snapshot t = 0.0ms 52
2-30: System Snapshot t = 2.5ms 56
2-31: System Snapshot t = 4.0ms 58
2-32: System Snapshot t = 75.5ms 59
2-33: System Snapshot t = 80.5ms 61

LIST OF FIGURES cont'd

3-1: SCSI Peripheral Functional Blocks 64
3-2: SCSI Bus Signal Termination 66
3-3: Pin Assignments for Differential Option 67
3-4: REQUEST / ACKNOWLEDGE Timing for Asynchronous Data Transfer 68
3-5: REQUEST / ACKNOWLEDGE Timing for Synchronous Data Transfer 69
3-6: SCSI Interface Layers 71
3-7: SCSI Message Definition 72
3-8: Time Line Diagram For Message System Example 77

4-1: SCSI Peripheral Formatter 80
4-2: Disk Formatter Block Diagram 82
4-3: Group 0 Command Codes for Direct Access Devices 84
4-4: Group 1 Command Codes for Direct Access Devices 85
4-5: SCSI Logical Block Mapping 86
4-6: SCSI Device Type I.D. Codes 87
4-7: "In-Line" Sparing Format Technique 92
4-8: Group 0 Command Codes for Sequential Access Devices. 100
4-9: Throughput vs. Block Size for Streaming Tape Drives. 103
4-10: Group 0 Command Codes for Printer Devices 105
4-11: Six Chip SCSI Host Adapter Block Diagram 109
4-12: Programmed I/O Host Adapter with T/R Machine 110
4-13: I/O Channel Block Diagram 111
4-14: Group 0 Command Codes for Processor Devices 113
4-15: Group 0 Command Codes for Write-Once Read-Multiple Devices 115
4-16: Group 1 Command Codes for Write-Once Read-Multiple Devices 116
4-17: Group 0 Command Codes for Read-Only Direct Access Devices 118
4-18: Group 1 Command Codes for Read-Only Direct Access Devices 119

A-1: FORMAT UNIT Command Variations A-4
A-2: Defect List - Block Format A-5
A-3: Defect List - Bytes From Index Format A-6
A-4: Defect List - Physical Sector Format A-7
A-5: REASSIGN Blocks Defect List A-9
A-6: Selected CHECK CONDITIONS for READ Command A-11
A-7: Selected CHECK CONDITIONS for WRITE Command A-12
A-8: INQUIRY Data A-14
A-9: Peripheral Device Type List A-14
A-10: Version A-15
A-11: MODE SELECT Parameter List A-16
A-12: Data Format of Extent Descriptors A-18

LIST OF FIGURES cont'd

A-13: COPY Parameter List A-22
A-14: COPY Functions A-23
A-15: Segment Descriptor for COPY Function Codes 00H and 01H A-25
A-16: Segment Descriptor for COPY Function CODE 02H A-27
A-17: Segment Descriptor for COPY Function CODE 03H A-28
A-18: MODE SENSE Data A-30
A-19: READ CAPACITY Data A-37
A-20: SEARCH DATA Parameter List A-45
A-21: READ BLOCK LIMITS Data A-50
A-22: MODE SELECT Parameter List A-62
A-23: MODE SENSE Data A-67
A-24: MODE SELECT Parameter List A-76
A-25: MODE SENSE Data A-88
A-26: MODE SELECT Parameter List A-90
A-27: MODE SENSE Data A-90

B-1: Status Byte B-1
B-2: Status Byte Code Bit Values B-1
B-3: Nonextended Sense Data Format B-3
B-4: Extended Sense Data Format B-4
B-5: Sense Key (0H-7H) Descriptions B-5

F O R W A R D

This publication, prepared by Adaptive Data Systems, Inc. (**ADSI**), is the second issue of the **SCSI GUIDEBOOK**. The purpose of the document is twofold. First, it provides a status report on the industry effort to standardize the Small Computer System Interface (SCSI) via the American National Standards Institute (ANSI). Second, it provides both users and suppliers of SCSI devices a reference point to ensure compatibility among products during the early stages of development and integration. As appropriate, the **SCSI GUIDEBOOK** will be reissued as the state-of-the-art in SCSI continues to progress.

The handbook is written in "top-down" fashion and provides detail at all levels for a wide range of readership. Individuals who wish only a high level overview of SCSI should read sections 1.0 through 2.4.2. Section 2.4.3 provides two detailed examples of SCSI performance which illustrate the operation of SCSI devices in a high performance I/O system.

Sections 3.0 and 4.0 deal with specific SCSI peripherals and performance features. These two sections can be utilized as a starting reference for comparison and evaluation of SCSI devices. As with other portions of the handbook, readers can use Sections 3.0 and 4.0 to the level of detail they require.

Appendices A, B, and C provide the most detail, with text taken directly from ANSI documents.

1.0 INTRODUCTION

The revolution in computer systems I/O continues at an accelerating pace. Performance standards for peripheral devices get rewritten regularly, as newer units press the boundaries of capacity, data rate, and access time. Further, the definition of I/O itself gets hazier as communication networks infiltrate computer systems and alternative computer architectures distribute the fundamental processing tasks over multiple processors, via I/O channels. This book documents a major effort by the computer industry to structure the I/O channel to better harness the power of new I/O devices. That effort is an emerging ANSI standard - the Small Computer System Interface (SCSI).

1.1 ISSUE NUMBER 2 OF THE SCSI GUIDEBOOK

Issue number 1 of this book, released in November, 1982, documented Revision 4 of the SCSI specification proposed as a standard by the X3T9.2 committee of the American National Standards Institute (ANSI). At that time, SCSI was known primarily to a select group of experts in computer systems I/O and to controller and peripheral companies interested in promulgating it as a standard. Now, three years later, the ANSI committee is reviewing revision 15 of the SCSI specification, which is nearing the end of the required public review period and which will shortly be adopted by ANSI as a standard. SCSI, including its progenitor SASI, has become one of the most widely-used interconnections for rigid disks.

Recently, the questions asked regarding SCSI have started to change. System architects, with a good understanding of what SCSI is, now want to know actual performance benchmarks in systems using SCSI. At the other end of the spectrum, many potential users, less technically sophisticated, need a better non-technical explanation of how SCSI works and why it should be used.

This second issue has been prepared in response to the new specification and the new issues facing SCSI. In Chapter 2, the introductory explanation of SCSI has been expanded. Chapters 3 and 4 have been updated to reflect the modifications in the SCSI specification from revision 4 to revision 15. The appendices, which include extensive extracts from the actual SCSI specification, have undergone extensive modification to incorporate the changes in the SCSI specification. This issue is consistent with revision 15 of the SCSI specification, which was released by the ANSI committee X3T9.2 on May 8, 1985.

1.2 HISTORY OF SCSI

In the late 1970's, Shugart began a program to develop an intelligent interface for its new generation of disks. The architecture chosen for the task was the IBM I/O channel, which IBM developed and began to utilize in the 1960's and 70's. IBM I/O allowed numerous intelligent peripherals to communicate with multiple hosts over a single bus.

The name given the new interface was "Shugart Associates Standard Interface." SASI, as the interface was known, received good market acceptance due to both its technical merit and the marketing resources of Shugart Associates.

In November of 1981, Shugart Associates and NCR convinced ANSI to adopt SASI as the working document for an Intelligent Peripheral Interface Standard. In February of 1982, an ANSI subcommittee (X3T9.2) began work on the Shugart Associates Standard Interface. The committee named the proposed I/O specification the Small Computer System Interface (SCSI). Figure 1-1 shows a typical SCSI I/O architecture.

The first generation of SASI products was actually a subset of the original SASI specification. Arbitration, a performance feature required for direct peripheral-to-peripheral communications, was not included in the initial product offering.

As this issue goes to press, it is a pleasure to report that SCSI has made steady progress on many fronts. First, the ANSI committee X3T9.2 has recommended revision 15 of the SCSI specification to committee X3 on I/O standards. The specification is close to completing the required period for public review. Soon, revision 15 will quite probably be recommended for acceptance as an ANSI standard.

Second, SCSI (and its progenitor SASI) by now represents one of the largest installed bases of peripheral interconnections in the computer industry. SASI received wide market acceptance several years ago as a simple interconnection approach for the then-new 5-1/4" Winchester disk drives.

Third, SCSI is now available on many controllers, peripherals, and subsystems. Controller suppliers, seeing the advantage of a common interface for controllers, have made SCSI controllers available for a wide spectrum of peripheral devices. Gradually, SCSI host adapters have been introduced for most systems busses. A few peripheral vendors have even integrated SCSI directly into the device electronics.

Fourth, and perhaps most important, SCSI has gained acceptance by many major computer systems vendors in the past few years. It is being designed into new generations of computers for a wide spectrum of applications. Some users are even tackling the thorny problems involved in retrofitting SCSI into existing systems. With this groundswell of acceptance, SCSI will be a major I/O bus for the remainder of the decade and into the 1990's.

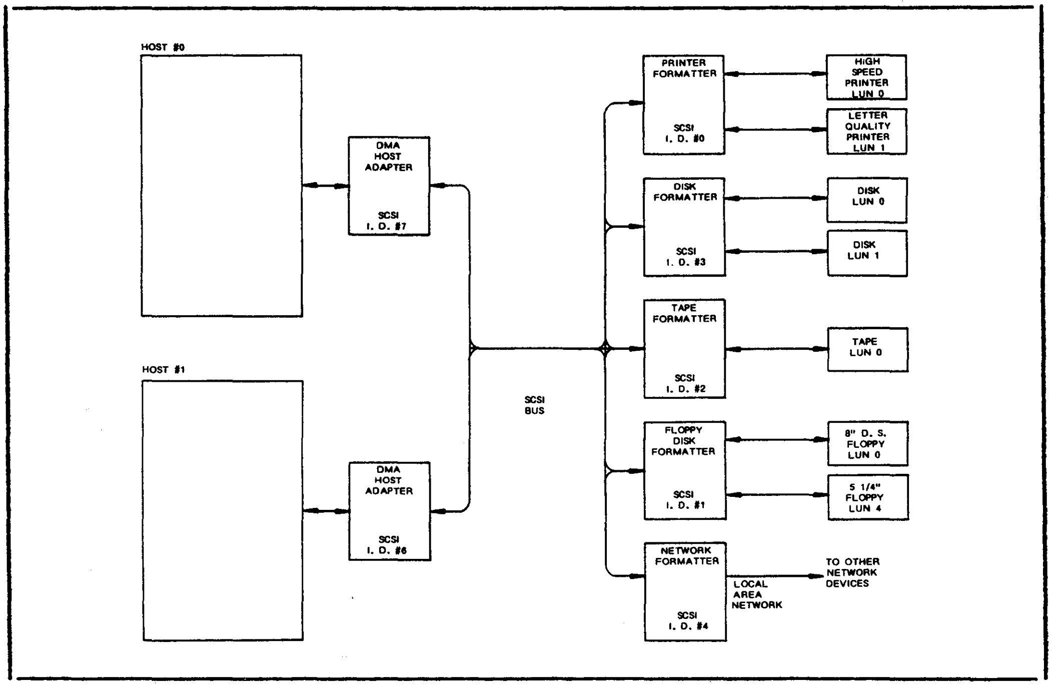

HOST #0
HOST #1
DMA HOST ADAPTER SCSI I. D. #7
DMA HOST ADAPTER SCSI I. D. #6
SCSI BUS
PRINTER FORMATTER SCSI I. D. #0
HIGH SPEED PRINTER LUN 0
LETTER QUALITY PRINTER LUN 1
DISK FORMATTER SCSI I. D. #3
DISK LUN 0
DISK LUN 1
TAPE FORMATTER SCSI I. D. #2
TAPE LUN 0
FLOPPY DISK FORMATTER SCSI I. D. #1
8" D. S. FLOPPY LUN 0
5 1/4" FLOPPY LUN 4
NETWORK FORMATTER SCSI I. D. #4
LOCAL AREA NETWORK
TO OTHER NETWORK DEVICES

Figure 1-1: Small Computer System Interface Architecture

2.0 BASIC SCSI CONCEPTS

The principles behind SCSI are simple yet powerful. To understand them does not require a grasp of new, complex concepts. Rather, SCSI is an innovative application of existing computer system principles, and the reader needs only a change in perspective toward the I/O subsystem to understand SCSI and appreciate its benefits.

2.1 THE SCSI BUS VS. THE SYSTEM BUS

SCSI is best understood by comparing it to a system, or "backplane" bus. Both the similarities and differences between the two are revealing.

2.1.1 SIMILARITIES BETWEEN THE SCSI BUS AND THE SYSTEM BUS

SCSI has many similarities to typical system busses, such as: 1) multiple slots, 2) multiple master capability, 3) operational phases, and 4) signal organization.

The concept of multiple slots, while common for system busses, is an innovation for microcomputer I/O connections and most minicomputer I/O connections also. For this discussion, the term "bus" refers to any interconnection which has multiple slots, while the term "interface" refers only to interconnections which allow only two devices - usually a master and a slave. Using this terminology, most disk and tape connection standards are interfaces. The ST506 connection allows only one disk and one controller on each radial connection. The controller may support four disk drives, but it uses four separate interfaces to accomplish the task. Similarly, the Pertec, QIC-02, and QIC-36 tape drive connections are interfaces.

The multiple slot feature of the SCSI bus is unusual for an I/O interconnection, and it is critical for the performance advantages of SCSI. Figure 2-1 shows a typical connection diagram for an I/O bus architecture. SCSI will also work satisfactorily with only two devices on the bus, and in this configuration (shown in Figure 2-2), it looks like an interface. In fact, this may be the most common configuration in use, since it is used with the older, low-performance SASI controllers and host adapters. However, it is important to remember that SCSI is designed to work most effectively in the bus configuration shown in Figure 2-1.

The second similarity between the SCSI bus and some system busses is multiple master capability. Some system busses establish priority by hard-wired techniques such as daisy-chained priority lines, while others use a voting, or arbitration, cycle on the bus. Fortunately for SCSI user convenience, SCSI establishes bus master priority exclusively in the arbitration cycle, with no physical modification of the cable or devices required. The only hardware activity required is for the SCSI address I.D. to be set once, via hardware jumpers, when the device is installed in the system.

Like many backplane busses, the SCSI specification defines several operational cycles, or bus phases. Of course, SCSI has the normal phases required for information transfer, but in addition it includes several phases to accomplish the arbitration necessary for multi-master operation.

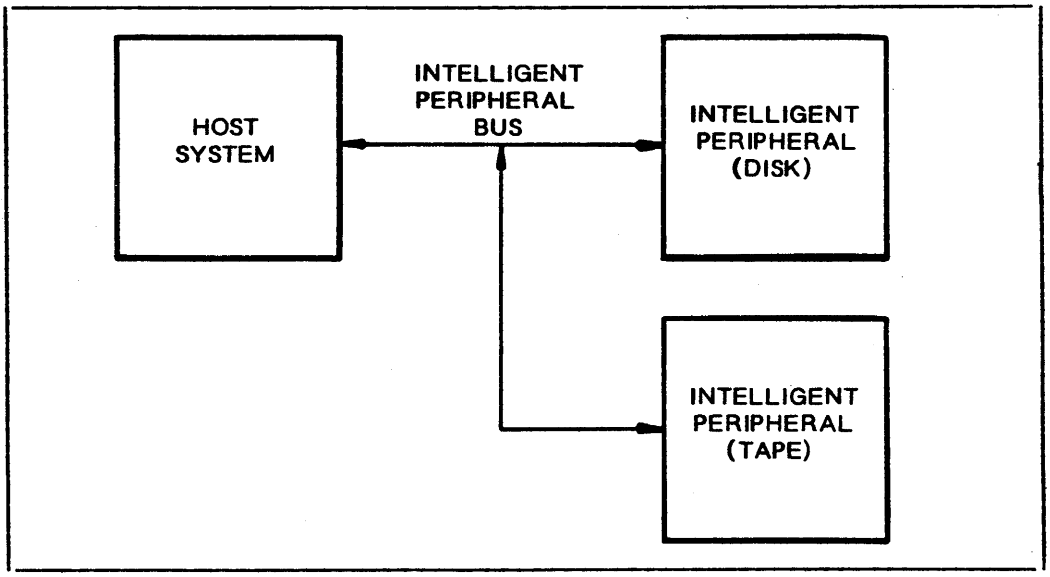

HOST SYSTEM
INTELLIGENT PERIPHERAL BUS
INTELLIGENT PERIPHERAL (DISK)
INTELLIGENT PERIPHERAL (TAPE)

Figure 2-1: I/O Bus Connection

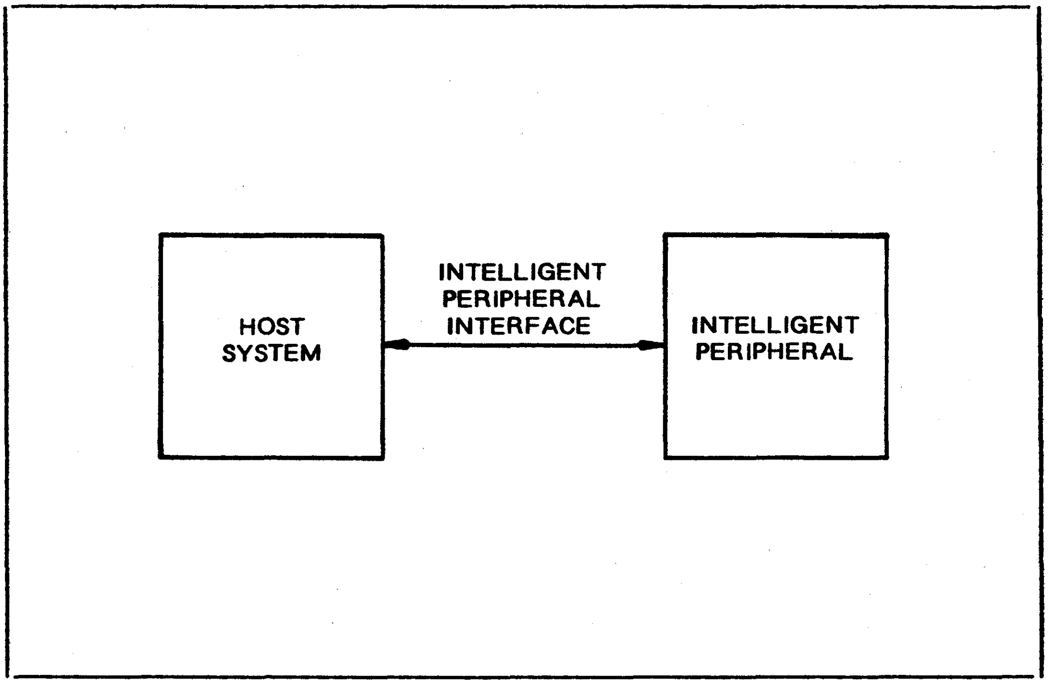


Figure 2-2: I/O Interface Connection

The SCSI signal set closely resembles typical signal sets for a system bus. It contains a data bus (eight bits plus parity) and various control/status signals. These signals are located on the same pins at all points on the bus, with no daisy-chaining or physical repositioning required for any purpose.

2.1.2 DIFFERENCES BETWEEN THE SCSI BUS AND THE SYSTEM BUS

While SCSI and system busses are both computer busses in the broad sense, the purpose of each is different, which gives rise to some broad, operational differences. These differences become obvious and reasonable, given a good understanding of the purposes of each.

Backplane busses are designed primarily for CPU-memory transfers, or instruction fetches. For high performance, the instruction fetch cycle must be rapid, which implies a large address space and short (single word) data transfers. In contrast, the SCSI bus is designed to facilitate large data transfers between peripheral devices and hosts. To enhance its function, SCSI differs from system busses as follows:

- Implementation (cable vs. backplane)
- Command Set Definition
- Message and Data Transfer Sequences
- Data Rate Restrictions

System busses are typically implemented on a printed circuit motherboard. The motherboard and card cage provide a structure for the various computer cards, which have no other natural mounting location. Peripheral formatters, on the other hand, have well-defined mounting locations directly on top of the associated peripheral devices. Therefore, the SCSI bus is implemented in a 50-pin ribbon cable (up to 6 meters long) which is very easy to daisy-chain between the devices.

SCSI includes a comprehensive command set definition, which most system busses do not. The command set is integral to the entire concept of the I/O bus. By fixing the commands, SCSI offers standardization and interchangeability across peripheral devices of the same type, and even basic commonality across different device types.

With the standard command set, SCSI provides a full set of messages with which SCSI devices supervise execution of the commands. The messages provide mechanisms for error handling and maximizing utilization of the SCSI bus.

For high performance I/O across the SCSI bus, devices must minimize time spent actually on the bus and maximize data transfer rates across the bus. To achieve these goals, the SCSI specification defines a bus release procedure (disconnection) which maintains the logical thread of the command but releases the bus when one device encounters a delay, such as a buffer full condition or electromechanical latency. To maximize data rates while on the bus, SCSI allows both asynchronous and synchronous transfers.

2.2 LEVELS OF SCSI USAGE

SCSI I/O systems can be connected in three distinct configurations: 1) single initiator/single target, 2) single initiator /multiple target, and 3) multiple initiator/multiple target. Basically, the configuration is determined by the number of hosts and peripherals on the SCSI bus. Each configuration adds one level of complexity and offers a significant increase in I/O performance potential. Fortunately, SCSI is designed so that devices which support the arbitration option can operate in any of the three configurations.

2.2.1 SINGLE INITIATOR / SINGLE TARGET

A majority of the SCSI systems in use are of the single initiator/single target configuration. Functionally, this configuration looks identical to the interface architecture shown in Figure 2-1, with only one host and only one peripheral device on the SCSI bus. The predominance of this architecture is due to the lack of different SCSI peripherals available in the past and not the lack of bus performance. As more SCSI peripherals become available, the single initiator/single target architecture can be upgraded to the single initiator/multiple target configuration discussed in section 2.2.2.

An important point to note about most existing systems is the lack of SCSI bus arbitration. A single initiator/multiple target configuration without SCSI bus arbitration does gain the advantage of connecting multiple peripheral devices through a single host adapter. But it forfeits two of the major SCSI bus features: peripheral-to-peripheral COPY capability and multi-threaded operation. (Both peripheral-to-peripheral COPY and multi-threading are discussed in later sections.)

2.2.2 SINGLE INITIATOR / MULTIPLE TARGET

Figure 2-3 shows a typical single initiator/multiple target configuration. In this architecture, the cost of the host adapter, or host interface port, is amortized over the two SCSI peripheral devices.

For systems without arbitration, the host may communicate with only one peripheral at a time and must remain connected to that peripheral until completion of the commanded operation. Because of this, the host must serve as a buffer memory for inter-peripheral communication. Systems with arbitration can support a wide array of performance features.

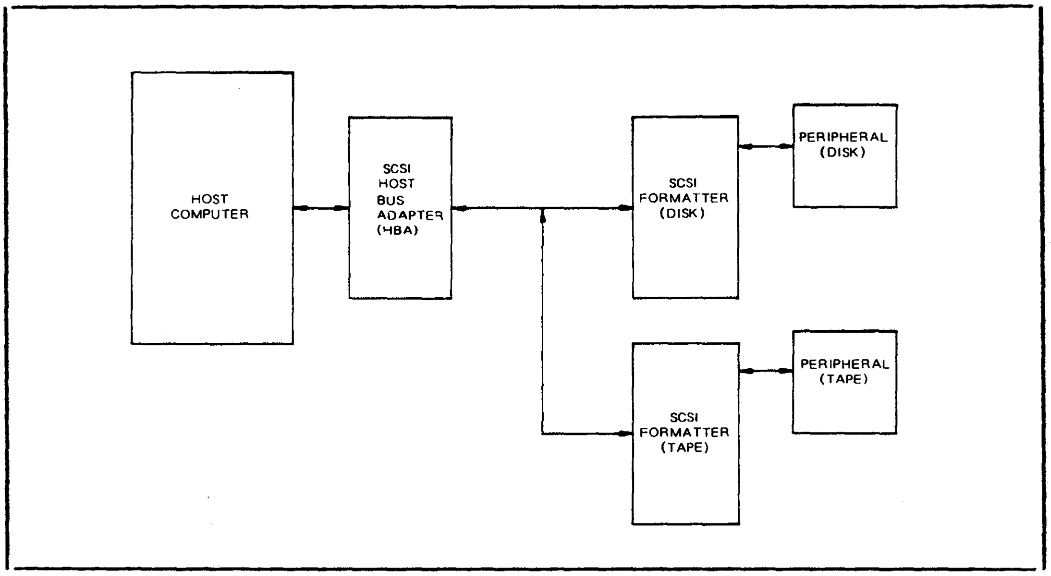


Figure 2-3: Single Initiator / Multiple Target Configuration

2.2.3 MULTIPLE INITIATOR/MULTIPLE TARGET

By far the most powerful SCSI configuration is the multiple initiator/multiple target system. Implied in the multiple initiator scenario is SCSI bus arbitration. Performance features of this configuration include:

- Host-to-Host Communication
- Direct Peripheral-to-Peripheral Data Transfer across the SCSI bus, with the COPY command
- Multi-Threaded Operations to a Peripheral Formatter
- Concurrent Operations (multi-tasking) on a Peripheral Formatter

Figure 2-4 shows the multiple initiator/multiple target architecture. This is the same basic configuration used by the IBM I/O channel discussed in Section 1.

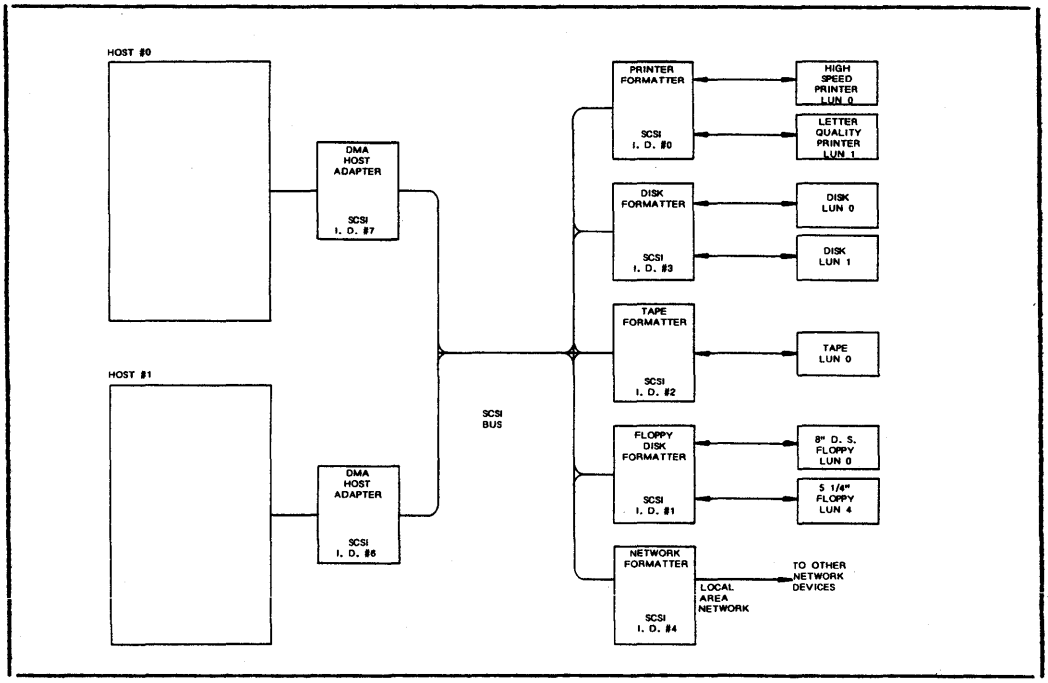

HOST #0
HOST #1
DMA HOST ADAPTER SCSI I. D. #7
DMA HOST ADAPTER SCSI I. D. #6
SCSI BUS
PRINTER FORMATTER SCSI I. D. #0
DISK FORMATTER SCSI I. D. #3
TAPE FORMATTER SCSI I. D. #2
FLOPPY DISK FORMATTER SCSI I. D. #1
NETWORK FORMATTER SCSI I. D. #4
HIGH SPEED PRINTER LUN 0
LETTER QUALITY PRINTER LUN 1
DISK LUN 0
DISK LUN 1
TAPE LUN 0
8" D. S. FLOPPY LUN 0
5 1/4" FLOPPY LUN 4
LOCAL AREA NETWORK
TO OTHER NETWORK DEVICES

Figure 2-4: Multiple Initiator / Multiple Target Configuration

2.3 SCSI SPECIFICATION SUMMARY

The versatility and performance features of the SCSI Bus are a direct result of the comprehensive specification developed by Shugart Associates and expanded upon by ANSI. The SCSI specification is more than a definition of the signals on the ribbon cable. The specification covers bus timing, device command sets, return status, a message system between devices and a DMA-oriented architecture for the exchange of commands, data and status between initiator and target.

2.3.1 BUS SIGNALS

The SCSI Bus consists of 18 signals, shown in Figure 2-6. Nine signals are for an eight-bit data bus with parity; the other nine signals are control lines with which SCSI devices coordinate bus access and transfers of command, data, status, and messages.

Status of the SCSI bus itself is a function of four control signals. These signals place the bus in one of eight phases:

BUS FREE phase

ARBITRATION phase

SELECTION phase

RESELECTION phase

```
COMMAND phase    \
DATA phase        \    These phases are collectively termed
STATUS phase      /    the information transfer phases.
MESSAGE  phase   /
```

The DATA and MESSAGE phases each have two variations, corresponding to the transfer direction being in or out, from the initiator's point of view. Figure 2-5 illustrates the SCSI phase sequence for arbitrating and non-arbitrating configurations.

All SCSI command sequences start from the BUS FREE phase. Arbitrating systems enter the ARBITRATION phase from BUS FREE phase while non-arbitration systems go directly to selection from BUS FREE. During arbitration, multiple SCSI devices arbitrate for the SCSI bus, with priority given to the highest-address SCSI device. An arbitrating system can have an initiator selecting a target via the SELECT phase, or a target reselecting an initiator via the RESELECT phase. The latter case is due to a previous disconnection by the target. (Disconnection is explained in later sections.) Non-arbitrating systems can only perform selection, since disconnection by the target is not allowed.

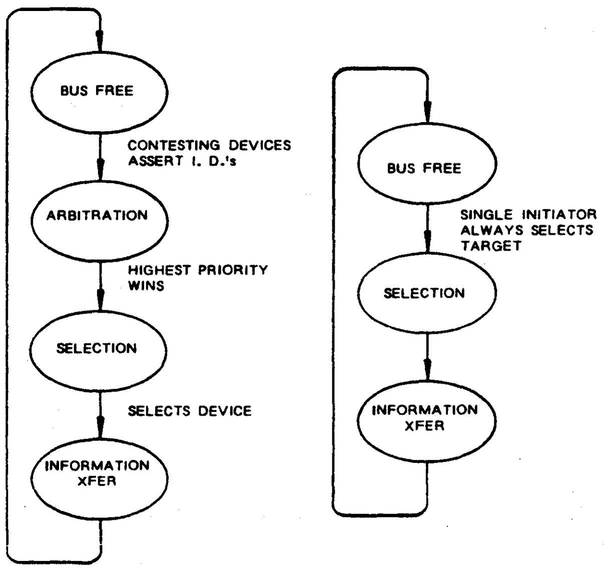


Figure 2-5: Bus Phase Diagrams

Figure 2-6 shows SCSI signal names, pin assignments for the single ended option, and brief definitions. Figure 2-7 shows which devices assert which signals during each phase.

Figure 2-6: SCSI Bus Signal Definition

BUS SIGNAL	DEFINITION	SINGLE ENDED OPTION PIN NUMBER
DB0*	Data Bus Bit 0	2
DB1*	Data Bus Bit 1	4
DB2*	Data Bus Bit 2	6
DB3*	Data Bus Bit 3	8
DB4*	Data Bus Bit 4	10
DB5*	Data Bus Bit 5	12
DB6*	Data Bus Bit 6	14
DB7*	Data Bus Bit 7	16
DBP*	Data Bus Parity	18
ATN*	Attention (Indicates INITIATOR has message to send target)	32
BSY*	Busy (SCSI bus is busy)	36
ACK*	Acknowledge (ACK of REQ/ACK Pair used for Information transfer on Data Bus)	38
RST*	Reset (Clears SCSI bus of all activity)	40
MSG*	Message (Indicates bus is in Message Transfer Phase)	42
SEL*	Select (Used to select/ reselect an SCSI Device)	44
C/D*	Command/Data (Indicates Command/Status Information Transfer or Data In/Out Transfer)	46
REQ*	Request (REQ of REQ/ACK Pair used for information transfer on Data Bus)	48
I/O*	Input/Output (Indicates direction of data flow on Data bus)	50

* indicates that the signal is negative true.

Figure 2-7: SCSI Signal Sources

	Signals				
Bus Phase	BSY	SEL	C/D, I/O, MSG, REQ	ACK/ATN	DB(7-0,P)
BUS FREE	None	None	None	None	None
ARBITRATION	All	Winner	None	None	SCSI ID
SELECTION	I&T	Init.	None	Initiator	Initiator
RESELECTION	I&T	Target	Target	Initiator	Target
COMMAND	Target	None	Target	Initiator	Initiator
DATA IN	Target	None	Target	Initiator	Target
DATA OUT	Target	None	Target	Initiator	Initiator
STATUS	Target	None	Target	Initiator	Target
MESSAGE IN	Target	None	Target	Initiator	Target
MESSAGE OUT	Target	None	Target	Initiator	Initiator

All:	The signal shall be driven by all SCSI devices that are actively arbitrating.
SCSI ID:	A unique data bit (the SCSI ID) shall be driven by each SCSI device that is actively arbitrating; the other seven data bits shall be released (i.e., not driven) by this SCSI device. The parity bit (DB(P)) may be undriven or driven to the true state, but shall never be driven to the false state during this phase.
I&T:	The signal shall be driven by the initiator, target, or both, as specified in the SELECTION phase and RESELECTION phase.
Initiator:	If this signal is driven, it shall be driven only by the active initiator.
None:	The signal shall be released; that is, not be driven by any SCSI device. The bias circuitry of the bus terminators pulls the signal to the false state.
Winner:	The signal shall be driven by the one SCSI device that wins arbitration.
Target:	If the signal is driven, it shall be driven only by the active target.

The information transfer phases can begin after a physical path has been established between two SCSI devices. These phases include six types of information exchange. Figure 2-8 shows the information transfer phases and associated control signal levels. The target qualifies each phase change by asserting REQ.

Figure 2-8: Information Transfer Phases

SCSI BUS			PHASE NAME	DESCRIPTION
MSG	C/D	I/O		
0	0	0	Data Out	Transfer data from initiator to target
0	0	1	Data In	Transfer data from target to initiator
0	1	0	Command	Transfer command from initiator to target
0	1	1	Status	Transfer status from target to initiator
1	1	0	Message Out	Transfer SCSI level message from initiator to target
1	1	1	Message In	Transfer SCSI level message from target to initiator

2.3.2 BUS PROTOCOL

The manner in which the SCSI devices interact, via the 18 SCSI bus signals, defines the bus protocol. The bus phases give a global view of the interaction, while the signal transition sequences within each phase completely specify the device interactions. The rigid definition of arbitration protocol, selection protocol and information transfer protocol provide sufficient framework for SCSI devices to be "plug compatible."

2.3.3 COMMAND / STATUS SET and MESSAGE SYSTEM DEFINITION

In order to obtain complete "plug compatibility" at a hardware and software level, a specification must go beyond the definition of bus interface and protocol. It must define command sets and return status blocks which will be common across SCSI devices from multiple vendors. The present SCSI specification defines command and status sets for the six types of devices shown in Figure 2-9.

Figure 2-9: SCSI Device Types

I/O DEVICE CLASS	I/O DEVICES COVERED
Direct Access Devices	Rigid Disks; both fixed and Removable Flexible Disk
Sequential Access Devices	Both Start/Stop and Streaming Tape
Printer Devices	Printers
Processor Devices	CPU's, Special Purpose Processors
Write-Once Read Multiple Devices	Some Optical Disks
Read-Only Direct Access Devices	Some Optical Disks

The SCSI specification defines a command set with which the host can determine from the device its I/O type. The INQUIRY command, implemented by all SCSI devices, returns a device type code along with descriptive parameters about the target device. With a complete set of device descriptions available via the INQUIRY command, the host OS can dynamically configure itself at power-on to work with whatever I/O devices are available.

SCSI also defines a message system by which two SCSI devices communicate information relating to bus and information transfer. These messages are relevant only to the SCSI devices (i.e., SCSI host bus adapter and SCSI device) and are transparent to any host operations.

2.3.3.1 DEVICE INDEPENDENT SOFTWARE INTERFACE

A major driving force behind the SCSI standard has been the need to reduce the system integration time for new peripherals. The typical OEM peripheral evaluation/integration cycle for a hard disk proceeds as follows:

1. Choose a drive or drives for evaluation based on performance, price and manufacturer's credentials.

2. Design, buy or utilize an existing controller to interconnect peripherals and test system.

3. Write, buy or utilize existing I/O drivers and test software to evaluate the drive(s).

4. After drive choice has been made, integrate hardware and I/O drivers into host system.

5. Test system extensively to ensure "Overall System Performance."

6. Begin system shipments.

Steps 1 through 5 can take from 1 to 18 months to complete. Depending on the class of drive selected, the OEM may only be able to evaluate one or two drives in that period.

A major design goal with SCSI was to shorten this cycle. Certainly a hardware interface standard for an intelligent disk drive will allow plug compatibility and shorten step 2 in the cycle. In order to shorten steps 3 and 4, software plug compatibility is required to allow utilization of existing I/O drivers and evaluation software.

SCSI I/O systems achieve software compatibility for hard disks by two means. First, SCSI defines a logical block (or sector) addressing scheme which is not related to the physical cylinder, head, and sector address. The number of logical blocks on a disk is a function of the disk size and block size, less spare blocks. For example, a 5MB disk formatted with 512 byte sectors will have block number addresses ranging from 0 to 8,000, while a 150 MB drive with the same sector size will contain 292,300 blocks.

The SCSI device formatter performs the conversion from absolute block to the physical cylinder, head and sector (see Figure 4-5). By viewing the disk as a string of consecutive blocks, the host system designer can write I/O drivers and evaluation software which will run any SCSI direct access device, independent of its physical configuration. The only information needed during actual system operation is the drive capacity, or maximum block number addressable on a particular drive.

The second provision which SCSI makes for software compatibility is the READ CAPACITY command, which provides the number discussed above - the maximum block number addressable on the device. This command returns both block size and block limit information to the host. In order to accommodate direct access devices of the future, the READ CAPACITY command allows for block sizes up to 2(exp32) and block addresses up to 2(exp32).

One method a host operating system may use to take advantage of these generic features is to perform the system generation (sysgen) task automatically during the cold boot rather than only upon command. The OS has software drivers for all acceptable devices which may be attached to the SCSI bus, but the OS activates and configures drivers only for devices present in the actual system. This allows unsophisticated users to add peripherals conveniently, without having to get factory assistance.

2.4 HOST MEMORY / HOST BUS ADAPTER / SCSI FORMATTER RELATION

SCSI architecture utilizes the concept of host memory blocks for command, data and status interchange between the host system and the SCSI formatter. In the middle of this exchange is the SCSI host bus adapter (HBA) which acts as the SCSI peripheral's gateway into host memory. The HBA is an important portion of the overall intelligence of SCSI. Along with providing an information path from the SCSI bus to the host bus, the HBA is intimately involved in assuring data integrity and proper performance of the I/O subsystem.

In order to understand SCSI operation fully, one must grasp the concepts of I/O memory blocks and logical threads. The host memory in Figure 2-10 (a single host/single peripheral configuration on the SCSI bus) contains three I/O blocks: command, data and status. The SCSI disk formatter needs to read the command block and write to the status block in order to perform the task specified by the host (in the command block). Likewise, the formatter must be able to read and write the data block.

The SCSI peripheral device "reaches into host memory" via the SCSI host adapter to exchange command and status information. But the host adapter must know the addresses of the commmand, data, and status blocks in order for it to "reach" into the right spot in memory. In other words, the host must give the host adapter a pointer to the start of each block. As the SCSI peripheral takes information from the command block, the memory pointer for the command block advances to the next byte. The same is true for the data and status pointers.

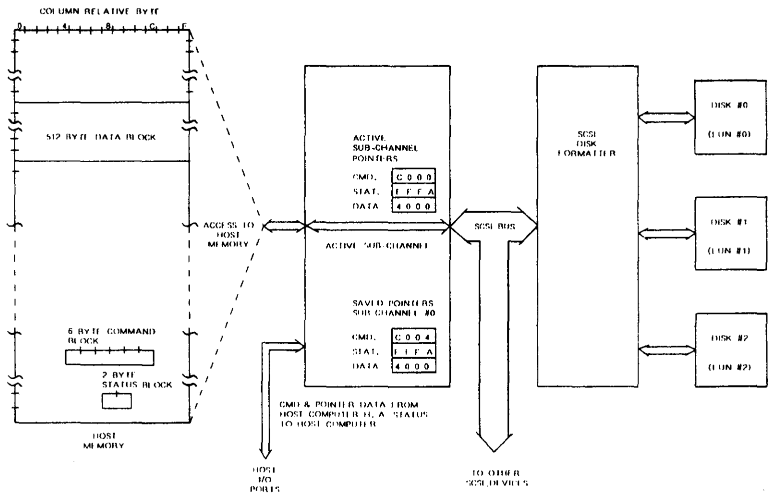


Figure 2-10: Host Memory / Host Adapter / SCSI Formatter

SCSI architecture provides for two sets of three pointers within the host adapter. The first set is known as the "Current (or Active) Pointer Values." These are the pointers to the next command, status or data byte to be transferred from host memory to the SCSI formatter. The second set is known as the "Saved Pointer Values." For command and status these pointers always point to the start of the memory command block and memory status block.

The "Saved Data Pointer" points to the start of the data block at the beginning of the SCSI operation. It remains at this value until the system performs a SAVE DATA POINTER function, which saves the value of the current data pointer. The system may retrieve this saved value by performing a RESTORE POINTERS function. This moves the saved value of each pointer into the current pointer registers. The current and saved pointers provide an efficient means of error retry and recovery during large data exchanges on the SCSI bus.

2.4.1 SCSI I/O SYSTEM READ DATA EXAMPLE

The best means to understanding the host/host adapter/SCSI device relationship is via an example. Consider a multi-sector read operation which will cross a cylinder boundary. While SCSI does not reference cylinders, the physical storage media is a disk, and seek latencies are an inherent characteristic of the device.

The first activity in the I/O operation is for the system to create a command descriptor block (CDB) in memory and determine where data and status are to be written in host memory. The host then sends an I/O command to the host adapter which includes the starting address (pointer) for each block and the SCSI address of the peripheral to perform the operation. In this example there is only one SCSI formatter and physical disk, but its address is required in order for the host adapter to select it.

Upon receiving the I/O command, the host adapter arbitrates for the SCSI bus and wins (due to the lack of competing devices) and proceeds to select the target SCSI device with the ATN (Attention) signal true. ATN indicates to the SCSI target that the initiator (the host adapter) has a message to send to the target.

After the SCSI bus SELECTION phase is completed, the disk formatter responds to the initiator's ATN condition by requesting a message from the initiator. This message, sent by the initiator, indicates the desired LUN on the target formatter and indicates whether the initiator can support bus disconnect. In this example, both the host and the formatter support disconnect.

I/O activity from this point will be controlled entirely by the target. The host adapter is simply an "arm" of the target used to reach into host memory. Utilizing this arm, the target reads in the CDB.

After decoding the instruction, the formatter determines that it must execute a disk seek to get the starting data block. Since the formatter cannot transfer data across the SCSI bus until it has read data from the disk, the target formatter disconnects from the bus by sending a DISCONNECT message to the initiator. Since the target has not yet sent any data, it does not need to send a SAVE DATA POINTER message prior to the DISCONNECT message. Finally, the target releases the BUSY signal, which returns the SCSI bus to the BUS FREE phase.

Although the initiator host adapter and target disk formatter are now physically disconnected, they are still logically connected or "threaded" together. Both devices know they have an I/O command to finish. The target must reconnect to the initiator to continue the transfer at a later time. The principle of logical threads allows many I/O commands to execute in the system simultaneously, utilizing a single physical bus. The thread is actually not between the host adapter and the disk formatter, but runs all the way from the host memory I/O block set to the physical LUN performing the operation (see Figure 2-11).

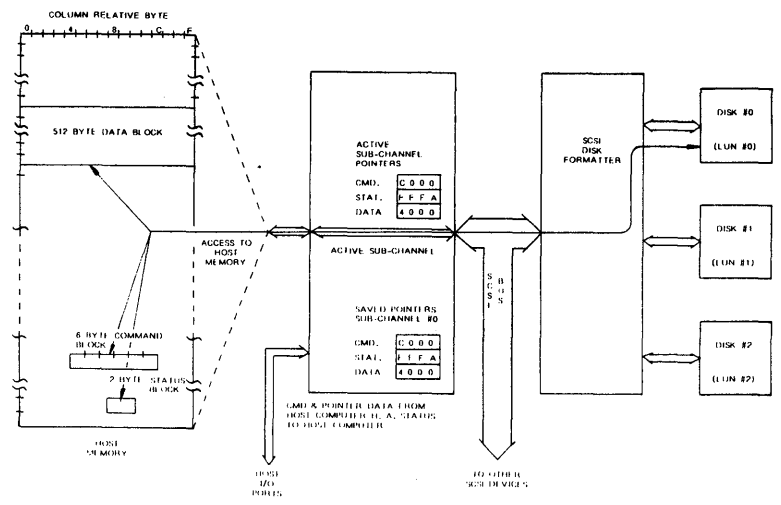


Figure 2-11: Logical Thread

Once the target has started filling its data buffers, it can transmit data to the initiator, but first it must re-establish the physical path, which it does via the RESELECTION phase. After the target arbitrates for the bus, wins control of it, and reselects the initiator, it sends an IDENTIFY message to the host adapter to indicate which target LUN is reconnecting. This information provides reconnection to the correct thread into host memory and implies a RESTORE DATA POINTER message. The host must execute the implied RESTORE DATA POINTER message prior to completing the IDENTIFY message.

After reconnection, the roles of the initiator and target are just as they were prior to disconnect. The target transfers data into host memory via the host adapter. The data transfer continues until the disk reaches the end of its cylinder and the disk formatter determines that it must execute a second physical seek to complete the I/O read command. The target must disconnect again, but this time, since it has started the data transfer, it must first send a SAVE DATA POINTER message to the initiator to save the current data pointer. Then the target sends the DISCONNECT message, which breaks the physical thread and frees the SCSI bus for use by other devices.

After seek completion and transfer of data into its buffer, the formatter reconnects to the host adapter and completes the data transfer as requested by the I/O read command. At this point, the formatter writes ending status into host memory via the host adapter. The final action of the target is to send the host adapter a command complete message and disconnect from the SCSI bus. The target has completed its operation and breaks the logical thread.

Upon receipt of the command complete message, the host adapter signals the host that the I/O command is complete. This signal can be an interrupt or the setting of a flag read by the host in a polled I/O environment. This action by the host adapter breaks the thread between the host adapter and the I/O memory blocks of the host. The host reviews the status of the operation in the status block and proceeds to utilize the data transferred into the data block.

2.4.2 I/O CHANNEL CONCEPT

While not defined by the SCSI specification, the I/O Channel concept fully utilizes the high performance capability of SCSI. The I/O channel is basically an intelligent SCSI host adapter which can maintain multiple simultaneous threads between host memory I/O blocks and different SCSI devices.

The I/O channel utilizes a single DMA path into host memory, supporting the DMA operations of numerous SCSI devices. Since the SCSI bus is a single physical bus and most host computers have a single physical backplane bus, multiple DMA channels into memory are not necessary. In a multiple DMA channel architecture, when a channel is accessing memory, all other channels are idle. Hence, a single channel supporting multiple threads can supply the same performance as separate DMA peripherals. Advantages to the host include lower system cost and fewer backplane card slots used.

In the I/O read example discussed in Section 2.4, the I/O channel is the SCSI host adapter. The host gives the I/O channel a command by providing it with pointers to the I/O memory blocks and the SCSI peripheral address. This establishes a thread between the host adapter and the host I/O memory blocks. The I/O channel then opens a subchannel, which is assigned the task of managing the physical link between the host adapter and the target formatter and the logical thread between host memory and the LUN. This subchannel supervises all physical connections and reconnections to the host adapter. The number of active or open subchannels an I/O channel can support is totally dependent upon its design. The SCSI definition could, in theory, support an I/O channel with up to 56 simple subchannels.

Upon completion of an I/O command by the target formatter, the subchannel notifies the host of the I/O operation completion. After the host has been notified, the I/O subchannel is considered inactive or closed and is ready to be assigned to another I/O command.

The complexity and capabilities which make the I/O channel such a powerful system should be clearly understood. While it is called a host adapter, the I/O channel is, in fact, an intelligent subsystem within itself and not a simple hardware coupler between the host bus and SCSI bus.

2.4.3 I/O CHANNEL OPERATION EXAMPLES

Detailed examples best illustrate the principles and capabilities of I/O channel operation. The following two sections provide both summaries and very detailed examples of such operation.

Example #1 deals with a multiple host system in which disk, tape and host-to-host transfers take place. Example #2 presents a tape-to-disk copy with concurrent host disk accesses.

2.4.3.1 EXAMPLE #1

This example demonstrates a system scenario in which the I/O system performs four concurrent operations. Figure 2-12 shows the system architecture and available peripherals. In order to give the example a sense of proportion , units of time (ms) are assigned to I/O operations. While these times are reasonable, they are for the purpose of example only. Times in a real system depend upon individual SCSI product architecture and hardware and firmware implementations.

The example starts at t = 0ms with no I/O activity. At a relative time t = 1.0ms, host #0 issues a disk READ command to SCSI device #3/LUN #0. A short time later, at t = 3.0 Host #1 issues a disk READ command to the same SCSI device but on LUN #1. At t = 35.0 host #0 issues a tape WRITE command to SCSI device #2/LUN #0. Finally at t = 43.0ms host #1 sends a SEND command to Host #0.

Figure 2-13 shows a time line diagram for each of the four I/O operations and the activity occurring on the SCSI bus. Note the relatively low percentage loading on the SCSI bus. The example covers a period of 346ms. During this time, the SCSI bus is busy for 16.5ms or 4.7% of the time. This low loading value is a result of high performance peripheral design. All the devices in the example acquire the bus only when they have information to transfer to another device. The information transfer occurs at a high rate. After each transfer, the device releases the bus. With a 4.7% loading factor, the bus could support many additional concurrent I/O operations before the system noticed any degradation in performance due to bus loading.

Another aspect of the example to consider is the delay associated with each I/O operation due to the fact that it could not obtain the bus at the exact instant it was required. Figure 2-14 provides a comparison between the minimum time the instruction could execute if it had exclusive control of the bus, and the actual time required in the example.

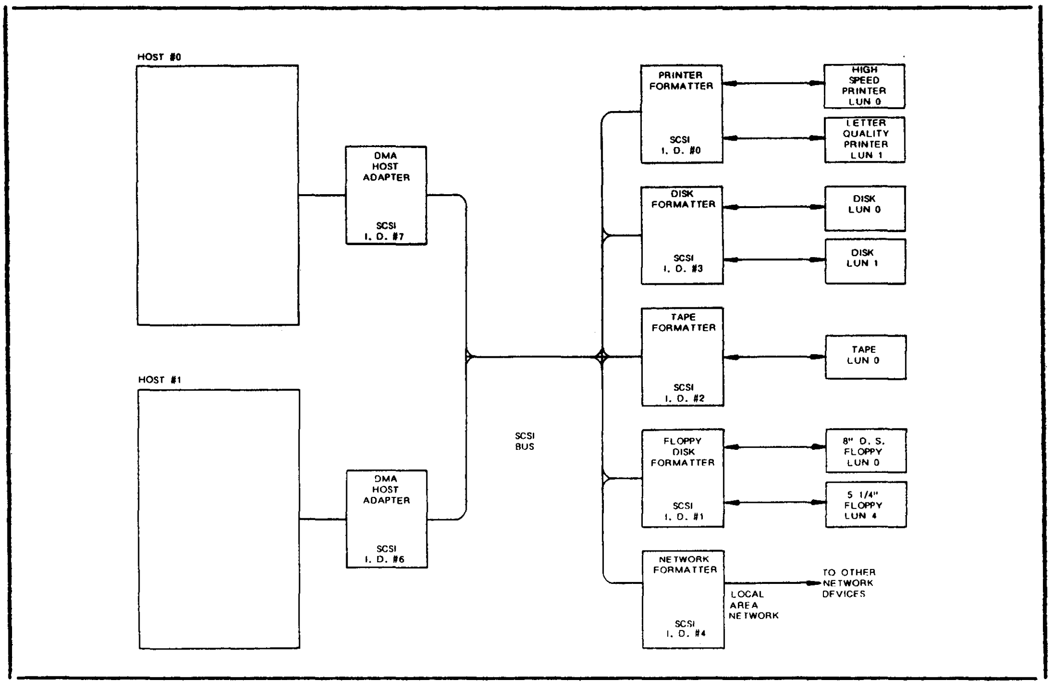

HOST #0
HOST #1
DMA HOST ADAPTER SCSI I. D. #7
DMA HOST ADAPTER SCSI I. D. #6
SCSI BUS
PRINTER FORMATTER SCSI I. D. #0
DISK FORMATTER SCSI I. D. #3
TAPE FORMATTER SCSI I. D. #2
FLOPPY DISK FORMATTER SCSI I. D. #1
NETWORK FORMATTER SCSI I. D. #4
HIGH SPEED PRINTER LUN 0
LETTER QUALITY PRINTER LUN 1
DISK LUN 0
DISK LUN 1
TAPE LUN 0
8" D. S. FLOPPY LUN 0
5 1/4" FLOPPY LUN 4
LOCAL AREA NETWORK
TO OTHER NETWORK DEVICES

Figure 2-12: I/O System Configuration

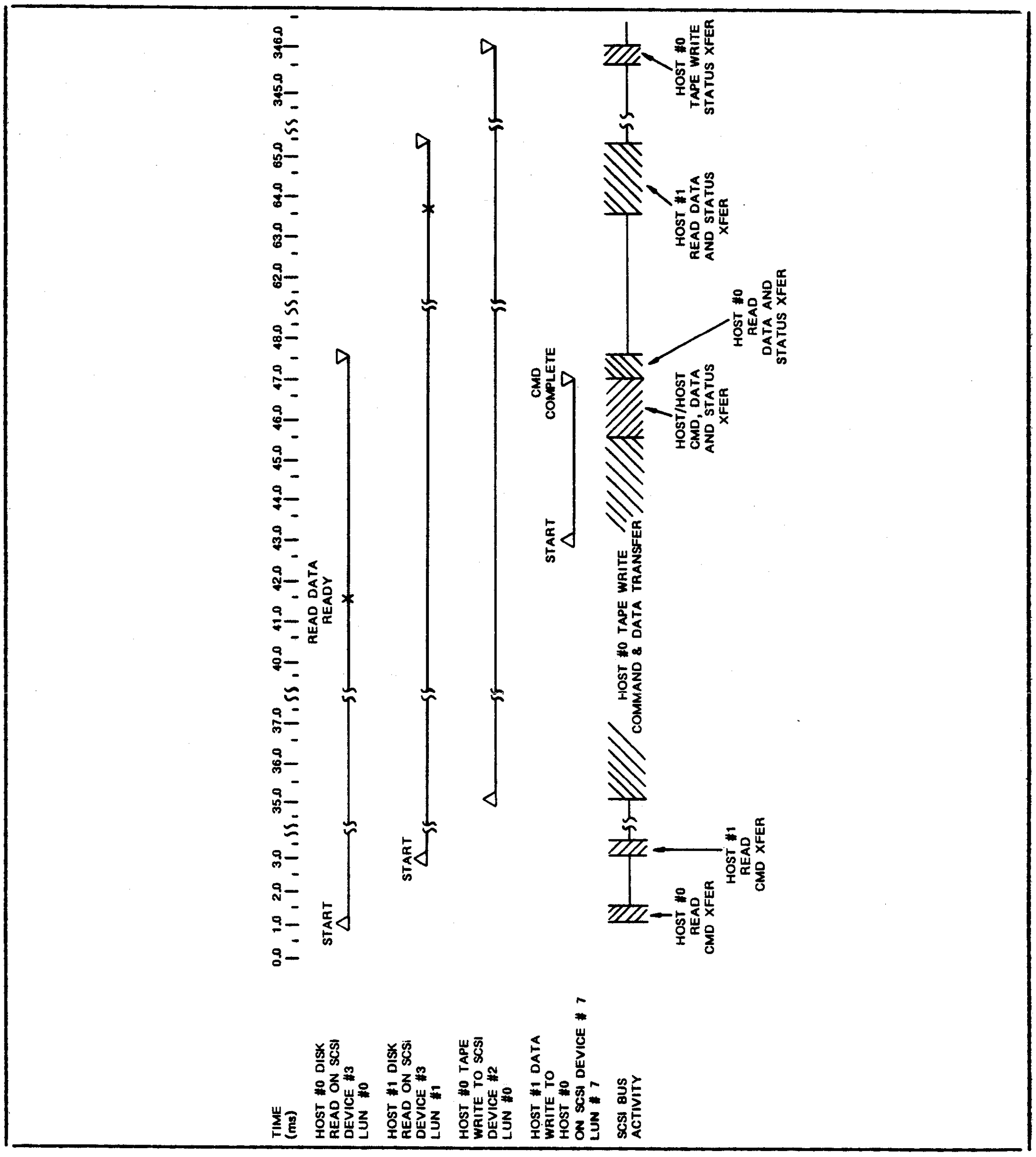

TIME (ms)
0.0 1.0 2.0 3.0 35.0 36.0 37.0 40.0 41.0 42.0 43.0 44.0 45.0 46.0 47.0 48.0 62.0 63.0 64.0 65.0 345.0 346.0
HOST #0 DISK READ ON SCSI DEVICE #3 LUN #0
START
READ DATA READY
HOST #1 DISK READ ON SCSI DEVICE #3 LUN #1
START
HOST #0 TAPE WRITE TO SCSI DEVICE #2 LUN #0
HOST #1 DATA WRITE TO HOST #0 ON SCSI DEVICE # 7 LUN # 7
START
CMD COMPLETE
SCSI BUS ACTIVITY
HOST #0 TAPE WRITE COMMAND & DATA TRANSFER
HOST #0 READ CMD XFER
HOST #1 READ CMD XFER
HOST/HOST CMD, DATA AND STATUS XFER
HOST #0 READ DATA AND STATUS XFER
HOST #1 READ DATA AND STATUS XFER
HOST #0 TAPE WRITE STATUS XFER

Figure 2-13: Example #1 Time Line Diagram

Figure 2-14: I/O Channel Performance Evaluation

COMMAND	MINIMUM TIME FOR EXECUTION	EXECUTION TIME IN EXAMPLE	DIFFERENCE	
			ACTUAL	% INCREASE OVER MINIMUM
Host #0 Disk READ	41.0ms	46.5ms	5.5ms	13.4%
Host #1 Disk READ	63.0ms	63.0ms	0.0ms	0.0%
Host #0 Tape WRITE	312.0ms	312.0ms	0.0ms	0.0%
Host/Host Communication	1.5ms	4.0ms	2.5ms	166.6%

Example #1 Detail

The example details are presented in terms of system snapshots and system periods. The system snapshot describes the state of the overall system and peripherals in the system at an instant in time. Some system snapshots will include a figure which graphically depicts the snapshot status. The system period describes activities occurring in the system during a specific time segment.

This example demonstrates the following system level features and functions:

- Host/disk transfer
- Disconnect/Reconnect
- Multi-Tasking on a disk formatter (concurrent command execution)
- Host/Tape transfer
- Host/Host communication
- Four concurrent system threads
- Arbitration between SCSI devices

Figure 2-12 shows the system configuration for this example.

SYSTEM SNAPSHOT
t = 0ms

SCSI BUS STATUS: BUS FREE

DEVICES WAITING TO ARBITRATE: None

EXISTING THREADS: None

SCSI DEVICE TASKS IN PROGRESS: None

COMMENTS:
1. System is in an idle state; no I/O commands are in progress

---

SYSTEM SNAPSHOT
t = 1.0ms

SYSTEM EVENT: Host #0 issues I/O channel command
(Host #0 disk Read from SCSI device #3 LUN #0)

SCSI BUS STATUS: BUS FREE

DEVICES WAITING TO ARBITRATE: None

EXISTING THREADS: 1. Host #0 CCB #0 to host adapter #0

SCSI DEVICE TASKS IN PROGRESS:

COMMENTS:
1. Host #0 has built command control block (CCB) #0 and issued an I/O channel command (requesting a READ from SCSI device #3 LUN #0), establishing a thread between the CCB and host adapter.

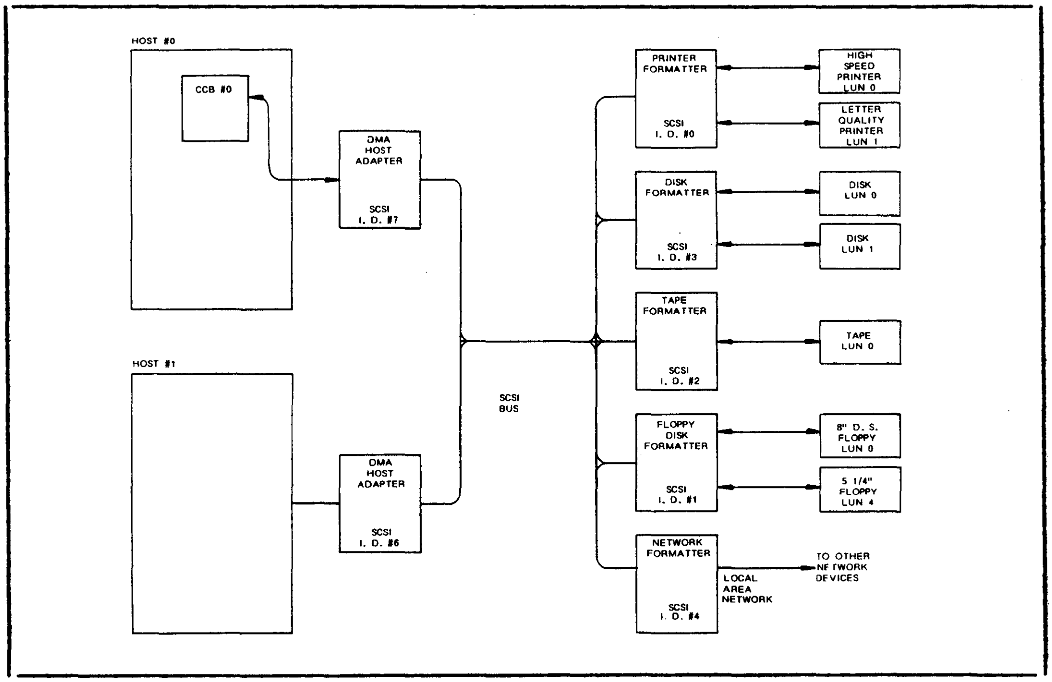


Figure 2-15: System Snapshot t = 1.0ms

SYSTEM PERIOD
t = 1.0ms to t = 1.5ms

SYSTEM ACTIVITY:

After receiving the I/O channel command, the host adapter places the SCSI bus in an arbitration phase and wins the bus since there are no competing arbitrators. Following arbitration, the host adapter selects the desired disk formatter and LUN, establishing the logical thread between the CCB and physical disk. The target formatter fetches the command from the host CCB via the host adapter DMA path to the command descriptor block. Once the target determines that a seek will be required to obtain the requested data, it disconnects, allowing other devices to be used on the bus.

SYSTEM SNAPSHOT
t = 1.5ms

SYSTEM EVENT: Disconnect by target formatter (SCSI device #3)

SCSI BUS STATUS: BUS FREE

DEVICES WAITING TO ARBITRATE: None

EXISTING THREADS: 1. Host #0 CCB #0 to SCSI device #3/LUN #0

SCSI DEVICE TASKS IN PROGRESS:
1. Disk formatter beginning seek operation on LUN #0
2. Host adapter #0 I/O subchannel #0 watching for reconnect with disk formatter (SCSI device #3)

COMMENTS:
1. The physical seek to the required cylinder and additional latencies of the disk and formatter will take 40.0 ms.

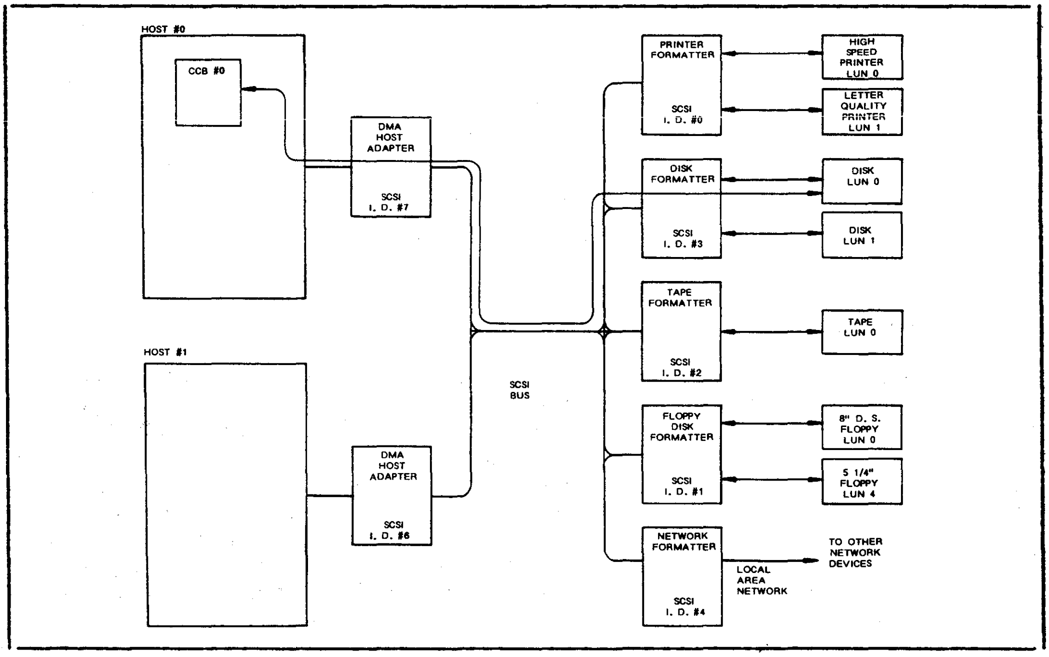


Figure 2-16: System Snapshot t = 1.5ms

SYSTEM SNAPSHOT
t = 2.0ms

SYSTEM EVENT: Host #0 issues a RECEIVE command to Host adapter #0

SCSI BUS STATUS: BUS FREE

DEVICES WAITING TO ARBITRATE: None

EXISTING THREADS: 1. Host #0 CCB #0 to SCSI device #3/LUN #0
2. Host #0 CCB #1 to SCSI host adapter #0 LUN #7

SCSI DEVICE TASKS IN PROGRESS:
1. Disk formatter #3 performing seek operation on LUN #0
2. Host adapter #0 I/O subchannel watching for reconnect with disk formatter (SCSI device #3 LUN #0)

COMMENTS:
1. Host adapter #0 can now receive host-to-host messages.

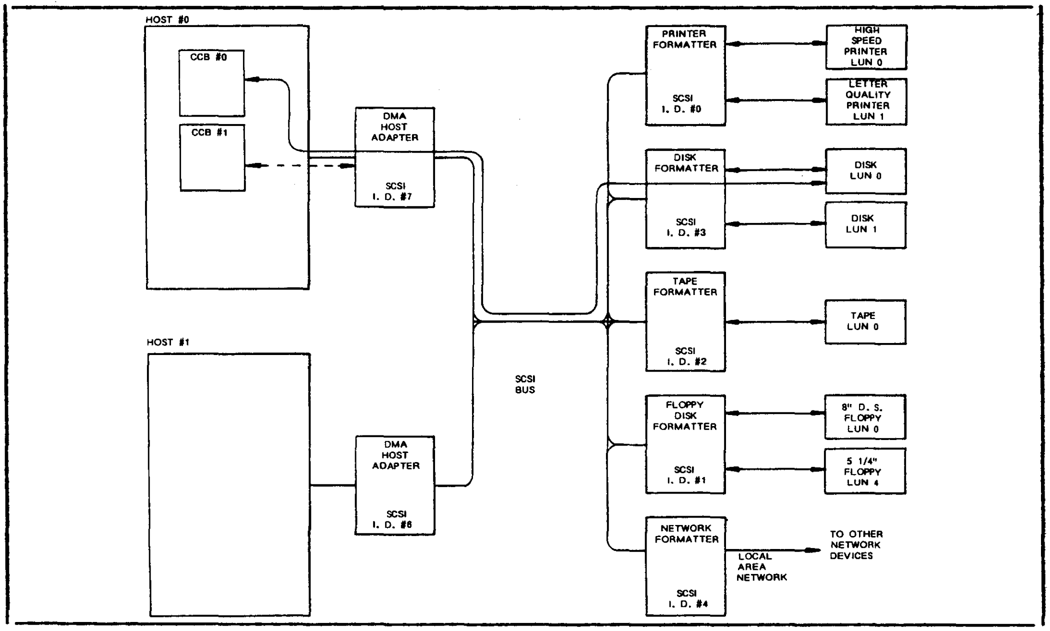

HOST #0
CCB #0
CCB #1
DMA HOST ADAPTER SCSI I. D. #7
HOST #1
DMA HOST ADAPTER SCSI I. D. #6
SCSI BUS
PRINTER FORMATTER SCSI I. D. #0
HIGH SPEED PRINTER LUN 0
LETTER QUALITY PRINTER LUN 1
DISK FORMATTER SCSI I. D. #3
DISK LUN 0
DISK LUN 1
TAPE FORMATTER SCSI I. D. #2
TAPE LUN 0
FLOPPY DISK FORMATTER SCSI I. D. #1
8" D. S. FLOPPY LUN 0
5 1/4" FLOPPY LUN 4
NETWORK FORMATTER SCSI I. D. #4
LOCAL AREA NETWORK
TO OTHER NETWORK DEVICES

Figure 2-17: System Snapshot t = 2.0ms

SYSTEM SNAPSHOT
t = 3.0ms

SYSTEM EVENT: Host #1 issues I/O channel command (Read 4 sectors from SCSI device #3 LUN #1)

SCSI BUS STATUS: BUS FREE

DEVICES WAITING TO ARBITRATE: None

EXISTING THREADS:
1. Host #0 CCB #0 to SCSI device #3/LUN #0
2. Host #0 CCB #1 to host adapter #0 LUN #7 (RECEIVE command)
3. Host #1 CCB #0 to SCSI host adapter #1

SCSI DEVICE TASKS IN PROGRESS:
1. Disk formatter performing seek operation on LUN #0
2. Host adapter #0 I/O subchannel watching for reconnect with disk formatter
3. Host adapter #0 watching for incoming communication from another host system

COMMENTS:
1. Host #1 issues disk READ command to SCSI device #3 LUN #1. Thread has been established between CCB and host adapter #1.

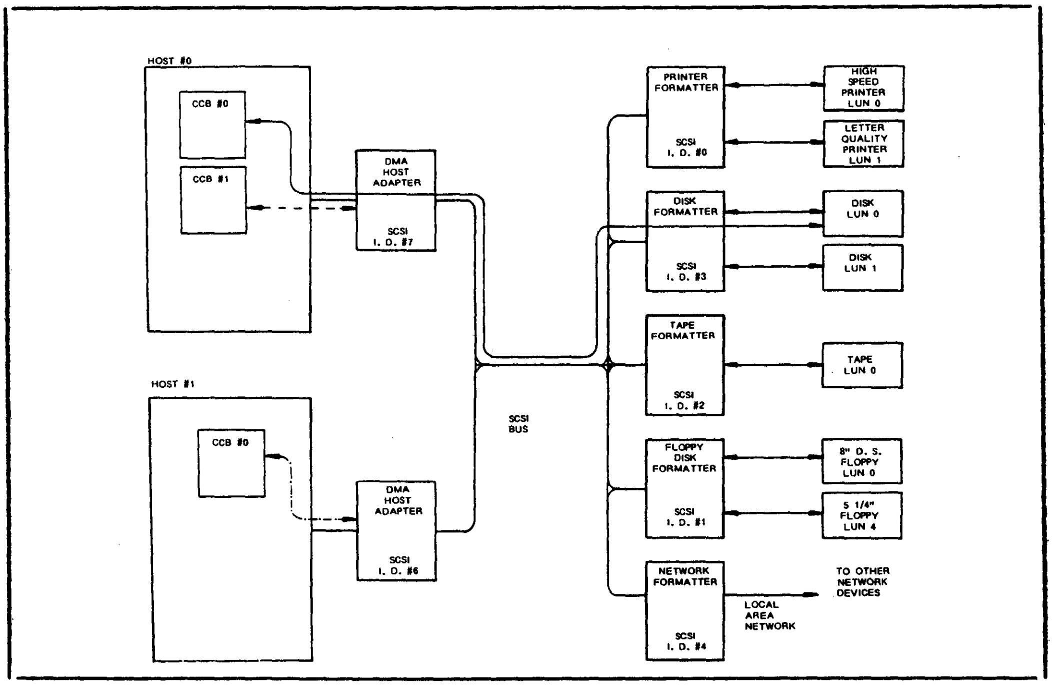


Figure 2-18: System Snapshot t = 3.0ms

SYSTEM PERIOD
t = 3.0ms to t = 3.5ms

SYSTEM ACTIVITY:

After receiving the I/O channel command, host adapter #1 places the SCSI bus in the ARBITRATION phase and wins the bus since there are no competing arbitrators. Following ARBITRATION, the host adapter selects the desired disk formatter (#3) and LUN (#1), establishing the logical thread. The target must now perform two tasks. First, it monitors the seek operation being performed on LUN #0; and second, it fetches the command from the host #1 CCB via the host adapter DMA path to the CDB. Once the target determines that a seek on LUN #1 will be required to obtain the requested data, the target formatter disconnects, allowing other devices to use the bus.

SYSTEM SNAPSHOT
t = 3.5ms

SYSTEM EVENT: Disconnect by target formatter

SCSI BUS STATUS: BUS FREE

DEVICES WAITING TO ARBITRATE: None

EXISTING THREADS:
1. Host #0 CCB #0 to SCSI device #3/LUN #0
2. Host #0 CCB #1 to host adapter #0 LUN #7 (RECEIVE communication)
3. Host #1 CCB #0 to SCSI device #3/LUN #1

SCSI DEVICE TASKS IN PROGRESS:
1. Disk formatter monitoring seek operation on LUN #0
2. Host adapter #0 I/O subchannel watching for reconnect with disk formatter #3 LUN #0
3. Host adapter #0 watching for incoming communication from another host system
4. Disk formatter beginning seek operation on LUN #1
5. Host adapter #1 I/O subchannel watching for reconnect with disk formatter (SCSI device #3 LUN #1)

COMMENTS:
1. The physical seek to the required cylinder (on LUN #1) and additional processing latencies of the disk and formatter will take 60.0 ms.

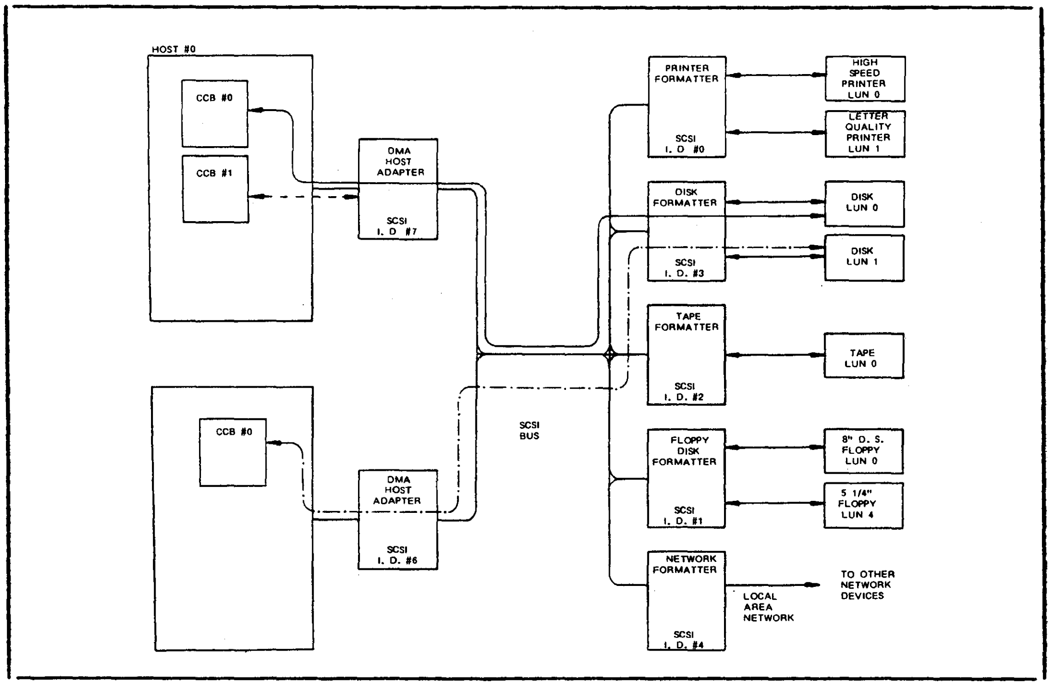


Figure 2-19: System Snapshot t = 3.5ms

SYSTEM SNAPSHOT
t = 35.0ms

SYSTEM EVENT: Host #0 issues I/O channel command
(WRITE to tape device #2 LUN #0)

SCSI BUS STATUS: BUS FREE

DEVICES WAITING TO ARBITRATE: None

EXISTING THREADS:
1. Host #0 CCB #0 to SCSI device #3/LUN #0
2. Host #0 CCB #1 to host adapter #0 LUN #7 (RECEIVE communication)
3. Host #1 CCB #0 to SCSI device #3/LUN #1
4. Host #0 CCB #2 host adapter #0

I/O CHANNEL READ/WRITE EXAMPLE

SCSI DEVICE TASKS IN PROGRESS:
1. Disk formatter monitoring seek operation on LUN #0
2. Host adapter #0 I/O Subchannel watching for reconnect with disk formatter #3 LUN #0
3. Host adapter #0 watching for incoming communication from another host system
4. Disk formatter monitoring seek operation on LUN #1
5. Host adapter #1 I/O subchannel watching for reconnect with disk formatter (SCSI device #3 LUN #1)

COMMENTS:
1. Tape WRITE command to SCSI device #2 requested by Host #0.

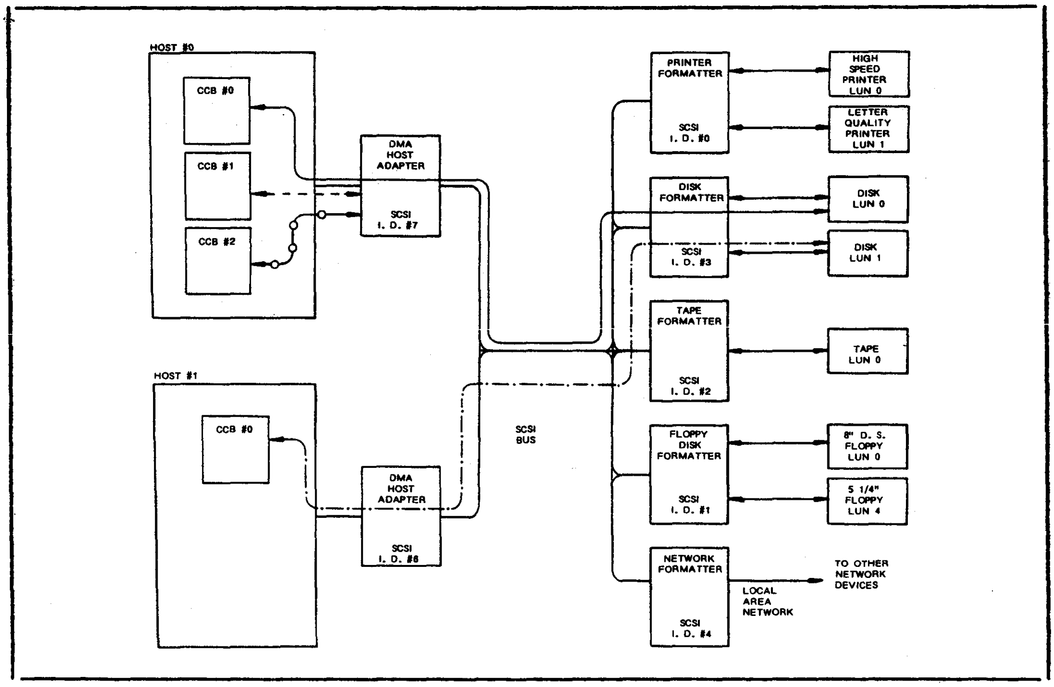


Figure 2-20: System Snapshot t = 35.0ms

SYSTEM PERIOD
t = 35.0ms to t = 35.5ms

SYSTEM ACTIVITY:
After receiving the I/O channel command, host adapter #0 places the SCSI bus in the ARBITRATION phase and wins the bus since there are no competing arbitrators. Following ARBITRATION the host adapter selects the desired tape formatter (#2) and LUN (#0), thus establishing the logical thread. The target formatter fetches the command from the host CCB #2.

SYSTEM PERIOD
t = 35.5ms to t =45.5 ms

SYSTEM ACTIVITY:
The tape formatter determines that it can accept data from the host and transfers the 10K bytes of data specified in the WRITE command. After the data transfer is complete, the tape formatter will disconnect and write the data in its buffer onto the tape. A period of 300ms (200ms tape ramp up time and 100ms actual data transfer time) is required to transfer the write data from the data buffer onto the tape.

SYSTEM SNAPSHOT
t = 41.5ms

SYSTEM EVENT: Disk formatter #3 has data from LUN #0 ready to send to host adapter #0

SCSI BUS STATUS: Bus involved in data transfer for tape WRITE command
Initiator: SCSI device #7 (host adapter #0)
Target: SCSI device #2 (tape formatter)

DEVICES WAITING TO ARBITRATE: Disk formatter (SCSI device #3)

EXISTING THREADS:
1. Host #0 CCB #0 to SCSI device #3/LUN #0
2. Host #0 CCB #1 to host adapter #0 LUN #7 (RECEIVE communication)
3. Host #1 CCB #0 to SCSI device #3/LUN #1
4. Host #0 CCB #2 to SCSI device #2/LUN #0 (Also the SCSI physical path)

SCSI DEVICE TASKS IN PROGRESS:

1. Tape WRITE command DATA transfer phase in progress
2. Disk formatter (LUN #0) waiting for transfer data from buffer to host
3. Host adapter #0 I/O subchannel watching for reconnect with disk formatter #3/LUN #0
4. Host adapter #0 watching for incoming communication from another host system
5. Disk formatter monitoring seek operation on LUN #1
6. Host adapter #1 I/O subchannel watching for reconnect with disk formatter (SCSI device #3 LUN #1)

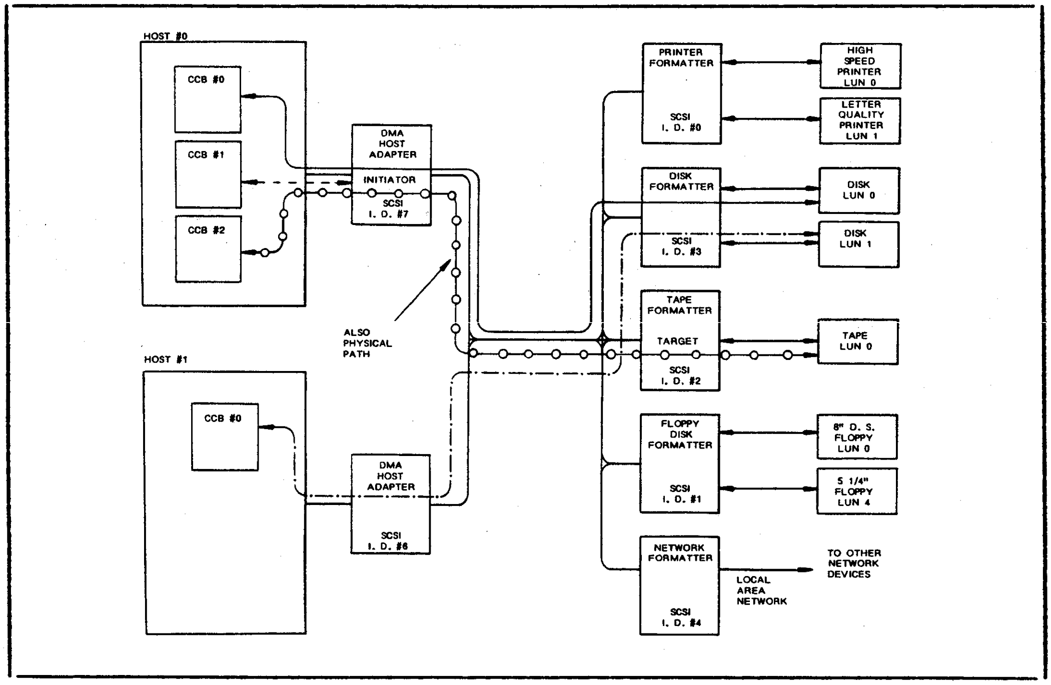


Figure 2-21: System Snapshot t = 41.5ms

SYSTEM SNAPSHOT
t = 43.0ms

SYSTEM EVENT: Host #1 issues a "Write to Host #0" I/O channel command

SCSI BUS STATUS: Bus involved in data transfer for tape WRITE command
Initiator: SCSI device #7 (host adapter #0)
Target: SCSI device #2 (tape formatter)

DEVICES WAITING TO ARBITRATE: Disk formatter (SCSI device #3)

EXISTING THREADS:
1. Host #0 CCB #0 to SCSI device #3/LUN #0
2. Host #0 CCB #1 to host adapter #0/LUN #7 (RECEIVE communication)
3. Host #1 CCB #0 to SCSI device #3/LUN #1
4. Host #0 CCB #2 to SCSI device #2/LUN #0 (Also the SCSI physical path)
5. Host #1 CCB #1 to host adapter #1

SCSI DEVICE TASKS IN PROGRESS:
1. Tape WRITE command DATA transfer phase in progress
2. Disk formatter (LUN #0) waiting for transfer data from buffer to host
3. Host adapter #0 I/O subchannel watching for reconnect with disk formatter #3/LUN #0
4. Host adapter #0 watching for incoming communication from another host system
5. Disk formatter monitoring seek operation on LUN #1
6. Host adapter #1 I/O subchannel watching for reconnect with disk formatter (SCSI device #3 LUN #1)

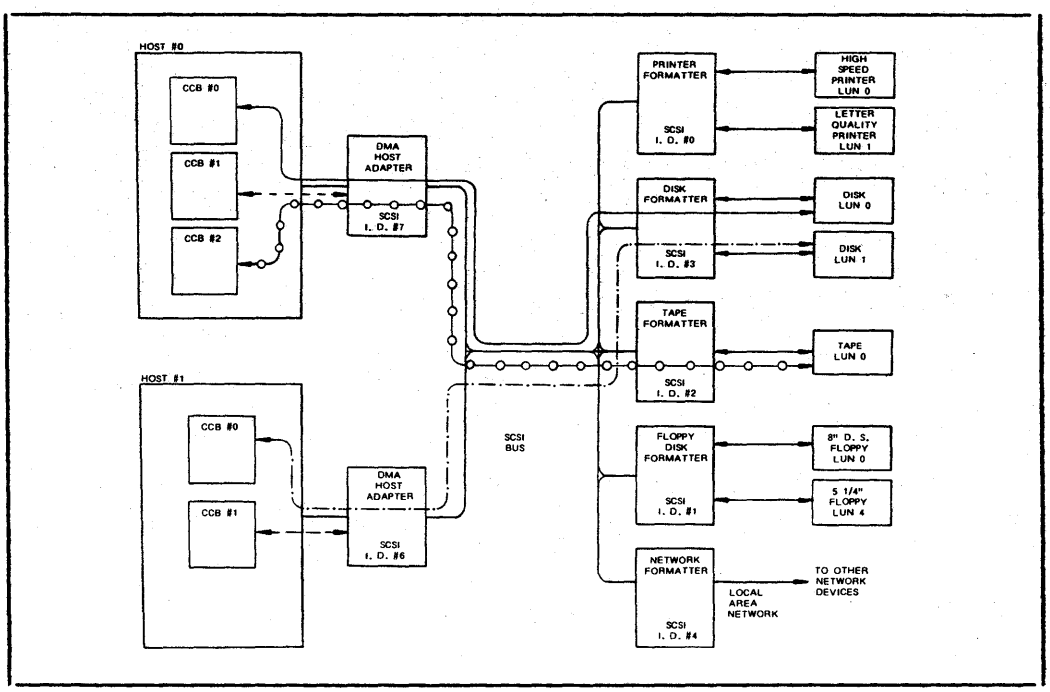


Figure 2-22: System Snapshot t = 43.0ms

SYSTEM PERIOD
t = 43.0ms to t =43.5 ms

SYSTEM ACTIVITY:
Host adapter processes I/O channel command and determines it must obtain the SCSI bus in order to perform the RECEIVE command issued. Since the bus is currently busy transferring data between host #0 and the tape formatter, the I/O subchannel of host adapter #1, which will transmit the data requested by the RECEIVE command, waits to arbitrate.

SYSTEM SNAPSHOT
t = 45.5ms

SYSTEM EVENT: Tape formatter releases SCSI bus after filling its buffer

SCSI BUS STATUS: BUS FREE

DEVICES WAITING TO ARBITRATE:
1. Disk formatter (SCSI device #3)
2. Host adapter #1 (SCSI device #6)

EXISTING THREADS:
1. Host #0 CCB #0 to SCSI device #3/LUN #0
2. Host #0 CCB #1 to host adapter #0/LUN #7 (RECEIVE communication)
3. Host #1 CCB #0 to SCSI device #3/LUN #1
4. Host #0 CCB #2 to SCSI device #2/LUN #0 (Also the SCSI physical path)
5. Host #1 CCB #1 to host adapter #1

SCSI DEVICE TASKS IN PROGRESS:
1. Tape formatter repositioning tape.
2. Host adapter #0 I/O subchannel watching for reconnect with tape formatter (SCSI device #2/LUN #0)
3. Disk formatter (SCSI device #3/LUN #0) waiting to transfer data from buffer to host
4. Host adapter #0 I/O subchannel watching for reconnect with disk formatter #3/LUN #0
5. Host adapter #0 watching for incoming communication from another host system
6. Disk formatter monitoring seek operation on LUN #1
7. Host adapter #1 I/O subchannel watching for reconnect with disk formatter (SCSI device #3 LUN #1)
8. Host adapter #1 I/O subchannel waiting for bus to complete host RECEIVE command

---

SYSTEM PERIOD
t = 45.5ms to 46.0ms

Once the bus has been free for at least 1200 nano-seconds, both host adapter #1 and the disk formatter will arbitrate for the bus. Since the host adapter SCSI address is higher than the disk formatter, it will win arbitration and control the bus. Host adapter #1, the target, now reselects host adapter #0, the initiator for the command.

SYSTEM SNAPSHOT
t = 46.0ms

SYSTEM EVENT: Physical path from host #1 to host #0 complete

SCSI BUS STATUS: Information Transfer Phase
Initiator: Host adapter #0
Target: Host adapter #1

DEVICES WAITING TO ARBITRATE: 1. Disk formatter (SCSI device #3)

EXISTING THREADS:
1. Host #0 CCB #0 to SCSI device #3/LUN #0
2. Host #1 CCB #1 to host #0 CCB #1 (Also the SCSI physical path)
3. Host #1 CCB #0 to SCSI device #3/LUN #1
4. Host #0 CCB #2 to SCSI device #2/LUN #0

SCSI DEVICE TASKS IN PROGRESS:
1. Tape formatter repositioning tape.
2. Host adapter #0 I/O subchannel watching for reconnect with tape formatter (SCSI device #2/LUN #0)
3. Disk formatter (SCSI device #3/LUN #0) waiting to transfer data from buffer to Host
4. Host adapter #0 I/O subchannel watching for reconnect with disk formatter #3/LUN #0
5. Disk formatter monitoring seek operation on LUN #1
6. Host adapter #1 I/O subchannel watching for reconnect with disk formatter (SCSI device #3/LUN #1)
7. Host RECEIVE command in progress

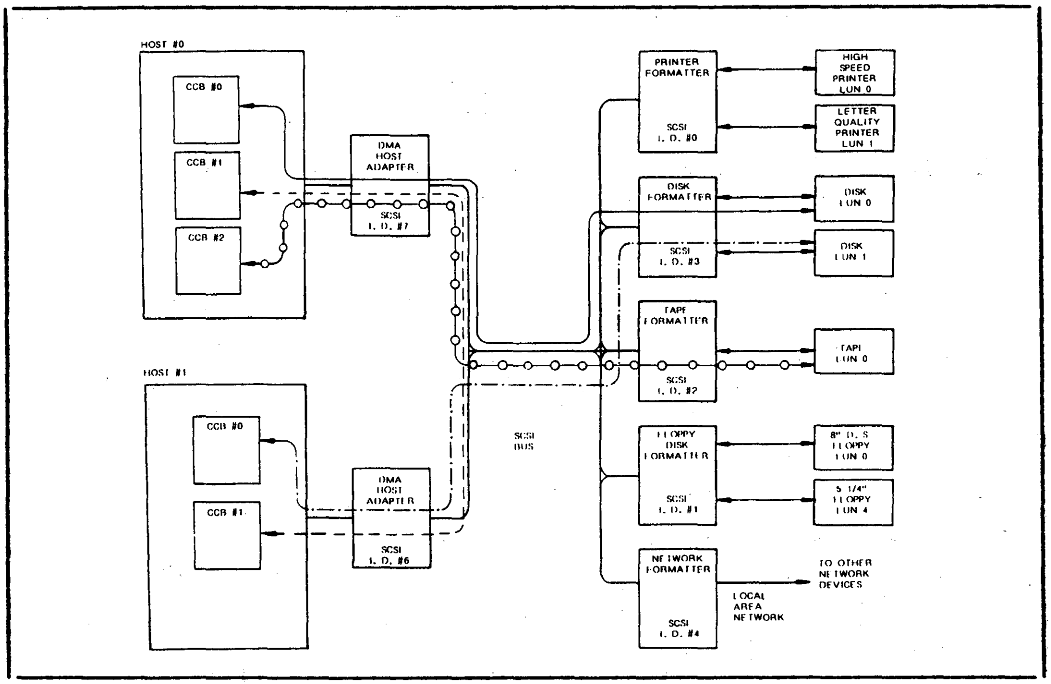


Figure 2-23: System Snapshot t = 46.0ms

SYSTEM PERIOD
t = 46.0ms to 47.0ms

Host adapter #0, the initiator of the host/host communication, receives the 1K byte of data required by the command. After the transfer, host adapter #1 returns a good status into host #0 memory and sends a command complete SCSI level message to host adapter #0. Host adapter #1 then disconnects from the bus and completes the RECEIVE command.

SYSTEM SNAPSHOT
t = 47.0ms

SYSTEM EVENT: Host adapter #0 releases SCSI bus after completion of host/host transfer

SCSI BUS STATUS: BUS FREE

DEVICES WAITING TO ARBITRATE: 1. Disk formatter (SCSI device #3)

EXISTING THREADS:
1. Host #0 CCB #0 to SCSI device #3/LUN #0
2. Host #1 CCB #0 to SCSI device #3/LUN #1
3. Host #0 CCB #2 to SCSI device #2/LUN #0

SCSI DEVICE TASKS IN PROGRESS:
1. Tape formatter repositioning tape.
2. Host adapter #0 I/O subchannel watching for reconnect with tape formatter (SCSI device #2/LUN #0)
3. Disk formatter (SCSI device #3/LUN #0) waiting to transfer data from buffer to host
4. Host adapter #0 I/O subchannel watching for reconnect with disk formatter (SCSI device #3/LUN #0)
5. Disk formatter monitoring seek operation on LUN #1
6. Host adapter #1 I/O subchannel watching for reconnect with disk formatter (SCSI device #3/LUN #1)

SYSTEM PERIOD
t = 47.0ms to 47.5ms

The disk formatter, which lost the last ARBITRATION phase, obtains the bus on this ARBITRATION phase and reselects host adapter #0. The formatter then transfers the 512 bytes of read data from the disk formatter buffer to host #0 memory via host adapter #0's DMA path. After the data has been transferred, the disk formatter returns a good status (again by the DMA path on host adapter #0) and sends a command complete SCSI level message to the host adapter. Following the message transmission, the disk formatter disconnects from the bus with its job complete. The host adapter informs the host that the I/O channel disk READ command is complete.

SYSTEM SNAPSHOT
t = 47.5ms

SYSTEM EVENT: Disk formatter (SCSI device #3/LUN #0) has completed disk READ command and disconnected from bus

SCSI BUS STATUS: BUS FREE

DEVICES WAITING TO ARBITRATE: None

EXISTING THREADS: 1. Host #1 CCB #0 to SCSI device #3/LUN #1
2. Host #0 CCB #2 to SCSI device #2/LUN #0

SCSI DEVICE TASKS IN PROGRESS:
1. Tape formatter repositioning tape.
2. Host adapter #0 I/O subchannel watching for reconnect with tape formatter (SCSI device #2/LUN #0)
3. Disk formatter monitoring seek operation on LUN #1
4. Host adapter #1 I/O subchannel watching for reconnect with disk formatter (SCSI device #3/LUN #1)

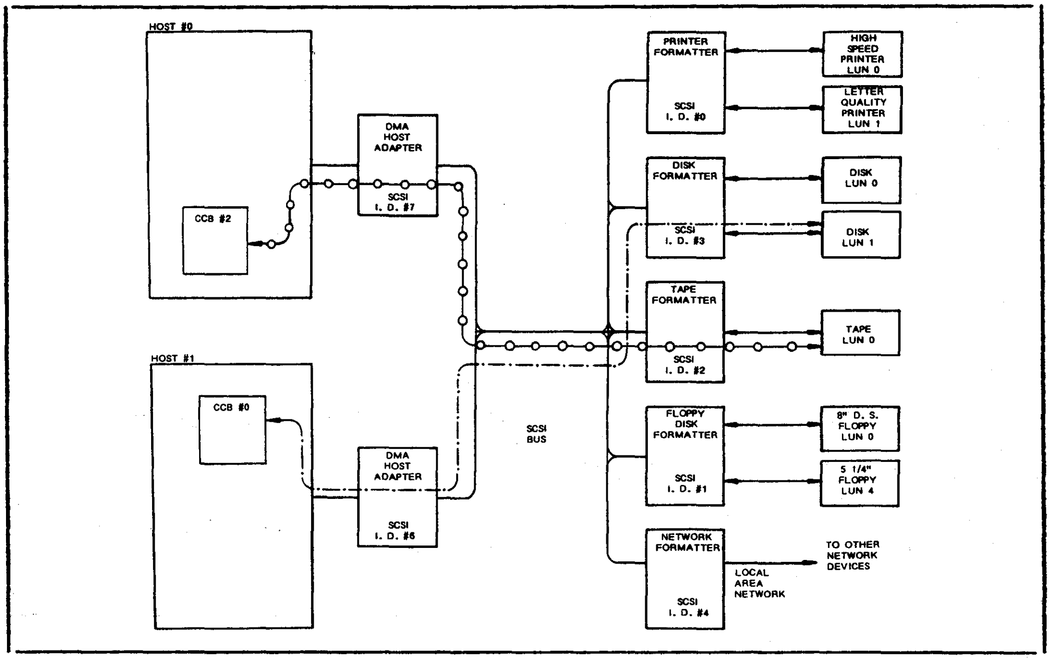


Figure 2-24: System Snapshot t = 47.5ms

SYSTEM PERIOD
47.5ms to 63.5ms

The disk formatter is completing the required seek operation and transferring data to its buffer in order to complete the second disk operation. The tape formatter is continuing to reposition the tape. The SCSI bus is free during the entire period.

SYSTEM PERIOD
63.5ms to 65.5ms

After arbitrating for the bus, the disk formatter reselects host adapter #1 to complete the READ command. The formatter transfers data from the formatter buffer into host #1 memory by the DMA facilities of host adapter #1. After data transfer, ending status is transferred, and the disk formatter disconnects from the bus. Host adapter #1 signals the host on the completion of the I/O channel command.

SYSTEM SNAPSHOT
t = 65.5ms

SYSTEM EVENT: Disk formatter (SCSI device #3/LUN #1) has completed disk READ command and disconnected from bus

SCSI BUS STATUS: BUS FREE

DEVICES WAITING TO ARBITRATE: None

EXISTING THREADS: 1. Host #0 CCB #2 to SCSI device #2/LUN #0

SCSI DEVICE TASKS IN PROGRESS:
1. Tape formatter repositioning tape.
2. Host adapter #0 I/O subchannel watching for reconnect with Tape formatter (SCSI device #2/LUN #0)

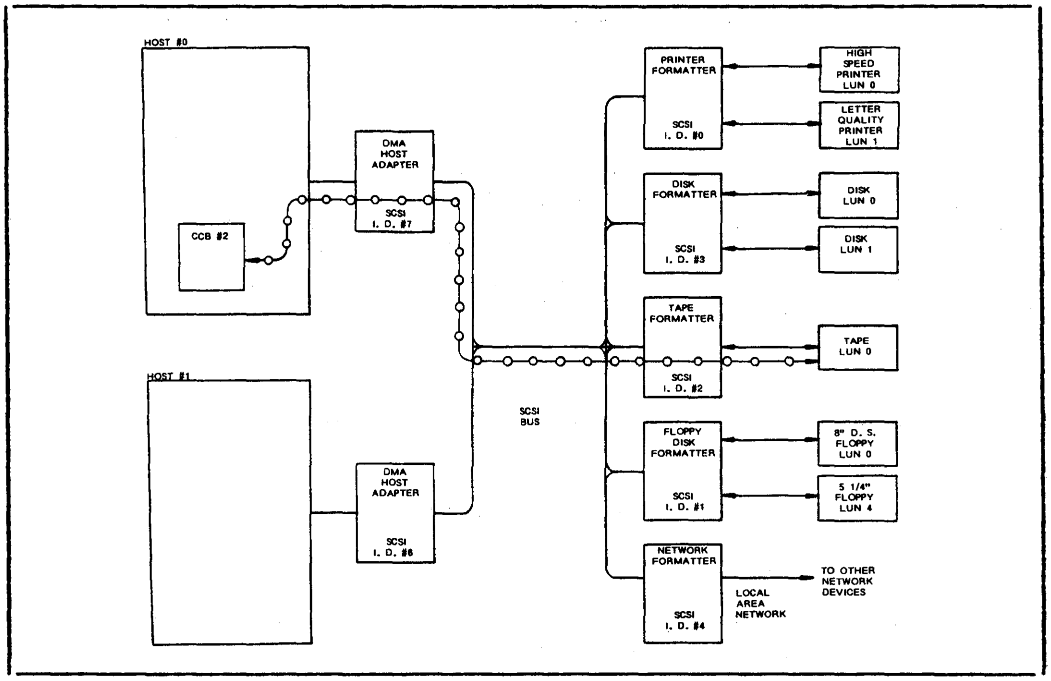


Figure 2-25: System Snapshot t = 65.5ms

SYSTEM PERIOD
t = 65.5ms to 345.5ms

The last open thread on the bus is the tape WRITE command. During this period, the tape formatter is moving the data from its buffer to the tape media.

SYSTEM PERIOD
t = 345.5ms to 346.0ms

Upon completion of the write operation the tape formatter arbitrates for the bus and reselects host adapter #0. The tape formatter returns good completion status to host memory via the host adapter DMA facility. The tape formatter will then indicate command completion and disconnect from the bus. The host adapter signals completion of the I/O Channel Tape Write Command.

SYSTEM SNAPSHOT
t = 346.0

SYSTEM EVENT: Tape formatter disconnects after completion of tape WRITE command

SCSI BUS STATUS: BUS FREE

DEVICES WAITING TO ARBITRATE: None

EXISTING THREADS: None

SCSI DEVICE TASKS IN PROGRESS: None

COMMENTS:
1. System is in an idle state with no I/O commands in progress

---

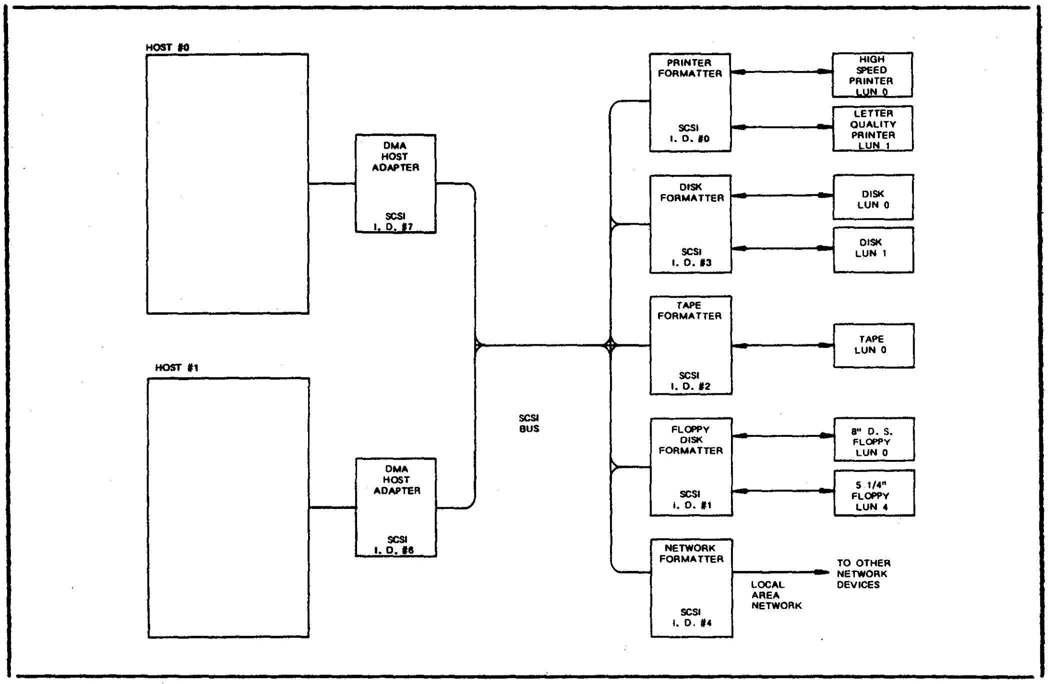

HOST #0
HOST #1
DMA HOST ADAPTER SCSI I. D. #7
DMA HOST ADAPTER SCSI I. D. #6
SCSI BUS
PRINTER FORMATTER SCSI I. D. #0
DISK FORMATTER SCSI I. D. #3
TAPE FORMATTER SCSI I. D. #2
FLOPPY DISK FORMATTER SCSI I. D. #1
NETWORK FORMATTER SCSI I. D. #4
HIGH SPEED PRINTER LUN 0
LETTER QUALITY PRINTER LUN 1
DISK LUN 0
DISK LUN 1
TAPE LUN 0
8" D. S. FLOPPY LUN 0
5 1/4" FLOPPY LUN 4
LOCAL AREA NETWORK
TO OTHER NETWORK DEVICES

Figure 2-26: System Snapshot t = 346.0ms

2.4.3.2 EXAMPLE #2

A major advantage of an I/O bus over a simple interface is the ability to move data between peripherals without involving host memory or the host bus. This example demonstrates such a case.

The example starts at t = 0.0ms. At time t = 1.0ms host #0 issues an I/O COPY command to the tape formatter (I.D. #2 LUN #0). This command directs the tape to read 64K bytes of data from disk formatter #3/LUN #1. At t = 30ms, host #2 issues a disk WRITE command to disk formatter #3/LUN #1.

Figure 2-27 presents a comparison between minimum command execution times and actual execution times on the SCSI bus. For this particular example, the minimum and actual times are identical. This is due to the long period of non-bus activity while the tape is writing from its 32K byte internal data buffer to the host.

Figure 2-28 shows a time line diagram of each of the three devices involved in the example. Loading on the SCSI bus is a low 7.2%. This is due to the block transfer protocol defined in SCSI.

Figure 2-27: I/O Channel Performance Evaluation

COMMAND	MINIMUM TIME FOR EXECUTION	EXECUTION TIME IN EXAMPLE	DIFFERENCE	
			ACTUAL	% INCREASE OVER OPTIMUM
Host #0 Tape COPY	952.5ms	952.5ms	0.0ms	0.0%
Host #0 Disk WRITE	77.5ms	77.5ms	0.0ms	0.0%

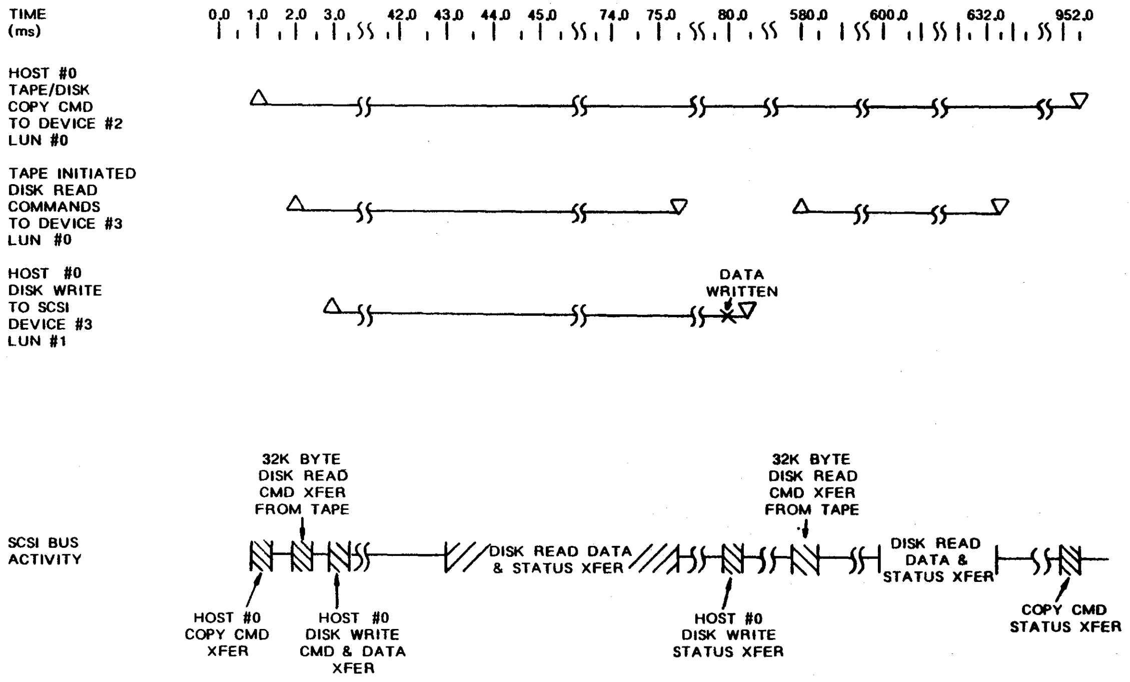

TIME (ms)
0.0
1.0
2.0
3.0
42.0
43.0
44.0
45.0
74.0
75.0
80.0
580.0
600.0
632.0
952.0
HOST #0 TAPE/DISK COPY CMD TO DEVICE #2 LUN #0
TAPE INITIATED DISK READ COMMANDS TO DEVICE #3 LUN #0
HOST #0 DISK WRITE TO SCSI DEVICE #3 LUN #1
DATA WRITTEN
SCSI BUS ACTIVITY
32K BYTE DISK READ CMD XFER FROM TAPE
DISK READ DATA & STATUS XFER
32K BYTE DISK READ CMD XFER FROM TAPE
DISK READ DATA & STATUS XFER
HOST #0 COPY CMD XFER
HOST #0 DISK WRITE CMD & DATA XFER
HOST #0 DISK WRITE STATUS XFER
COPY CMD STATUS XFER

Figure 2-28: Example #2 Time Line Diagram

EXAMPLE #2 DETAIL

This example demonstrates the following SCSI system level features and functions:

- o Peripheral-to-peripheral off line COPY
- o Peripheral acting as an initiator
- o Multi-tasking on disk formatter
- o Host/disk transfer
- o Three concurrent system threads

Figure 2-29 shows the basic system at the start of the example (t=0).

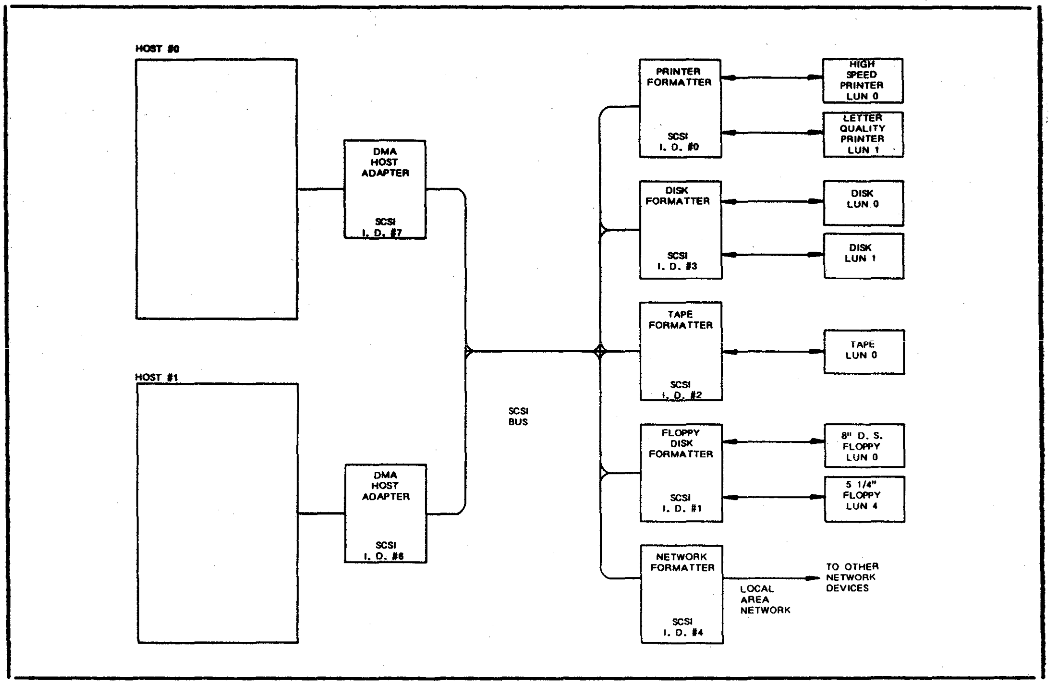

HOST #0
DMA HOST ADAPTER SCSI I. D. #7
HOST #1
DMA HOST ADAPTER SCSI I. D. #6
SCSI BUS
PRINTER FORMATTER SCSI I. D. #0
HIGH SPEED PRINTER LUN 0
LETTER QUALITY PRINTER LUN 1
DISK FORMATTER SCSI I. D. #3
DISK LUN 0
DISK LUN 1
TAPE FORMATTER SCSI I. D. #2
TAPE LUN 0
FLOPPY DISK FORMATTER SCSI I. D. #1
8" D. S. FLOPPY LUN 0
5 1/4" FLOPPY LUN 4
NETWORK FORMATTER SCSI I. D. #4
LOCAL AREA NETWORK
TO OTHER NETWORK DEVICES

Figure 2-29: System Snapshot t = 0.0ms

SYSTEM SNAPSHOT
t = 0.0ms

SYSTEM EVENT: None

SCSI BUS STATUS: BUS FREE

DEVICES WAITING TO ARBITRATE: None

EXISTING THREADS: None

SCSI DEVICE TASKS IN PROGRESS: None

COMMENTS:
1. System is in an idle state with no I/O commands in progress

---

SYSTEM SNAPSHOT
t = 1.0ms

SYSTEM EVENT: I/O channel command issued by host #0
(Host #0 Tape Copy from Disk SCSI device
#3/LUN #0.)

SCSI BUS STATUS: BUS FREE

DEVICES WAITING TO ARBITRATE: None

EXISTING THREADS: 1. Host #0 CCB #0 to host adapter #0

SCSI DEVICE TASKS IN PROGRESS: None

COMMENTS:
1. Host #0 has built command control block #0 and issued an I/O channel command (requesting a COPY by tape SCSI device #2/LUN #0 from disk device #3/LUN #0) establishing a thread between the CCB and host adapter.

SYSTEM PERIOD
t = 1.0ms to t = 1.5ms

SYSTEM ACTIVITY:

After receiving the I/O channel command, the host adapter arbitrates and wins control of the SCSI bus, selects the specified tape formatter, and transfers the command descriptor block to the tape formatter. Once the target (tape formatter) determines the nature of the command, it disconnects from the bus.

SYSTEM SNAPSHOT
t = 1.5ms

SYSTEM EVENT: Disconnect by tape formatter after command transfer

SCSI BUS STATUS: BUS FREE

DEVICES WAITING TO ARBITRATE: None

EXISTING THREADS: 1. Host #0 CCB #0 to SCSI device #2/LUN #0

SCSI DEVICE TASKS IN PROGRESS:
1. Tape formatter decoding COPY command

SYSTEM PERIOD
t = 1.5ms to t = 2.0ms

SYSTEM ACTIVITY:

Having decoded the COPY command, the tape formatter creates a CDB to perform a disk READ on SCSI device #3/LUN #0.

SYSTEM PERIOD
t = 2.0ms to t = 2.5ms

SYSTEM ACTIVITY:

At t = 2.0ms the tape formatter arbitrates for the SCSI and assumes the role of an initiator in requesting a 32K byte disk read from SCSI device #3/LUN #0. Once the command has been transferred, the disk formatter determines that a seek is necessary to obtain the required data. The target disk formatter disconnects from the bus.

Seek, rotational, and buffer fill latencies will total 40.5ms, making data available to the tape formatter at t = 43.0ms.

SYSTEM SNAPSHOT
t = 2.5ms

SYSTEM EVENT: Disconnect by disk formatter after READ command transfer from tape

SCSI BUS STATUS: BUS FREE

DEVICES WAITING TO ARBITRATE: None

EXISTING THREADS:
1. Host #0 CCB #0 to SCSI device #2/LUN #0
2. Tape formatter #0 to SCSI device #3/ LUN #0

SCSI DEVICE TASKS IN PROGRESS:
1. Tape formatter waiting for data from disk
2. Disk formatter starting seek operation on LUN #0

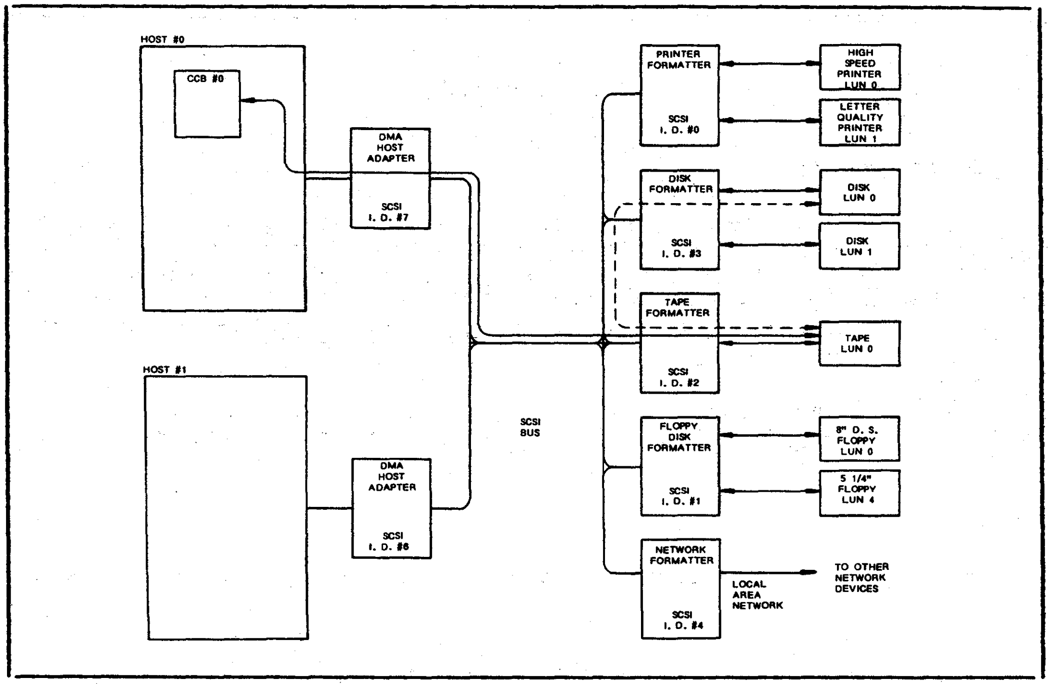


Figure 2-30: System Snapshot t = 2.5ms

SYSTEM SNAPSHOT
t = 3.0ms

SYSTEM EVENT: Host #1 issues I/O channel command (Write a sector to SCSI device #3/LUN #1)

SCSI BUS STATUS: BUS FREE

DEVICES WAITING TO ARBITRATE: None

EXISTING THREADS:
1. Host #0 CCB #0 to SCSI device #2/LUN #0
2. Tape formatter #0 to SCSI device #3/ LUN #0
3. Host #1 CCB #0 to host adapter #1

SCSI DEVICE TASKS IN PROGRESS:
1. Tape formatter waiting for data from disk
2. Disk formatter starting seek operation on LUN #0

SYSTEM PERIOD
t = 3.0ms to t = 4.0ms

SYSTEM ACTIVITY:

Host adapter #0 arbitrates and wins the SCSI bus, selects SCSI device #3/LUN #1 and transfers the CDB. The disk formatter transfers the 512 bytes of data from the host and stores it in its internal buffer. After the data transfer the disk formatter disconnects.

In order to write data onto disk, the formatter must perform a seek operation. The seek, rotational latencies, and buffer write time total 76.0ms.

SYSTEM SNAPSHOT
t = 4.0ms

SYSTEM EVENT: Disconnect by disk formatter after WRITE command and 512 byte data transfer

SCSI BUS STATUS: BUS FREE

DEVICES WAITING TO ARBITRATE: None

EXISTING THREADS:
1. Host #0 CCB #0 to SCSI device #2/LUN #0
2. Tape formatter #3 to SCSI device #3/LUN #0
3. Host #1 CCB #0 to SCSI device #3/LUN #1

SCSI DEVICE TASKS IN PROGRESS:
1. Tape formatter waiting for data from disk
2. Disk formatter waiting for seek operation to complete on LUN #0
3. Disk formatter starting seek operation on LUN #1

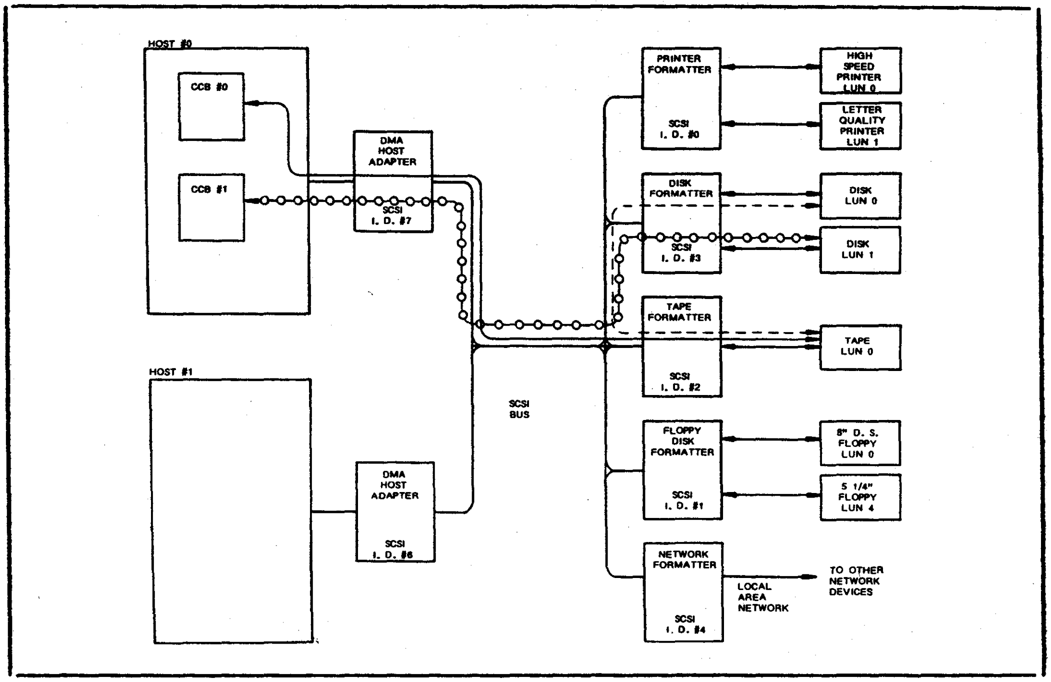


Figure 2-31: System Snapshot t = 4.0ms

SYSTEM PERIOD
t = 43.0ms to t = 75.5ms

SYSTEM ACTIVITY:

At t = 43.0ms the disk formatter reconnects to the tape formatter and begins transfer of the 32K bytes of data requested by the tape formatter. The transfer continues for 16.0ms, at which point the disk formatter transfers ending status for the disk READ command. At this point (t = 75.5ms) the disk releases the SCSI bus, and the disk READ command is completed.

The tape formatter now starts to write the data onto the tape media. Assuming a 200ms ramp up time and 320ms data transfer time, the tape will require additional data at or before t = 595.5ms in order to keep streaming.

SYSTEM SNAPSHOT
t = 75.5ms

SYSTEM EVENT: Disconnect by disk formatter after completion of disk READ operation for tape

SCSI BUS STATUS: BUS FREE

DEVICES WAITING TO ARBITRATE: None

EXISTING THREADS: 1. Host #0 CCB #0 to SCSI device #2 LUN #0
2. Host #0 CCB #1 to SCSI device #3 LUN #1

SCSI DEVICE TASKS IN PROGRESS:
1. Tape formatter writing buffer data onto tape
2. Disk formatter waiting for completion of seek operation on LUN #1

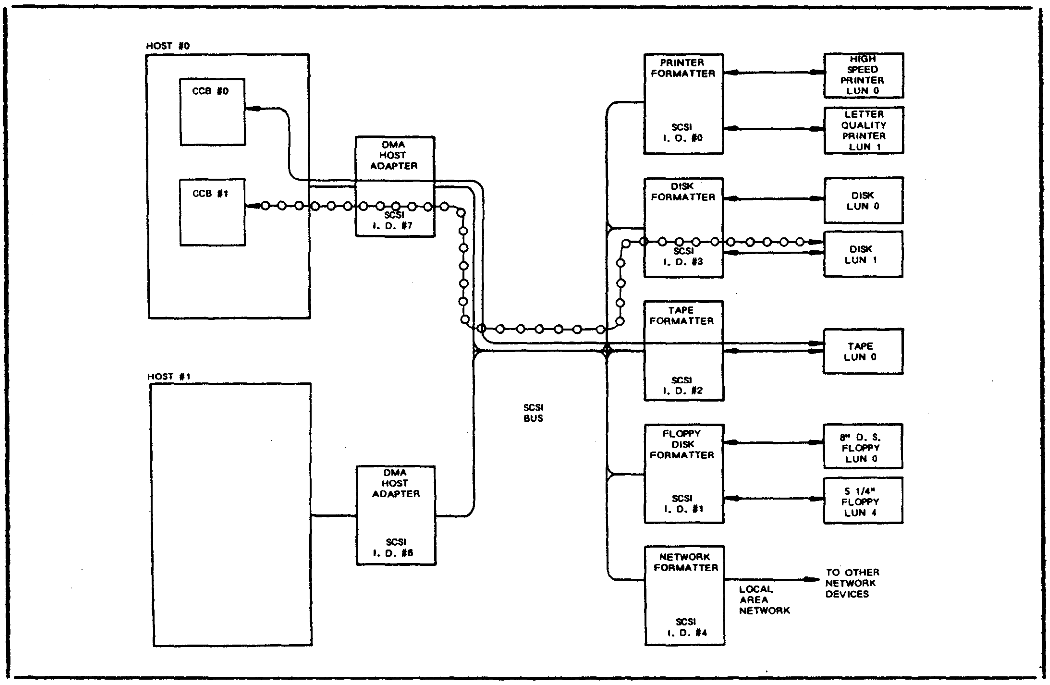


Figure 2-32: System Snapshot t = 75.5ms

SYSTEM PERIOD
t = 80.0ms to t = 80.5ms

SYSTEM ACTIVITY:

After flushing the 512 bytes of host #0 write data, the disk formatter arbitrates for and wins the bus. The device then sends ending status for the disk WRITE command. After the STATUS phase, the disk formatter sends a COMMAND COMPLETE message, then disconnects from the bus. After disconnect, the host adapter informs the host of the I/O sub-channel completion.

SYSTEM SNAPSHOT
t = 80.5ms

SYSTEM EVENT: Disconnect by disk formatter after completion of COMMAND COMPLETE message STATUS phase of host #1 disk WRITE command.

SCSI BUS STATUS: BUS FREE

DEVICES WAITING TO ARBITRATE: None

EXISTING THREADS: 1. Host #0 CCB #0 to SCSI device #2/LUN #0

SCSI DEVICE TASKS IN PROGRESS:
1. Tape formatter writing buffer data onto tape

COMMENTS:
1. The tape formatter requires an additional 32K bytes of disk data in order to complete the COPY command. This data will be obtained by a second disk read.

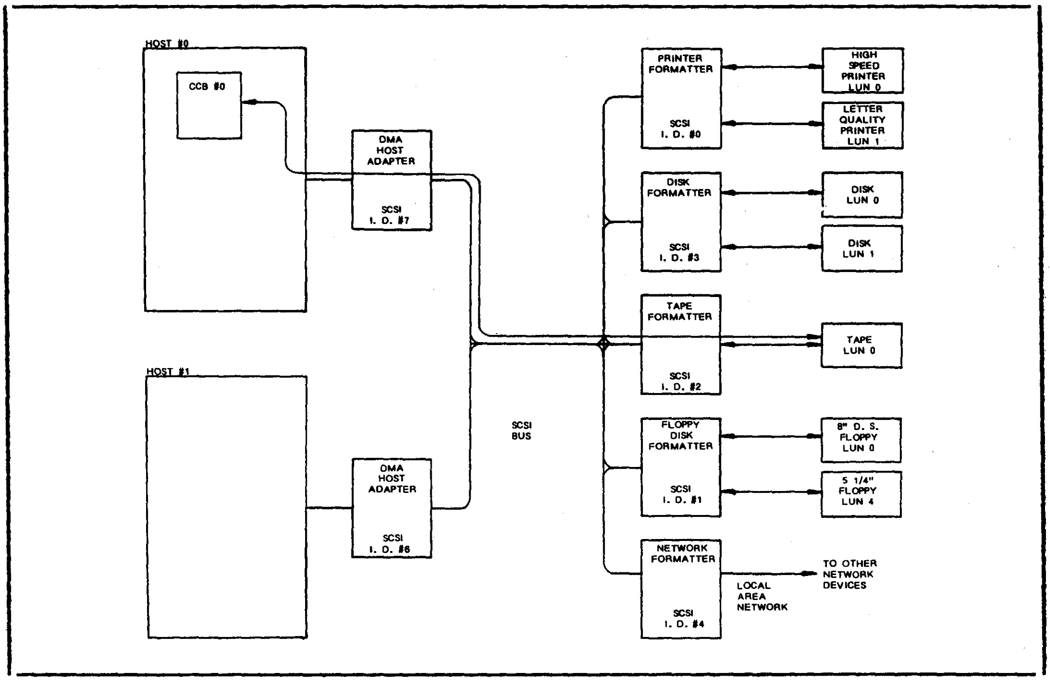


Figure 2-33: System Snapshot t = 80.5ms

SYSTEM PERIOD
t = 80.5ms to t = 580ms

SYSTEM ACTIVITY:

During this period the tape formatter is flushing its buffer onto the non-volatile tape media. The bus is free, and numerous disk operations could be occurring during the flushing process. At t = 580ms the formatter completes the buffer flush and is ready to obtain the second 32K bytes of data from the disk.

SYSTEM PERIOD
t =580.ms to t = 580.5ms

SYSTEM ACTIVITY:

At t = 580.ms the tape formatter acquires the bus and sends a 32K byte READ command to the SCSI device #3, the disk formatter. The target formatter determines there will be a rotational and buffer fill latency of 19.5ms before data can be transferred and disconnects from the bus.

SYSTEM SNAPSHOT
t = 580.5ms

SYSTEM EVENT: Disconnect by disk formatter after transfer of disk READ command by tape formatter

SCSI BUS STATUS: BUS FREE

DEVICES WAITING TO ARBITRATE: None

EXISTING THREADS:
1. Host #0 CCB #0 to SCSI device #2/LUN #0
2. Tape formatter #2 to SCSI device #3/ LUN #0

SCSI DEVICE TASKS IN PROGRESS:
1. Tape formatter waiting for data from disk
2. Disk formatter starting search for data on LUN #0

SYSTEM PERIOD
t =580.5ms to t = 600.0ms

SYSTEM ACTIVITY:

The disk formatter searches for the start of the requested 32K byte block. After filling a portion of its data buffer the formatter reconnects with the tape formatter to transfer data.

SYSTEM PERIOD
t =600.0ms to t = 632.5ms

SYSTEM ACTIVITY:

After reconnection the disk formatter transfers the full 32K bytes of data followed by ending status for the disk READ command. At t = 632.0ms the disk formatter disconnects from the tape formatter. Now the tape formatter has the final data block required to complete its COPY command.

The tape formatter now writes the data in the buffer onto the tape media. This process will take 320ms, if the tape drive did not ramp down after completing the last tape block write.

SYSTEM PERIOD
t =632.5ms to t = 952.0ms

SYSTEM ACTIVITY:

The tape formatter is flushing its 32K bytes of buffer data. Since the formatter does not need to use the SCSI bus, it is free for other operations.

SYSTEM PERIOD
t =952.0ms to t = 952.5ms

SYSTEM ACTIVITY:

The overall COPY command is complete once the tape formatter has finished writing and checking the second 32K byte block onto tape. At this point (t = 952.0) the tape formatter acquires the bus and sends good completion status to host adapter #0. The tape formatter then disconnects from the bus. Upon disconnection the host adapter notifies the host of COPY command completion. The I/O channel has now completed all open commands, and the overall system is back to a no activity status.

3.0 SCSI PERIPHERALS

Any SCSI formatter can be partitioned into two functional blocks - the SCSI front end and the device unique interface - as shown in Figure 3-1. This section deals with the performance characteristics of the SCSI front end, while section 4 discusses the features associated with the device unique interface for each type of SCSI device.

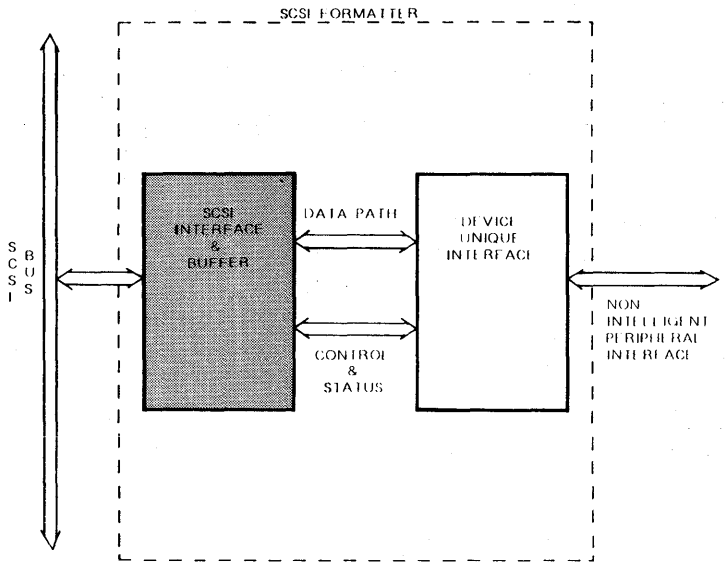


Figure 3-1: SCSI Peripheral Functional Blocks

The SCSI interface contains both hardware and firmware. Hardware functions or characteristics include arbitration, reselection capability, data bus parity, single ended or differential bus interface and data transfer rate. Firmware functions include SCSI message system protocol, SCSI command decode and execution with device unique electronics, SCSI error recovery and multi-tasking.

The amount and type of memory used for the data buffer on a SCSI formatter is a function of cost and available space on the printed circuit board. As device capacities increase and memory costs decrease, the typical capacity of formatter buffers will climb proportionately. Buffer management and interaction with SCSI command and execution protocol is usually a firmware task and is a critical factor in overall throughput.

3.1 HARDWARE FEATURES

3.1.1 ARBITRATION AND RESELECTION CAPABILITY

In order for SCSI to operate as an I/O bus, all targets should be capable of arbitrating for the bus and reselecting an initiator. With these capabilities, SCSI devices should utilize the bus only when they have information to transfer. Older SASI devices without arbitration and reselection capability can be used as targets with arbitrating initiator devices. The penalty for this arrangement is the loss of the bus to other SCSI devices for the entire command execution period. Since the target cannot reselect the initiator, it will hold the SCSI bus busy from selection to ending status transfer. While this does not provide ultimate bus utilization, it does meet the ANSI SCSI specification which classifies arbitration and reselection as options.

3.1.2 DATA BUS PARITY

The parity bit associated with the SCSI data bus is defined as an option by ANSI. Parity provides a means of ensuring data integrity over the SCSI cable. If a system is to use parity, all SCSI devices on the bus must generate odd parity in order to be compatible.

3.1.3 SINGLE / DIFFERENTIAL BUS DRIVE

ANSI has defined two alternatives for the physical characteristics of SCSI bus signals, i.e., the type of bus drivers which can be used. Single ended drivers are cheaper and will be most common. Differential drivers, while costly, have better immunity to cable noise and may offer slightly faster data transfer speeds.

3.1.3.1 SINGLE-ENDED BUS DRIVE

Figure 2-6 shows the pin assignments for the single ended SCSI bus. The ribbon cable connecting the peripherals in a daisy-chained manner is limited to 6.0 meters. All SCSI signals in the cable are terminated at each end by 220/330 Ohm termination resistors, as shown in Figure 3-2.

In general, a single-ended SCSI device must utilize both open collector and active high assertion drivers. The open collector function is necessary for "wired-or" arbitration on the bus. For high data transfer rates, active high assertion provides the fast-rise times. Components which meet the SCSI specification for the single-ended interface include the 7438 (TTL) and 74S240 (STTL) drivers, and 74LS14 (LSTTL) receiver.

3.1.3.2 DIFFERENTIAL BUS DRIVE

The differential bus drive option is recommended for total SCSI cable lengths from 6 meters to 25 meters. As with the single-ended driver, the bus is terminated at both ends. Figure 3-3 shows the pin assignments for the differential alternative. Note that the differential scheme requires a "wired-or" capability during arbitration. This is an uncommon characteristic of differential driver/receivers. Semiconductor vendors have worked closely with SCSI users to provide parts which meet this requirement, such as the 75176 differential transceiver.

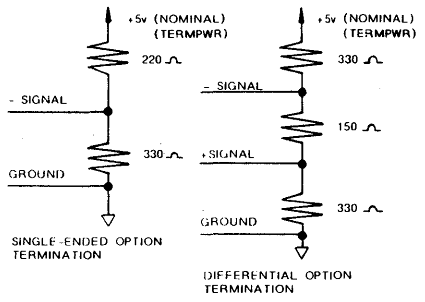


Figure 3-2: SCSI Bus Signal Termination

Figure 3-3: Pin Assignments for Differential Option

SIGNAL	PIN	PIN	SIGNAL
SHIELD GROUND	1	2	GROUND
+DB(0)	3	4	-DB(0)
+DB(1)	5	6	-DB(1)
+DB(2)	7	8	-DB(2)
+DB(3)	9	10	-DB(3)
+DB(4)	11	12	-DB(4)
+DB(5)	13	14	-DB(5)
+DB(6)	15	16	-DB(6)
+DB(7)	17	18	-DB(7)
+DB(P)	19	20	-DB(P)
DIFFSENS	21	22	GROUND
GROUND	23	24	GROUND
TERMPWR	25	26	TERMPWR
GROUND	27	28	GROUND
+ATN	29	30	-ATN
GROUND	31	32	GROUND
+BSY	33	34	-BSY
+ACK	35	36	-ACK
+RST	37	38	-RST
+MSG	39	40	-MSG
+SEL	41	42	-SEL
+C/D	43	44	-C/D
+REQ	45	46	-REQ
+I/O	47	48	-I/O
GROUND	49	50	GROUND

3.1.4 DATA TRANSFER RATE

Efficient use of the SCSI bus requires that a device transfer information as fast as possible once it has acquired the bus. In order to maximize data transfer rates on the SCSI bus, SCSI devices should use DMA type transfers rather than programmed I/O transfers. Typically, programmed I/O transfers range in speed from 50K bytes/second to 300K bytes/second depending on the types of microprocessors commonly used in SCSI devices. DMA transfers typically start at rates exceeding 500K bytes/second and range up to 5 to 10M bytes/second.

The SCSI bus supports two modes of data transfer: asynchronous and synchronous. Figure 3-4 shows the REQUEST/ACKNOWLEDGE timing for asynchronous transfers. Factors limiting throughput for asynchronous transfers are the transmission time associated with the SCSI cable and SCSI device response times.

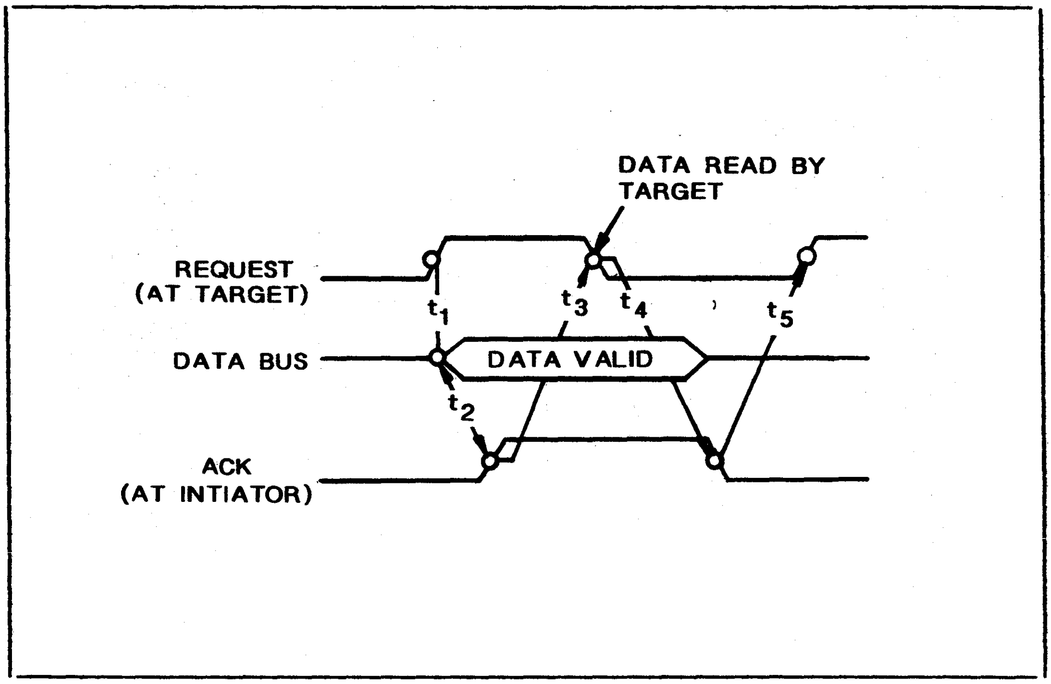


Figure 3-4: REQUEST/ACKNOWLEDGE Timing for Asynchronous Data Transfer

The synchronous data transfer method, sometimes known as "offset-interlock," operates at speeds independent of bus length or cable delay. In this mode the target issues REQUEST strobes synchronously, without waiting for ACKNOWLEDGE responses. Both the initiator and target must be capable of operating at the same synchronous rate, which is determined by an extended message exchange. To ensure the correct number of data bytes have been transferred, the target counts the number of ACKNOWLEDGE strobes generated by the initiator. This is the interlock portion of the transfer. Figure 3-5 illustrates the REQUEST/ACKNOWLEDGE timing for synchronous data transfer.

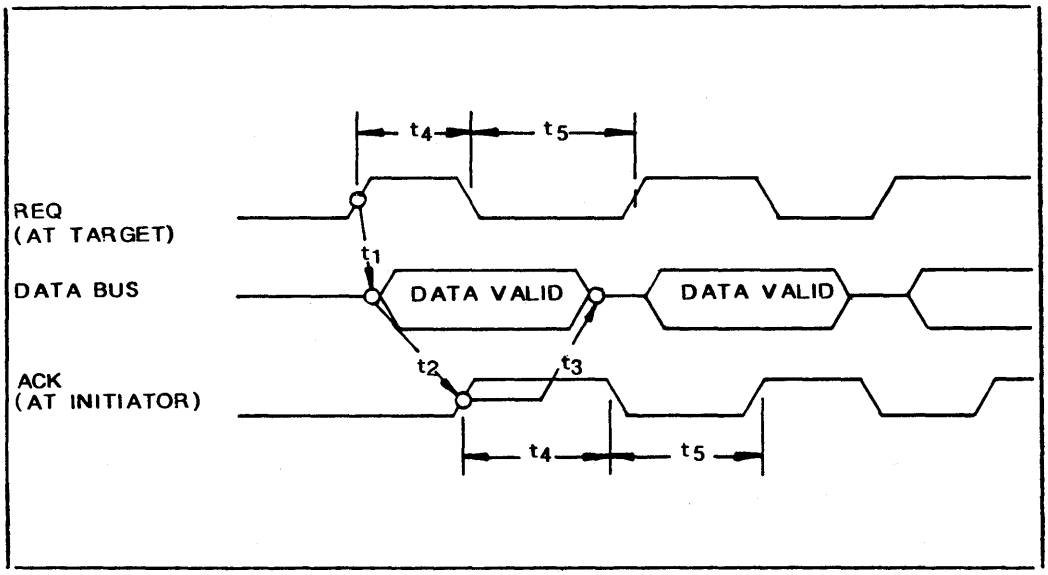


Figure 3-5: REQUEST / ACKNOWLEDGE Timing for Synchronous Data Transfer

3.1.5 FORMATTER BUFFER CAPACITY

The on-board formatter buffer serves many purposes in the overall SCSI environment. In most formatters, the first role of buffer is speed matching between the raw device data rate and the SCSI bus. This speed matching is often bi-directional. For example, a very high performance disk may provide data at 1.2M bytes/second during sector bursts, while the system SCSI host adapter is capable of receiving data at only 900 Kbytes/second. The buffer serves as a speed match and a damper between the disk sector data and the host's DMA sub-channel. In this particular case, the host is not capable of using the bus to the level of performance provided by the peripheral.

An example of the opposite type is a printer peripheral. A moderate size printer, 600 lines/minute, requires an average input data rate of 1320 bytes/second, a data rate 1000 times lower than the SCSI specification. The printer formatter buffer allows a burst transfer of data from the host to the device, thereby efficiently utilizing the SCSI bus. The buffer then supplies data to the printer at its slower data rate, allowing other devices to utilize the SCSI bus.

A second use for the on-board formatter buffer is to accelerate completion of the SCSI command. In this type of application, the formatter returns good completion status when it has all the data in the on-board buffer (for a WRITE command), as opposed to waiting until the data is actually written onto the media. The advantage, particularly for slow devices such as tape drives or printers, is that the host can execute multiple SCSI commands to the device without waiting for the electromechanical latencies.

The disadvantage is that an error may occur in writing data for which good completion status has already been sent. One possible solution, for hosts that wish to use the accelerated completion feature, is to use a command which forces a flush of the buffer before the command is completed. This is the WRITE FILEMARKS command for sequential access devices and the FLUSH BUFFERS command for printer devices. The host issues the command at every point where the host feels the need to ensure that the data has successfully been transferred to the media.

Throughput and size are the two obvious parameters associated with formatter buffers. Typically, larger buffers can provide a larger "shock absorber" to accommodate electromechanical device latencies without interrupting the data flow across the SCSI bus. Large buffers on disk devices can virtually eliminate the need for interleaving, except in unusual real time applications. Buffers up to 16 KBytes are becoming common on SCSI devices, and larger buffers will undoubtedly be available in the future.

Buffer parity generation and checking is a formatter feature which should not be overlooked when evaluating SCSI devices. Since the main task of a peripheral is the correct storage or transmission of data, an undetected error in buffer memory destroys the integrity of the peripheral. Buffer parity is increasingly common on SCSI devices.

3.2 FIRMWARE FEATURES

The SCSI interface features discussed in the following sections are classed as "firmware features," which implies a formatter design approach utilizing a processor under firmware control. Today, the firmware approach is common. However, these features can certainly be implemented in gate arrays or custom IC's.

3.2.1 SCSI MESSAGE SYSTEM

A significant portion of the SCSI specification is dedicated to the message system protocol utilized by SCSI devices. Figure 3-6 shows how the message system fits into the overall SCSI data transfer.

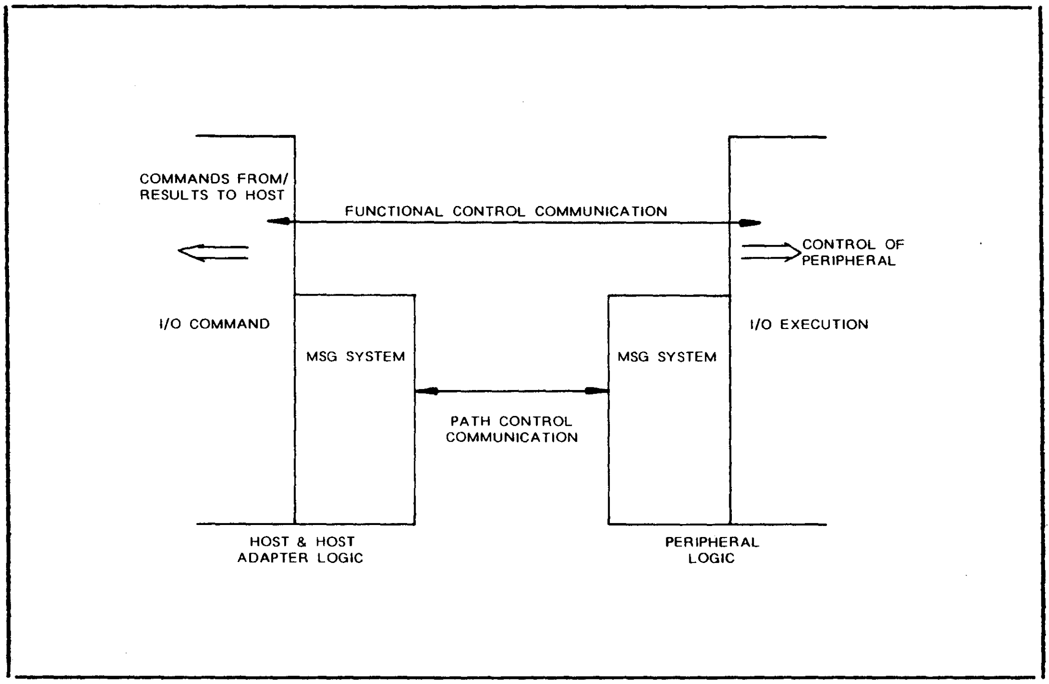

COMMANDS FROM/ RESULTS TO HOST
FUNCTIONAL CONTROL COMMUNICATION
CONTROL OF PERIPHERAL
I/O COMMAND
MSG SYSTEM
MSG SYSTEM
I/O EXECUTION
PATH CONTROL COMMUNICATION
HOST & HOST ADAPTER LOGIC
PERIPHERAL LOGIC

Figure 3-6: SCSI Interface Layers

In Figure 3-6, the major link lies between the I/O command generated in the host/host adapter pair and the I/O execution process performed in the SCSI device. This is a direct logical path from the host command, data, and status blocks to the SCSI device. Outside of this direct logical path is the message system, which supervises the physical SCSI path. The message system is only for communication between SCSI interfaces for management of the physical path and does not directly relate to the host or the non-intelligent peripherals. The effects of an SCSI message may indeed be seen by a host or peripheral but not as a direct communication.

Figure 3-7 summarizes the SCSI message set. Mandatory messages must be implemented in order to be compatible, while the optional messages can be used to obtain a higher degree of overall I/O performance.

For compatibility, the SCSI message system is just as important as the bus signal definition and timing protocol. For high performance, SCSI devices must implement many of the optional messages.

Figure 3-7: SCSI Message Definition (page 1 of 4)

NAME	CODE and DIRECTION*	DEFINITION
COMMAND COMPLETE (Mandatory)	00H In	This message is sent from a target to an initiator to indicate that the execution of a command (or series of linked commands) has terminated and that valid status has been sent to the initiator.
EXTENDED MESSAGE (Optional)	01H I/O	This message is sent from either the initiator or the target as the first byte of a multiple-byte message.
SAVE DATA POINTER (Optional)	02H In	This message is sent from a target to direct the initiator to save a copy of the present active data pointers for the currently attached logical unit.
RESTORE POINTERS (Optional)	03H In	This message is sent from a target to direct the initiator to restore the most recently saved pointers (for the currently attached logical unit) to the active state. Pointers to command, data, and status locations for the logical unit shall be restored to the active pointers. Command and status pointers shall be restored to the beginning of the present command and status areas. The data pointer shall be restored to the value at the beginning of the data area or to the value at the point at which the last SAVE DATA POINTER message occurred for that logical unit.

Figure 3-7: SCSI MESSAGE DEFINITION (page 2 of 4)

NAME	CODE and DIRECTION*	DEFINITION
DISCONNECT (Optional)	04H In	This message is sent from a target to inform an initiator that the present physical path is going to be broken (the target plans to disconnect), but that a later reconnect will be required in order to complete the current operation. This message shall not cause the initiator to save the data pointer. Therefore, if DISCONNECT messages are used to break a long data transfer into two or more shorter transfers, then a SAVE DATA POINTER should be issued before each DISCONNECT message.
INITIATOR DETECTED ERROR (Optional)	05H Out	This message is sent from an initiator to inform the target that an error has occurred that does not preclude the target from retrying the operation. The target may use a RESTORE POINTERS message or perform a disconnect/reconnect sequence to repeat the operation from the last defined state of the pointers.
ABORT (Optional)	06H Out	This message is sent by the initiator to clear the present operation. If a LUN has been identified, all pending data and status for the issuing initiator from that LUN is cleared, and the target will immediately go to the BUS FREE phase. If a LUN has not been identified, the target will simply go to the BUS FREE phase. No status is sent for the operation.
MESSAGE REJECT (Optional)	07H I/O	Sent from either the initiator or target to indicate that the last message it received was inappropriate or has been implemented.

Figure 3-7: SCSI MESSAGE DEFINITION (page 3 of 4)

NAME	CODE and DIRECTION*	DEFINITION
NO OPERATION (Optional)	08H Out	This message is sent from an initiator in response to a target's request for a message when the initiator does not currently have any other valid message to send.
MESSAGE PARITY ERROR (Optional)	09H Out	This message is sent from the initiator to the target to indicate that one or more bytes in the last message it received had a parity error.
LINKED COMMAND COMPLETE (Optional)	0AH In	Sent from a target to an initiator to indicate that the execution of a linked command has completed and that status has been sent. The initiator is then allowed to set up the pointers for the initial state for the next linked command.
LINKED COMMAND COMPLETE WITH FLAG (Optional)	0BH In	Sent from a target to an initiator to indicate that the execution of a linked command (with the FLAG set) has completed and that status has been sent. The initiator is then allowed to set up the pointers for the initial state of the next linked command. Typically the FLAG would be an interrupt in the initiator.
BUS DEVICE RESET (Optional)	0CH Out	This message can be sent from an initiator to direct a target to reset all current I/O operations on that BUS DEVICE (formatter). This message forces the BUS DEVICE to an initial state with no I/O operations pending for an initiator.

Figure 3-7: SCSI MESSAGE DEFINITION (page 4 of 4)

NAME	CODE and DIRECTION*	DEFINITION	
IDENTIFY (Optional)	80H I/O	This message can be sent by either the initiator or target. It is used to establish the physical path connection between initiator and target for a particular LUN.	
		Bit 7	This bit is always set to distinguish this message from other messages
		Bit 6	This bit is only set by the initiator. When it is set indicates that the initiator has the ability to accommodate disconnection and reconnection.
		Bits 5-3	Reserved
		Bits 2-0	These bits specify an LUN address in a target.

* Message direction key:

In - from target to initiator only
Out - from initiator to target only
I/O - In or Out

SCSI has a defined message format for extended messages, and includes three extended messages, all of which are optional. With the MODIFY DATA POINTER message (opcode 00_H), the target can send a signed argument (two's complement) to the initiator and request that the initiator add it to the value of the current data pointer. To set up synchronous data transfers, the initiator and target exchange SYNCHRONOUS DATA TRANSFER REQUEST messages (opcode 01_H) to establish the transfer period and offset.

The EXTENDED IDENTIFY message (opcode 02H), used in conjunction with the normal IDENTIFY message, expands the LUN addressing in a target to up to 256 sub-LUNs within each LUN. With EXTENDED IDENTIFY, the maximum addressing capability is 2048 units on a single target.

3.2.1.1 MESSAGE SYSTEM EXAMPLE: MULTIPLE CYLINDER READ

While Figure 3-7 is definitive, it does not provide a clear understanding of the message system. The following paragraph provides a detailed example of SCSI message system protocol.

In this example the host system requests a multiple block disk read which will cross a cylinder boundary thereby requiring a seek operation on the disk. Figure 3-8 shows a time line diagram of the overall operation. Note the message system activity during the operation.

At t = 0, the host issues an I/O command to the host adapter. The host adapter follows by arbitrating for the bus and selecting the requested SCSI device. During the SELECTION phase the initiator (host adapter) asserts the ATTENTION signal, indicating its desire to send a message to the target after the SELECTION phase is complete. The IDENTIFY message, sent by the initiator as soon as the target puts the bus in the MESSAGE OUT phase, contains the desired LUN on the SCSI device and indicates the initiator's ability to support disconnection.

Once the target has received the message, it proceeds to transfer the CDB from host memory to its internal memory. In this particular example, the target determines that a time-consuming seek will be required so it can release the bus. At t = 0.6ms, the target sends a DISCONNECT message to the initiator and releases the SCSI bus. Upon receipt of the DISCONNECT message, the initiator notes the pending reconnection with the target in order to complete the command.

After the seek is complete and the target has filled its data buffers to a certain level, it arbitrates and wins the SCSI bus. After a RESELECTION phase, the target sends an IDENTIFY message to the initiator to indicate which LUN is reconnecting. The initiator uses this message to reconnect the logical thread between the host memory blocks and the physical peripheral. After transferring a number of data blocks, the target determines that a seek to the next cylinder is necessary to complete the command. The target now sends a SAVE DATA POINTERS message, which causes the initiator to remember the next data location for the coming transfer. Then the target sends a DISCONNECT message, which breaks the physical path, so that the SCSI bus will be free while the target performs a seek.

At this point the initiator data pointer is set for the next byte to be transferred. The data which has already been transferred is considered (parity) error-free and would not be retransmitted if an error occurred in the next data transfer.

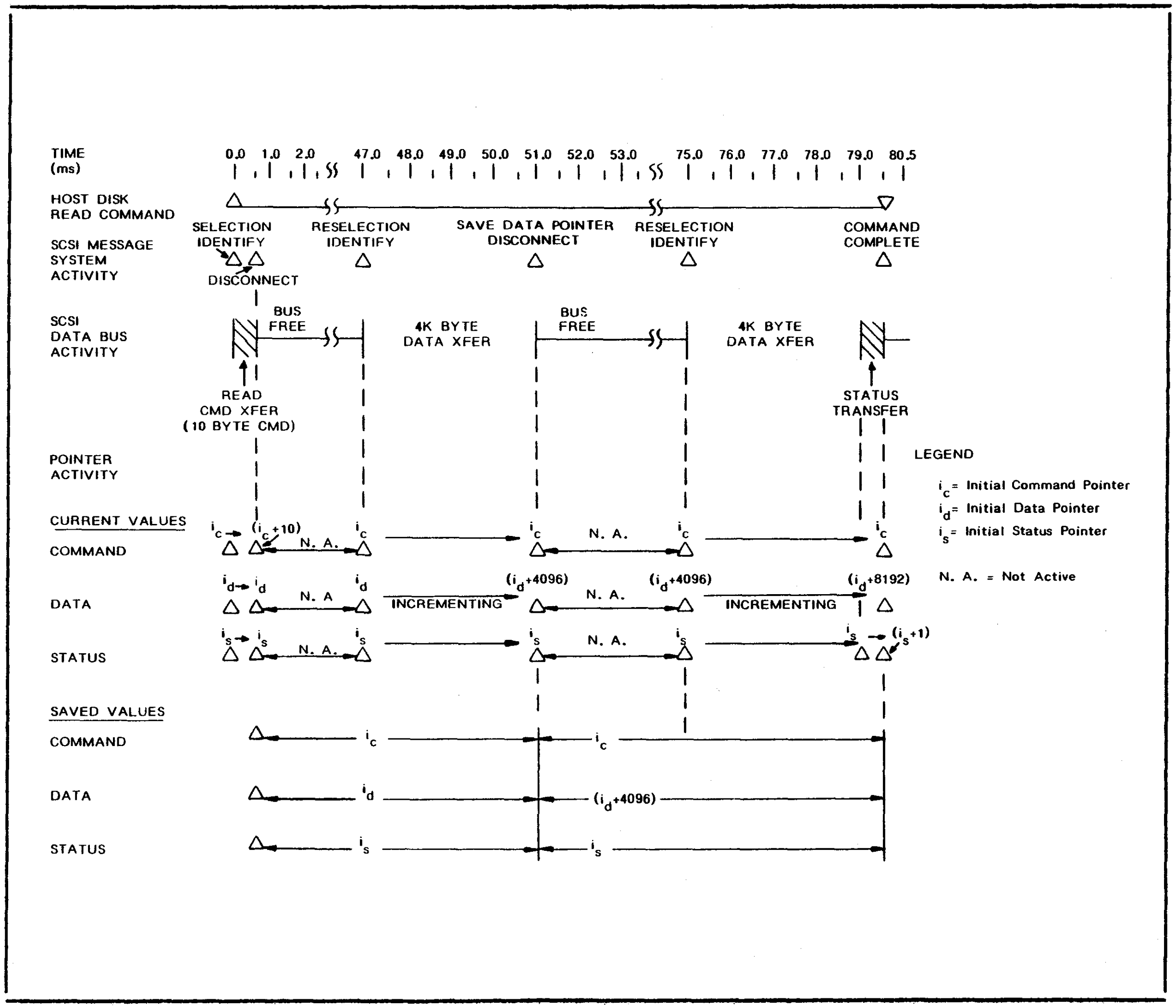


Figure 3-8: Time Line Diagram for Message System Example

After completion of the second seek, the target reconnects to re-establish the physical path, then transfers the remaining data blocks. Again, the target uses the IDENTIFY message in the reconnection process. After the target has transferred all data and placed the return status in the host STATUS block, the target issues a COMMAND COMPLETE message and releases the bus. The initiator (host adapter) responds by informing the host that the I/O sub-channel operation is complete.

3.2.2 SCSI COMMAND DECODE & EXECUTION

Command decode and execution is a middle ground between the SCSI interface function and the device unique interface function. Important to the user are the commands and functions which a particular formatter implements. Of greater importance is the compatibility between formatter command sets on the SCSI bus. For example, a disk/tape transfer may require that the disk unit support the RESERVE DEVICE command and the tape formatter support the COPY command. Details regarding commands for each SCSI device type are summarized in section 4.0, which deals with specific peripheral device types.

3.2.3 SCSI BUS ERROR RECOVERY

For SCSI devices, most of the logic is for the purpose of handling the least frequent events - errors. Error detection and recovery can occur at many points in a SCSI formatter. The area covered in this section deals with errors on the SCSI bus itself.

The major error detection element on the SCSI bus is the bus parity bit. When implemented, this option ensures the integrity of each and every byte transmitted over the SCSI bus. Once a transmission parity error occurs, two techniques are available for recovery. The first and most desirable recovery technique is via the SCSI message system and block pointers. Utilizing SAVE and RESTORE messages, the SCSI I/O system can retry block transmission to recover from a bus parity error.

The second recovery technique is command termination and retry. In this case, once the parity error is detected, the I/O system aborts the I/O command and returns an error condition to the host with command complete. The host must then retry the entire I/O command.

A higher level in the hierarchy of error detection and recovery is the overall command/status and message system protocol defined by SCSI. All SCSI devices must observe certain protocols during each phase of a command; otherwise other devices assume that an error has occurred and take corrective action. Examples of such protocol are:

- o Target transmission of DISCONNECT or COMMAND COMPLETE message prior to SCSI bus disconnection
- o Target verification of parity during SELECTION
- o Target verification of only two bits asserted during SELECTION

3.2.4 MULTI-TASKING

Just as multi-user/multi-tasking computer systems are replacing the single-user, single-task computer, multi-tasking formatters are encroaching on single-task formatters. The SCSI specification defines a protocol and architecture which supports multi-tasking at the formatter level. The examples provided in section 2.0 present real time scenarios in which a SCSI formatter is required to process concurrent commands from different initiators.

Many peripheral applications for SCSI do not require full multi-command execution, but all arbitrating SCSI formatters must be capable of managing the non-intelligent peripheral device, internal data buffers, and the SCSI bus as concurrent tasks.

4.0 SCSI PERIPHERAL FORMATTERS

The following sections discuss each particular type of SCSI device, presenting features or options important to the operation of each device. Each section focuses on the specified SCSI command set for the particular type. However, beyond the actual specification, many issues are left to the discretion of the implementer. The sections document the current industry status of unspecified, discretionary issues and outline possible directions which future developments will take.

This section deals only with the features or characteristics relating to the device unique interface of SCSI devices (see Figure 4-1). Section 3.0 covers the SCSI interface and buffer functions.

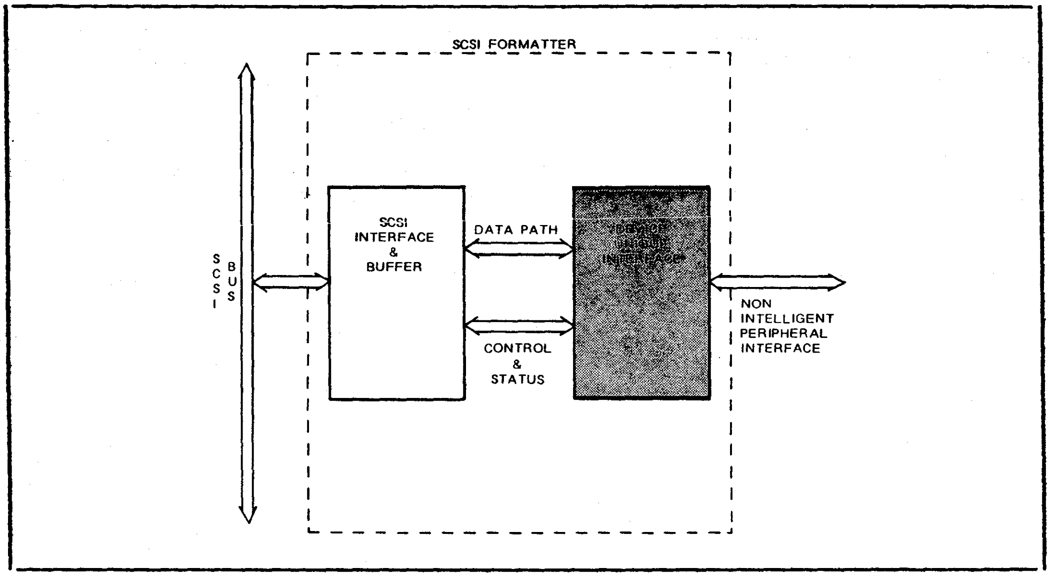
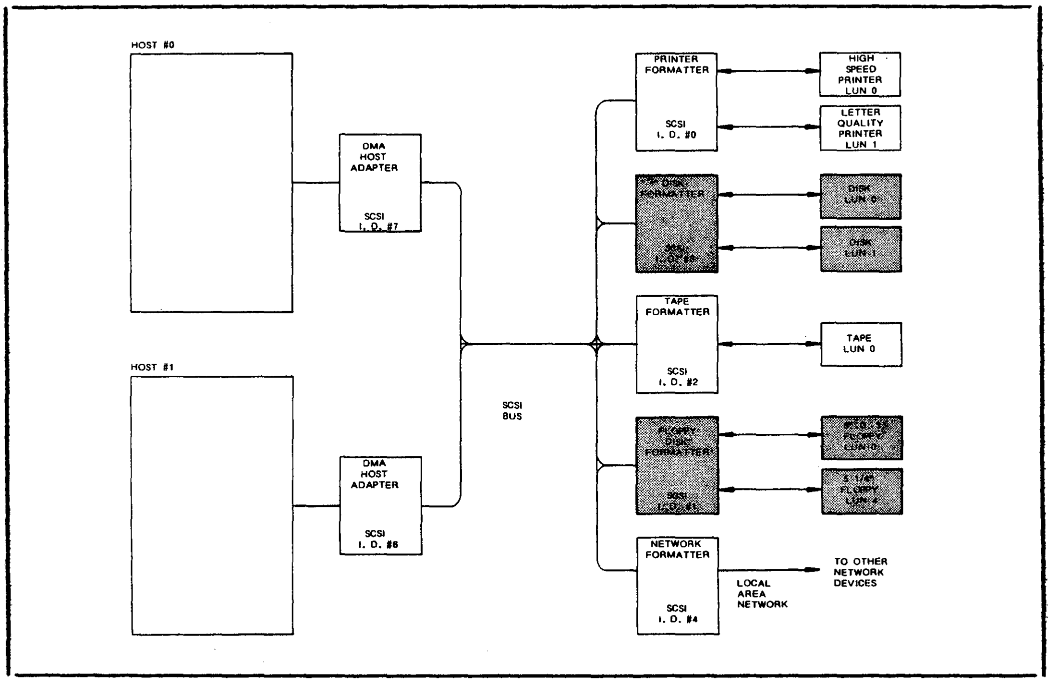


Figure 4-1: SCSI Peripheral Formatter

4.1 HARD AND FLEXIBLE DISK FORMATTERS



The critical peripheral resource in most computer systems is the main random access storage device (usually a rigid disk). This device contains the computer's operating system (OS), application programs, and data bases. With this in mind, the original SASI authors and the current ANSI SCSI committee have invested considerable effort defining a usable and powerful command set for direct access devices.

As shown in Figure 4-2, the disk formatter device unique interface block can be partitioned into two functional areas. The first area is common to all disk formatters regardless of the type of non-intelligent disk drive connected. The second area is the formatter/disk drive interface. Features common to all SCSI disk formatters include the command set, error detection, error correction, defective media handling, sector interleaving, and data throughput capability. Characteristics specific to the formatter/disk interface area include the serial data path format, serial data rate, drive control path, rotational speed, head positioning time, formatting, and handling "grown" defects.

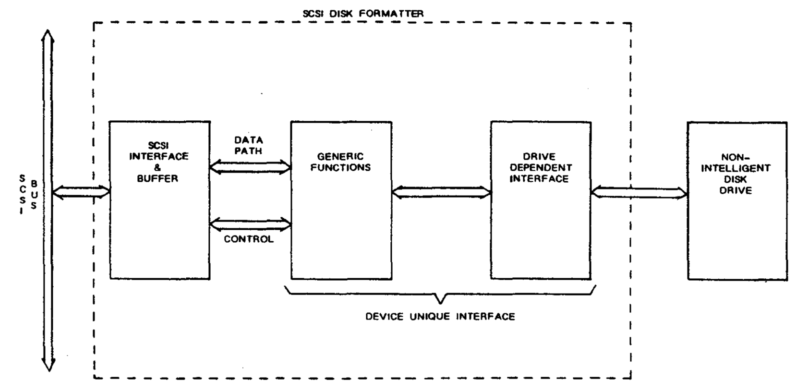


Figure 4-2: Disk Formatter Block Diagram

4.1.1 COMMAND SET

A good starting point for evaluation of any peripheral device is its command set. After all evaluation, the system designer must use the command set to perform the desired system tasks. In fact, an in-depth understanding of the command set can magnify the need for other peripheral features which may seem unimportant at first glance.

Figures 4-3 and 4-4 present tabular summaries of the two groups of commands used with direct access devices. The difference between the two is the number of bytes in the Command Descriptor Block (CDB). Group 0 commands consist of 6 bytes, while Group 1 commands contain 10 bytes. Although READ and WRITE commands in both groups perform the same function, the group 1 (10 byte) commands allow for addressing of 2(exp32) blocks, while the group 0 (6 byte) commands address only 2(exp21) blocks. The expanded addressing instructions will support future peripheral devices with even greater capacities.

Figures 4-3 and 4-4 reference the detailed CDB's found in Appendix A.

Figure 4-3: Group 0 Command Codes for Direct Access Devices

OP CODE	TYPE	DESCRIPTION	CDB DEFINITION
00	O	Test Unit Ready	A.1.1
01	O	Rezero Unit	A.1.2
02	V		
03	M	Request Sense	A.1.3
04	M	Format Unit	A.1.4
05	V		
06	V		
07	O	Re-Assign Blocks	A.1.5
08	M	Read	A.1.6
09	V		
0A	M	Write	A.1.7
0B	O	Seek	A.1.8
0C	V		
0D	V		
0E	V		
0F	V		
10	V		
11	V		
12	E	Inquiry	A.1.9
13	V		
14	V		
15	O	Mode Select	A.1.10
16	O	Reserve Unit	A.1.11
17	O	Release Unit	A.1.12
18	O	Copy	A.1.13
19	V		
1A	O	Mode Sense	A.1.14
1B	O	Start/Stop Unit	A.1.15
1C	O	Receive Diagnostic Results	A.1.16
1D	O	Send Diagnostic	A.1.17
1E	O	Prevent/Allow Medium Rmvl.	A.1.18
1F	R		

TYPE KEY:	M =	Mandatory (minimum implementation for ANSI compliance.)
	E =	Extended (Required for SCSI devices that support device independent self-configuring software.)
	O =	Optional
	R =	Reserved by ANSI for future standardization.
	V =	Reserved for Vendor Unique Definition

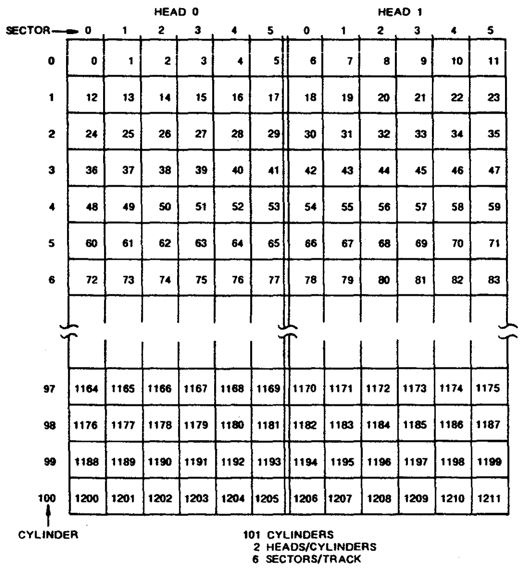
Figure 4-4: Group 1 Command Codes for Direct Access Devices

OP CODE	TYPE	DESCRIPTION	CDB DEFINITION
20	V		
21	V		
22	V		
23	V		
24	V		
25	E	Read Capacity	A.2.1
26	V		
27	V		
28	E	Read	A.2.2
29	V		
2A	E	Write	A.2.3
2B	O	Seek	A.2.4
2C	V		
2D	V		
2E	O	Write And Verify	A.2.5
2F	O	Verify	A.2.6
30	O	Search Data High	A.2.7
31	O	Search Data Equal	A.2.7
32	O	Search Data Low	A.2.7
33	O	Set Limits	A.2.8
34	R		
35	R		
36	R		
37	R		
38	R		
39	O	Compare	A.2.9
3A	O	Copy and Verify	A.2.10
3B	R		
3C	R		
3D	R		
3E	R		
3F	R		

TYPE KEY:		
M	=	Mandatory (minimum implementation for ANSI compliance.)
E	=	Extended (Required for SCSI devices that support device independent self-configuring software.)
O	=	Optional
R	=	Reserved by ANSI for future standardization.
V	=	Reserved for Vendor Unique Definition

The large majority of random access disk devices store information at physical coordinates referenced by cylinder, head and sector. System users must convert from some coordinate system internal to the operating system (OS) to the cylinder, head and sector number on the disk. Since various manufacturers build disks with different numbers of cylinders, heads, and sectors per track, the actual mapping of the OS coordinates onto the disk changes with each new disk. This remapping function is usually handled in device specific I/O driver, which must be rewritten for each new disk. Even if the I/O driver itself is simple to write, maintaining compatibility with units in the field becomes a formidable task.

Since one of the goals of SCSI was to give systems an increased degree of device independence, the SCSI specification defines the mapping function as a formatter task. All direct access device commands reference information in one dimension via a logical block number, which is independent of the physical device configuration. Figure 4-5 shows the relationship between a typical physical disk configuration and the logical block map used by the command set. With logical block addressing, a single I/O driver can be written to handle a number of disk drives, because the SCSI disk formatter handles the device specific logical - physical mapping function.



	HEAD 0						HEAD 1					
SECTOR	0	1	2	3	4	5	0	1	2	3	4	5
0	0	1	2	3	4	5	6	7	8	9	10	11
1	12	13	14	15	16	17	18	19	20	21	22	23
2	24	25	26	27	28	29	30	31	32	33	34	35
3	36	37	38	39	40	41	42	43	44	45	46	47
4	48	49	50	51	52	53	54	55	56	57	58	59
5	60	61	62	63	64	65	66	67	68	69	70	71
6	72	73	74	75	76	77	78	79	80	81	82	83
97	1164	1165	1166	1167	1168	1169	1170	1171	1172	1173	1174	1175
98	1176	1177	1178	1179	1180	1181	1182	1183	1184	1185	1186	1187
99	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199
100	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211

Figure 4-5: SCSI Logical Block Mapping

The following paragraphs discuss selected direct access device commands. Each paragraph includes a brief explanation of the command and comments about its use in a system. Further, notes are included on any common practices or emerging trends among implementers regarding variations or modifications of the command.

INQUIRY COMMAND

In order to provide the operating system with as much information as possible, each type of SCSI device is assigned a unique identification code (See Figure 4-6). This code, accessed via the INQUIRY command, allows the OS to examine all peripherals attached to the SCSI bus and determine the device type during system initialization. The OS then utilizes this information to generate its active resource list during automatic system generation (sysgen).

In addition to the type of device, the INQUIRY command returns a removable medium bit (RMB) which indicates whether the device has fixed or removable media. Some vendors are now incorporating model information about the formatter and direct access device in the vendor unique parameters of the INQUIRY command. With the device model number and internal look up tables, the OS can perform even more sophisticated optimization during sysgen and can select diagnostics appropriate to the device.

Figure 4-6: SCSI DEVICE TYPE I.D. CODES

CODE	SCSI DEVICE CLASS
00	Direct Access Device
01	Sequential Access Device
02	Printer Device
03	Processor Device
04	Write-Once Read-Multiple Device (e.g. Some optical disks)
05	Read-Only Direct Access Device (e.g. Some optical disks)
06-7E	Reserved
7F	Logical Unit not present or implemented
80-FF	Vendor Unique

COPY COMMAND

Many OS functions involve the transfer of large amounts of data from one peripheral device to another without any modification by the host system. This block-move operation can consume valuable CPU time. The SCSI COPY commands allow the CPU to manage large data transfers between peripherals. The actual details of the movement are left to the peripherals, and the CPU only initiates the operation, then checks the operation ending status after the transfer is complete.

Potential users have many concerns about this command, such as its use in file oriented backup operations, its ability to keep 1/4" streaming tape devices running without reposition cycles, and SCSI bus availability for other operations during execution of a COPY comand. Until recently, few formatters were available which supported the COPY command, so the questions have remained largely unanswered. However, with the COPY command now available in both disk and tape formatters, the next few years should see increased experimentation with the COPY command at the systems level. As a result, successful implementations of the COPY command will eventually emerge, and any necessary modifications to formatters will rapidly proliferate across vendors.

COPY AND VERIFY

An extension of the COPY command, COPY AND VERIFY allows the initiator to request that the initiator and target verify the data after it is written. A byte check bit (BytChk) specifies whether the verification is simply a medium verification (ECC, CRC, etc.) or a byte-by-byte comparison.

MODE SELECT and MODE SENSE

The MODE SELECT and MODE SENSE commands were originally designed so that formatters and host systems could communicate media information regarding flexible disk drives. The three basic parameters included in the specification include write protection (WP bit), single- or double-sided media, and single- or double-density recording.

However, flexible disks today represent a much smaller population of SCSI devices than rigid disks. Although the MODE SELECT and MODE SENSE commands were not designed for SCSI rigid disk formatters, they present a tempting avenue for vendors to communicate various vendor unique attributes for SCSI rigid disks and formatters. These attributes include special formatting options, error recovery procedures, and alternatives for handling "grown" defects. With no other satisfactory alternatives available, the MODE SELECT and MODE SENSE commands may become the de facto channel for communicating special attributes on SCSI rigid disks and formatters.

READ CAPACITY

The host must know the capacity of each direct access device in order to configure its disk parameter tables during an automatic sysgen operation, following power-on. With the READ CAPACITY command, the host can request capacity information from the SCSI device in the form of the number of logical blocks available and length (in bytes) of each logical block. With the first logical block referenced as 0, the maximum logical block number which can be addressed is equal to the number of logical blocks - 1.

REASSIGN BLOCKS

Occasionally during operation, some sectors in the disk must be replaced, usually because of a "grown" defect or because of other host system considerations. With the REASSIGN BLOCKS command, the host can request that the SCSI device replace specified logical blocks. However, the host cannot specify the physical location of the replacement logical blocks. See section 4.1.2.3 for a discussion of grown defects.

RESERVE and RELEASE UNIT

In multiple initiator systems, it is necessary to exclude a second initiator from a LUN until the preceding initiator has completed its entire operation, such as a COPY command. SCSI offers initiators the mutual exclusion capability via the RESERVE and RELEASE commands. Once an initiator has executed a RESERVE command on an SCSI device, the reserved LUN will be reported as busy to all other initiators which try to access it. The original initiator gives up the LUN via the RELEASE command.

SEARCH COMMANDS

The data search commands allow the host to specify a data record as an input. The formatter then compares the specified data record with data in the specified range of logical blocks. Some SCSI devices read data into the buffer to do the comparison, while others do the comparison "on the fly" at disk speed (1.2 MBytes/second for large disks). The SEARCH DATA EQUAL command returns good completion if an exact match exists between the data record and disk data. In addition to a good completion code, the command also returns the block number in which the match was found. The SEARCH DATA HIGH and SEARCH DATA LOW commands perform "greater-than" or "less-than" searches.

Performing search functions in formatter hardware can greatly increase application program throughput during data base searches. But because SCSI devices which implement the SEARCH commands have become available only recently, little software exists which actually utilizes them. Only after the SCSI bus becomes available on a wide spectrum of machines will many software developers begin to use these powerful commands.

SEND DIAGNOSTIC and RECEIVE DIAGNOSTIC RESULTS

SCSI formatters should be able to perform a self test and also test the peripheral devices attached to them. While self tests can be very comprehensive, they can also require large amounts of program memory. SCSI devices which cannot afford large amounts of program memory can perform the same type of testing with the diagnostic command set.

The SEND DIAGNOSTIC command allows an initiator to send a diagnostic program to an SCSI device. This program can be a test number, a micro-coded sequence of tests, or actual machine code to be executed by the on-board processor. The RECEIVE DIAGNOSTIC RESULTS command allows the initiator to retrieve the result of the test for evaluation. With these two commands, the ability for diagnostic self test and failure diagnosis can be expanded far beyond the limit of on-board EPROM or ROM space.

One major consideration in the use of this command pair is the necessity for cooperation between SCSI device users and suppliers. For the host system to download microcoded test sequences or actual machine code requires that the system designer have an intimate knowledge of the hardware structure of the SCSI device. It is questionable as to whether SCSI device vendors will release design information at that level. Even if device vendors do release object code for tests, interpretation of the results will still probably require detailed knowledge of the internal architecture.

A second issue is standardization. Because SCSI device suppliers will probably use different internal designs, the test sequences will be different for each device, thus requiring the user to develop and maintain a unique set of test sequences for each supplier's device. This goes against the grain of SCSI's goal to increase commonality and interchangeability of devices.

For the near future, use of the SEND DIAGNOSTIC and RECEIVE DIAGNOSTIC RESULTS commands may be limited to internal testing during manufacturing. But at least these commands offer the opportunity for better testing.

4.1.2 PERFORMANCE CONSIDERATIONS

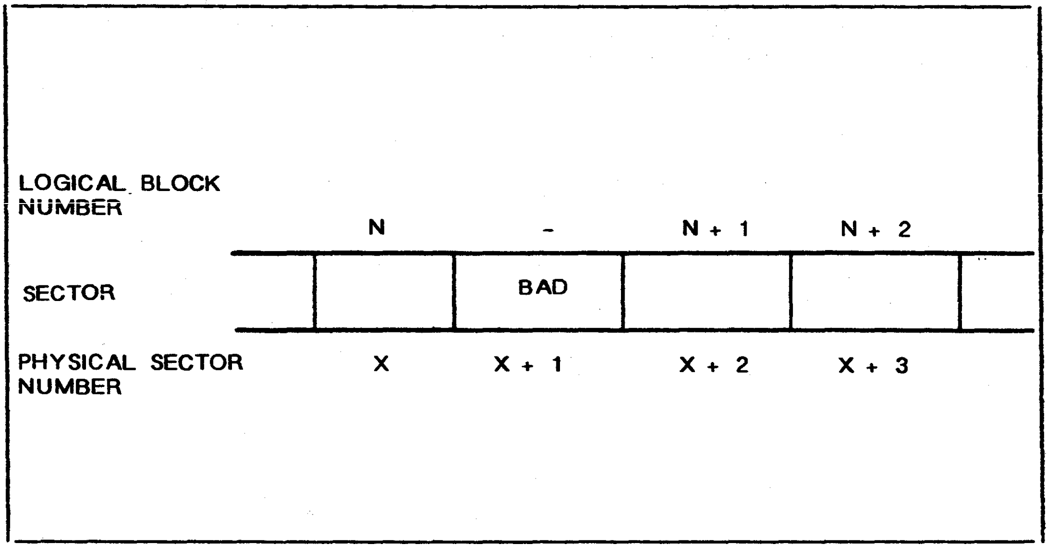
4.1.2.1 FORMATTING AND DEFECTIVE MEDIA MAPPING

Historically, the tasks of formatting the disk and mapping out bad media were functions performed by the host OS. But in order to implement SCSI logical block addressing, the formatter must present the host with a "logically perfect" device. This condition places the responsibility of formatting, defective media mapping, and correction within the disk formatter.

If the system uses single block accesses, then the defective block replacement strategy is not critical. But if the system uses the multi-block accesses envisioned in an SCSI environment, the replacement strategy can cause a significant time penalty. For instance, if the system accesses three consecutive blocks in one read command the middle block is defective and the spare block is on the innermost cylinder, then the additional time for accessing the replacement block is twice the average seek time, plus twice the average rotational latency. This time delay, on the order of 75 milliseconds, is no longer acceptable.

To reduce the access time for replacement blocks, most SCSI vendors have selected a format approach sometimes called "in-line sparing." Figure 4-7 below illustrates in-line sparing. Simply put, during the format operation the formatter replaces any defective block with the subsequent block on the cylinder. With in-line sparing, the time delay for accessing any replacement block is simply the rotational latency for one block, or about 500 microseconds for a 512 byte block and a 1.2 MByte/s disk data rate. No seek is required.

Figure 4-7: "In-Line" Sparing Format Technique



4.1.2.2 ERRORS DURING OPERATION

Since SCSI assumes a logically perfect disk, the formatter must always transmit correct data patterns from the disk, or at least flag uncorrectable data patterns. Error detection has been a formatter function for some time, and error correction has recently emerged as a standard formatter feature. One development forcing improved error detection and correction is the continued increases in the areal density of bits on the media, which makes read errors more likely.

The typical error detection and correction mechanism is an error correction code (ECC), which is simply a long polynomial (32 factors or more) with particular mathematical characteristics. During a write operation, the polynomial is divided into the data pattern and the formatter appends the remainder to the data field. During subsequent read operations, the ECC is redivided into the data field plus the remainder. For correct data, the result is zero. If the result is nonzero, the syndrome indicates the location of the error(s) and the correct bit pattern.

The choices for ECC polynomials are wide and represent an ongoing area for development. But for any ECC polynomial, three key figures of merit are: 1) undetected error probability, 2) uncorrectable error probability, and 3) miscorrected error probability. Of the three events, only uncorrectable errors are acceptable, even if not desirable. Undetected errors and miscorrected errors both imply that the host received incorrect data for processing, which is unacceptable.

The present industry norm for uncorrectable (but detected) bit error probability is 10(exp-13) to 10(exp-15). The norm for the probability of undetected errors and miscorrected errors is 10(exp-15) to 10(exp-18). Undoubtedly, future formatters will continue to improve upon these standards.

4.1.2.3 GROWN DEFECTS

One source of errors during operation is defects in the media which have "grown" since the format operation. The most effective method for removing errors from grown defects is to replace the defective blocks. Since in-line sparing cannot be used without rewriting all the remaining blocks on the disk, many other techniques have been developed for replacing sectors with "grown" defects without disturbing any other sectors. These include spare sectors on each track and spare cylinders located either in the center of the disk, or at the inside of the disk. The tradeoff is access time for replacement blocks versus percent of media required for spares.

With SCSI, determination of grown defects has also emerged as a concern. The formatter now supervises error correction, but the host commands block replacement via the REASSIGN BLOCKS command. The coordination between the host and formatter will certainly see much change and experimentation in the next few years before a standard approach emerges.

4.1.2.4 THROUGHPUT AND SECTOR INTERLEAVING

Historically, many disk applications have involved reading consecutive logical blocks by multiple single block read commands. Since there is processing time overhead between reads in the operating system and formatter, it is not possible to read consecutive logical blocks which are also consecutive physical sectors on the same revolution of the disk. During the processing time, the disk continues to turn and passes over the start of the next sector or sectors. In order to allow reading of consecutive logical blocks in less than a revolution/block, the logical blocks can be interleaved around the disk. The interleave distance (number of physical sectors between two adjacent logical blocks) is the interleave factor. Applications requiring high throughput and single sector reads need a formatter capable of interleaving logical sectors.

With the advent of SCSI, multiple block read commands are beginning to supplant long strings of single-block read commands. Average data throughput is higher, and large buffers accomplish the speed matching function much better than sector interleaving. Further, with a fixed 1:1 interleave and block size, device vendors can significantly improve the percent of media actually available for data.

4.1.3 FORMATTER/DISK INTERFACE

Presently, most SCSI compatible direct access devices are being delivered as a disk with a "native" interface and a compatible SCSI formatter. However, a few disk vendors have announced products with SCSI, instead of a native interface, and the trend is likely to accelerate. Some of the issues discussed in this section are not applicable to an integrated SCSI disk drive, but most of these issues apply to any SCSI direct access device, regardless of how it is packaged.

4.1.3.1 SERIAL DATA PATH

The rigid disk delivers a serial bit stream to the formatter in one of two formats - encoded or decoded. Encoded formats, such as the ST506 interface, provide clock and data information encoded together in a single signal. The encoded format places the encoder/decoder (ENDEC) circuitry on the formatter, which lowers the drive cost. But it requires sending the encoded signal over a cable, which adds additional timing noise. The result is that encoded formats are limited in maximum data rate and in the allowable transition window margins.

Decoded data formats provide two signals from the disk to the formatter. One signal is the clock, while the other is the decoded (NRZ) data. Decoded formats, such as ESDI, place the ENDEC back on the disk, near the read/write channel. In general, decoded formats yield higher data rates and better transition margins. As complete ENDEC IC's become available, the decoded format will predominate.

4.1.3.2 DRIVE CONTROL PATH

Drive control functions include drive selection, head selection and head assembly positioning. Most disk interfaces are designed to support multiple disk drives (four or eight) from the same formatter. This increases total storage capacity without the cost of additional formatters. The initiator selects the desired drive via the LUN field in SCSI commands.

For SCSI devices, the initiator does not specify the head directly. Instead, it specifies logical block number and lets the formatter convert to head/cylinder/sector coordinates. For high throughput systems, head switching time can become a consideration. If the head switch time is more than a few microseconds, then the formatter must skew the logical block numbers of each track during format, precisely enough to accommodate the head switch time without additional delay.

Control over head assembly positioning is typically implemented in one of two ways - step pulses or a cylinder address. Early 5-1/4" drives (ST506 interface) required a direction (IN/OUT) and a number of step pulses, or tracks, to step from the present position. This approach requires the formatter to remember the head position and command all moves, one step at a time, relative to that position. It can be very slow.

Another approach is to transfer the desired cylinder number directly to the drive on a bus. The drive produces a BUSY signal and performs a seek to the requested cylinder. The formatter polls the BUSY line to determine when the seek is complete. With this technique, the drive knows the destination address at the start of the seek, so it can take the optimum seek trajectory.

4.1.3.3 ROTATIONAL SPEED

Rotational speed of the disk platter assembly has a significant effect on overall data access time. If the requested sector has just passed the read/write head, the disk must complete a full revolution before the sector can be accessed. This delay is the rotational latency. Disk drive specifications usually include an average rotational latency value which is one-half the full revolution time. High performance, multi-tasking peripherals utilize the idle time during rotational latencies to perform other tasks on other disks which are at the correct angular position. Formatters which perform "rotational position sensing" (RPS) can estimate the rotational latency and use that time for other functions. This is an important feature in combined rigid/floppy formatters where the floppy rotates at 6 rps and disk moves at 40 to 60 rps.

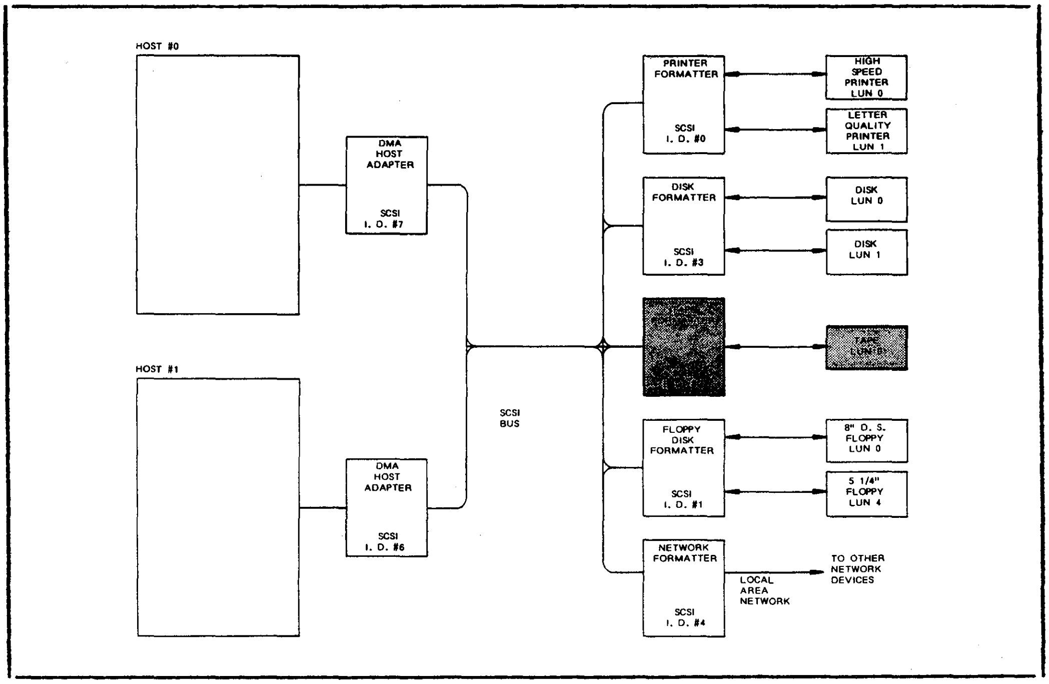
4.1.3.4 HEAD ASSEMBLY POSITIONING TIME

Seek latency is the time required for the head assembly to move from the current cylinder to the requested cylinder. This time is a function of the size and type of head position mechanism used by the disk drive. Average seek times for present 5-1/4" drives are in the 25 - 45 millisecond range, while 8" and 9" disks have seek times in the 15 - 25 millisecond range.

Large capacity disks in multi-user/multi-tasking applications must perform numerous seeks to access the requested blocks. Seek time can become a significant factor in the performance of the overall system. Decreasing seek time directly improves system performance but is becoming more difficult.

Another technique to improve seek times is to use more head positioning assemblies in the system, i.e., more disk drives. The implication is that, for a given system and seek time, there is some optimum data capacity per head positioner; exceeding that optimum number reduces overall data access time. This has been an issue in the mainframe world for some time and may now become a concern in the microcomputer world, if decreasing costs do not render the entire discussion largely academic.

4.2 TAPE FORMATTERS


HOST #0
HOST #1
DMA HOST ADAPTER SCSI I. D. #7
DMA HOST ADAPTER SCSI I. D. #6
SCSI BUS
PRINTER FORMATTER SCSI I. D. #0
HIGH SPEED PRINTER LUN 0
LETTER QUALITY PRINTER LUN 1
DISK FORMATTER SCSI I. D. #3
DISK LUN 0
DISK LUN 1
TAPE LUN 0
FLOPPY DISK FORMATTER SCSI I. D. #1
8" D. S. FLOPPY LUN 0
5 1/4" FLOPPY LUN 4
NETWORK FORMATTER SCSI I. D. #4
LOCAL AREA NETWORK
TO OTHER NETWORK DEVICES

The tape drive is an important peripheral device widely used with the mainframe computers and mini-computers. It serves three distinct functions - backup, archival, and program/data distribution. The IBM 1/2" format (1600 BPI) is the accepted format for data distribution and interchange.

To date, 1/4" cartridge tape drives have found only limited acceptance in the microcomputer marketplace. Possible explanations are the lack of a single accepted data format for interchange, inexperience of microcomputer users with the need for backup, and its inconvenience in use. But tape will eventually be a common microcomputer peripheral, as the obstacles to its use are gradually removed.

The SCSI specification has provided a structure which supports the traditional, 1/2" reel-to-reel tape drives as well as the less costly 1/4" cartridge drives. By easing tape integration and use in systems, as well as helping to standardize tape formats, SCSI may hasten the day when 1/4" cartridge tape becomes a common microcomputer peripheral.

4.2.1 STREAMING VS. START/STOP

The tape drive industry is polarized around two distinct tape recording techniques - streaming and start/stop. Streaming tape drives move the tape continuously, as long as data is available, and do not put any gaps between data blocks. Start/stop devices record data with interrecord gaps large enough for the device to stop and restart without losing any data. Tape drive cost, tape width, block size, and data format all separate neatly according to the recording technique.

The traditional, 1/2" reel-to-reel tape drives utilize the start/stop technique, with variable length block and inter-record gaps where the tape is stopped between each block operation. These machines allow the application program to write data blocks which may vary in length from block to block. The powerful capstan motors utilized to start and stop tape provide milli-second access time to the next record.

Inexpensive, 1/4" cartridge tape drives use the streaming technique. These drives have fixed block lengths and no inter-record gaps. Utilizing streaming techniques, data can be trans-ferred to tape at a 5M byte/minute rate. Streaming tape drives use small drive motors to reduce cost, power consumption, and space. While streaming, these small motors are quite adequate, but because they are small, the start/stop times can exceed 250 milliseconds. As a result, the penalty for small data block transfers is a reposition time of approximately one second or more between data blocks.

Certain 1/2" tape drives combine both start/stop variable length block performance and streaming capability in a single device. These units provide the advantage of both tape formats with minimal disadvantages.

4.2.2 COMMAND SET

Figure 4-8 presents a tabular summary of the group 0 sequen-tial access device command set. Sequential access devices have no unique group 1 commands but do have the powerful COMPARE and COPY AND VERIFY group 1 commands, which are common to all SCSI device types. Detailed CDBs are provided in Appendix A and are referenced in Figure 4-8.

Many of the SCSI protocol commands for sequential access devices, such as INQUIRY, RESERVE, and RELEASE, are identical with those for direct access devices. These commands are discussed in detail in other portions of section 4 and in Appendix A. The following paragraphs deal with commands unique to sequential access devices.

REWIND

Tape, being a sequential media, has both a beginning of tape (BOT) and end of tape (EOT). The REWIND command positions the media to its BOT position.

READ BLOCK LIMITS

Variable block length devices usually have a minimum and maximum block length which can be written or read with a single command. The READ BLOCK LIMITS command allows the host to determine these minimum and maximum values. Fixed block length devices return with the fixed block length as both the minimum and maximum block length.

READ

The READ command contains a single bit which indicates whether the command is a single block read of a variable length block or a multiple block fixed block read. In the first case the length parameter in the command indicates the number of bytes to read. For the second case this parameter indicates the number of consecutive blocks to read. The READ command leaves the tape positioned on the end-of-media (EOM) side of the last block read.

WRITE

The WRITE command is similar to the READ command. The "Fixed" bit determines whether a single block of N byte will be written or N fixed length blocks will be written.

TRACK SELECT

TRACK SELECT allows the initiator to skip over sequential tracks to a specific tape track. This is valuable on devices which can organize sequential records by track and allow the user access to them in this manner.

Figure 4-8: Group 0 Command Codes for Sequential Access Devices

OP CODE	TYPE	DESCRIPTION	CDB DEFINITION
00	O	Test Unit Ready	A.1.1
01	M	Rewind	A.3.1
02	V		
03	M	Request Sense	A.1.3
04	R		
05	E	Read Block Limits	A.3.2
06	V		
07	V		
08	M	Read	A.3.3
09	V		
0A	M	Write	A.3.4
0B	O	Track Select	A.3.5
0C	V		
0D	V		
0E	V		
0F	O	Read Reverse	A.3.6
10	M	Write Filemarks	A.3.7
11	O	Space	A.3.8
12	E	Inquiry	A.1.9
13	O	Verify	A.3.9
14	O	Recover Buffered Data	A.3.10
15	O	Mode Select	A.3.11
16	O	Reserve Unit	A.3.12
17	O	Release Unit	A.3.12
18	O	Copy	A.1.13
19	O	Erase	A.3.13
1A	O	Mode Sense	A.3.14
1B	O	Load/Unload	A.3.15
1C	O	Receive Diagnostic Results	A.1.16
1D	O	Send Diagnostic	A.1.17
1E	O	Prevent/Allow Medium Rmvl.	A.3.16
1F	R		

TYPE KEY: M = Mandatory (minimum implementation for ANSI compliance.)
E = Extended (Required for SCSI devices that support device independent self-configuring software.)
O = Optional
R = Reserved by ANSI for future standardization.
V = Reserved for Vendor Unique Definition

READ REVERSE

Some sequential access devices have the capability to read data in a reverse direction. This feature requires complex formatter electronics which is usually accompanied by high cost. As a system feature, READ REVERSE may have application in large tape based data sort algorithms.

WRITE FILEMARKS

Filemarks are specially coded data blocks used to separate records on sequential media. The WRITE FILEMARKS command allows the host to write a specified number of consecutive file marks with a single command. It also causes the buffers to be flushed to the media.

SPACE

SPACE allows repositioning of the sequential media in both a forward and reverse direction. The initiator specifies a count and a space code. Negative counts (2's complement) imply reverse positioning. The space code can designate blocks, filemarks, sequential filemarks, or physical end-of-data. This comprehensive repositioning capability supplies a limited degree of random access capability to the sequential device.

RECOVER BUFFERED DATA

The RECOVER BUFFERED DATA command is a means of recovering data in the formatter buffer which cannot be written to the tape media but has already been accepted by a WRITE command. It is primarily used in error recovery.

COPY

A third party initiator, usually a host, issues a COPY command to the target sequential access device. The sequential access device, now as initiator, starts a data transfer with the SCSI device specified in the COPY command. When the data transfer is complete, the sequential access device, once again acting as the target of the COPY comand, reports completion status to the COPY command initiator. The purpose is to implement a data transfer independent of the host which initiates it. The file extension format of the command allows complete files to be copied with a single COPY command.

ERASE

The ERASE command simply erases all or part of the media, destroying any data which may have been on the tape.

MODE SELECT AND MODE SENSE

For sequential access devices, the MODE SELECT and MODE SENSE commands have specified values for selecting device speed and data format. Defined data formats for 1/2" tape are 800 BPI (NRZI), 1600 BPI (PE), 3200 BPI (PE), and 6250 BPI (GCR). Defined data formats for 1/4" tape are QIC-11 and QIC-24. As with disk drives, this command pair may become the repository for various vendor unique device options.

LOAD/UNLOAD

The LOAD option requests that the media be positioned at BOT or load point. UNLOAD requests that the media be moved to a position where it can be removed from the tape drive. In the case of 1/2" reel-to-reel tape, UNLOAD will unthread the tape to a single reel.

4.2.3 BUFFERS

As mentioned earlier, streaming 1/4" cartridge tape drives incur long start/stop times as a byproduct of the low cost drive motors used. This limitation does not penalize the streamer when the host provides data continuously, or in large bursts (64 KBytes or more). But when the host transfers short data bursts to the tape, forcing the tape to execute a reposition cycle between each data burst, effective throughput drops dramatically. Figure 4-9 shows the effective throughput of a 90 ips streaming device for various burst, or block, sizes. Note that as the block size decreases, the effective throughput (in Kbytes/second) drops from the maximum of 88.6 to well below 10.

A means to increase effective tape throughput and still maintain start/stop flexibility with the host is via a large FIFO buffer on the tape formatter itself. For example, a formatter with a 32Kbyte buffer would not start tape transfer until the buffer contained at least some predetermined amount of data. This data could either be written to the tape as a single burst or used to give the host a head start at keeping the tape drive streaming. Hosts which could supply data at or above streaming speed could still take advantage of the 88.6K byte/second maximum throughput.

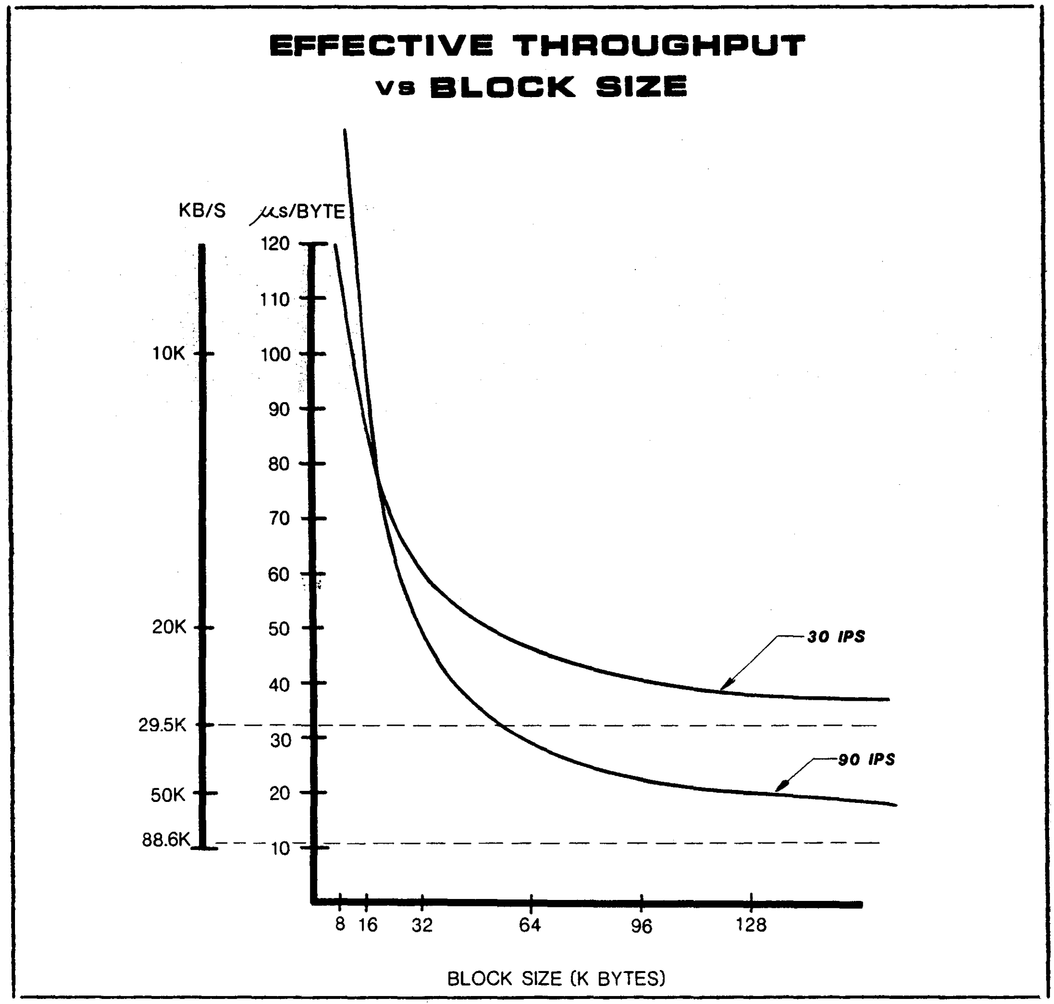
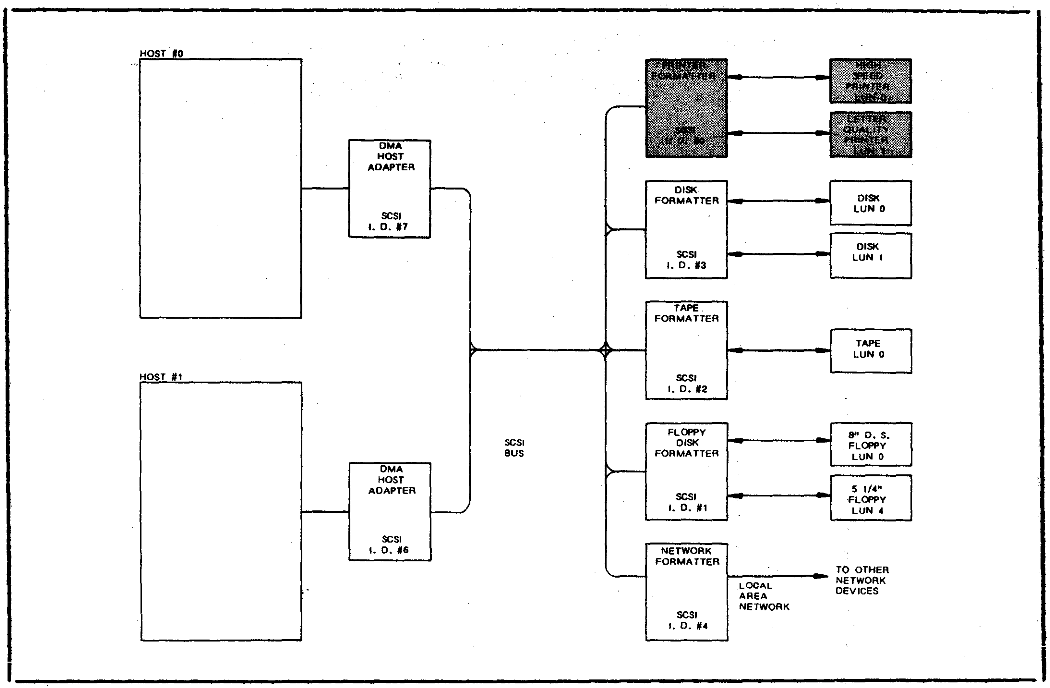


Figure 4-9: Throughput vs. Block Size for Streaming Tape Drives

4.3 PRINTER FORMATTERS


HOST #0
DMA HOST ADAPTER SCSI I. D. #7
HOST #1
DMA HOST ADAPTER SCSI I. D. #6
SCSI BUS
PRINTER FORMATTER SCSI I. D. #0
HIGH SPEED PRINTER LUN 0
LETTER QUALITY PRINTER LUN 1
DISK FORMATTER SCSI I. D. #3
DISK LUN 0
DISK LUN 1
TAPE FORMATTER SCSI I. D. #2
TAPE LUN 0
FLOPPY DISK FORMATTER SCSI I. D. #1
8" D. S. FLOPPY LUN 0
5 1/4" FLOPPY LUN 4
NETWORK FORMATTER SCSI I. D. #4
LOCAL AREA NETWORK
TO OTHER NETWORK DEVICES

Rapid advances in technology and market growth have combined to make high speed printers and laser printers common peripherals for all computers, almost regardless of the computer size. Simultaneously, graphics capabilities of many printers have expanded significantly, thus increasing the data rate to the printer. As a result, it is appropriate to include printers as a distinct type of SCSI device. With the COPY command, despooling print files to a printer can be done across the SCSI bus, off-line from the host. Figure 4-10 shows the group 0 command set for SCSI printer devices. In group 1, printer devices have the common commands - COMPARE and COPY AND VERIFY.

Figure 4-10: Group 0 Command Codes for Printer Devices

OP CODE	TYPE	DESCRIPTION	CDB DEFINITION
00	O	Test Unit Ready	A.1.1
01	V		
02	V		
03	M	Request Sense	A.1.2
04	O	Format	A.4.1
05	V		
06	V		
07	V		
08	V		
09	V		
0A	M	Print	A.4.2
0B	O	Slew and Print	A.4.3
0C	V		
0D	V		
0E	V		
0F	V		
10	O	Flush Buffer	A.4.4
11	V		
12	E	Inquiry	A.1.8
13	V		
14	O	Recover Buffered Data	A.4.5
15	O	Mode Select	A.4.6
16	O	Reserve Unit	A.4.7
17	O	Release Unit	A.4.7
18	O	Copy	A.1.12
19	V		
1A	O	Mode Sense	A.4.8
1B	O	Stop Print	A.4.9
1C	O	Receive Diagnostic Results	A.1.16
1D	O	Send Diagnostic	A.1.17
1E	R		
1F	R		

TYPE KEY: M = Mandatory (minimum implementation for ANSI compliance.)
E = Extended (Required for SCSI devices that support device independent self-configuring software.)
O = Optional
R = Reserved by ANSI for future standardization.
V = Reserved for Vendor Unique Definition

FORMAT

Through the FORMAT command, the initiator can specify forms or fonts to programmable printers. The format information sent is vendor unique, since it is specific to the particular printer.

PRINT

The PRINT command transfers the specified number of bytes from the initiator to the target printer.

SLEW AND PRINT

The SLEW AND PRINT command is similar to the PRINT command, except that it provides form positioning outside the print data. A CHANNEL bit controls the interpretation of the SLEW byte. If the CHANNEL bit is zero, then the SLEW byte indicates the number of lines to advance the form prior to printing. If the CHANNEL bit is one, then the SLEW byte specifies the forms control channel number to which the form shall be advanced prior to printing the data.

FLUSH BUFFER

With the FLUSH BUFFER command. the initiator can force all data to be printed before it releases the printer device.

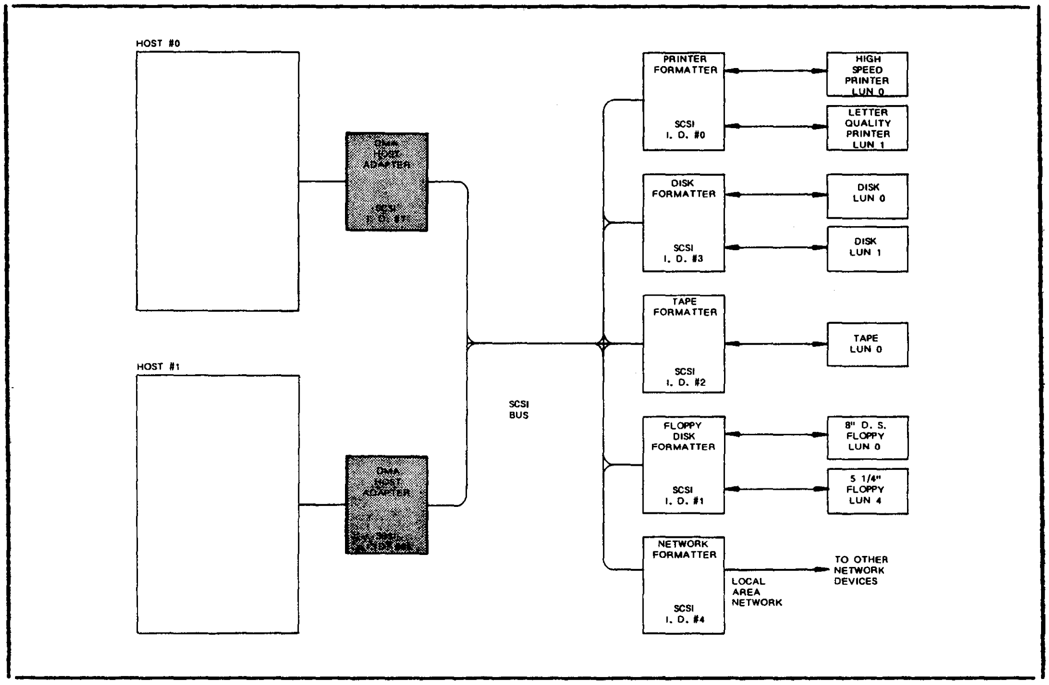
RECOVER BUFFERED DATA

By issuing the RECOVER BUFFERED DATA command, the initiator can retrieve data which has been sent to the target but not yet printed. This command is for use primarily in error recovery.

STOP PRINT

The STOP PRINT command provides a method to terminate printing buffered data in an orderly fashion. The STOP PRINT command includes a RETAIN bit which determines the fate of data remaining in the target buffer. If the RETAIN bit is zero, the data is discarded; otherwise, it is saved. The actual point at which printing is stopped is device specific.

4.4 PROCESSOR DEVICES AND HOST ADAPTERS



The capability and power of the SCSI bus is of no use to a host computer without a host bus adapter (HBA) to convert SCSI signals and protocol to information meaningful to the host. In addition to acting as an initiator, a host adapter can be the target of a host-to-host communication. This activity utilizes the SCSI bus as a communication medium.

4.4.1 HOST ADAPTER FUNCTIONS AND FEATURES

The SCSI architecture requires a means to communicate with host memory. This access is necessary to read CDB's, transfer data and write Return Status Blocks. The host bus adapter (HBA) performs these functions. HBA architectures range from simple latches, driven by the host CPU, to complete I/O subsystem managers with on-board processor, DMA control for the host bus, and sizable data buffer. As a rule of thumb, the simpler the HBA, the more host intervention is necessary to manage the SCSI bus.

An additional feature of a higher performance host adapter is the ability to act as an incoming message receiver. It can take on the role of a target on the SCSI bus.

Performance features to consider when evaluating SCSI host adapters include SCSI bus arbitration capability, ability to act as a target, buffer capacity, SCSI bus utilization, host memory transfer speed and host involvement in SCSI protocol operations. In a high performance SCSI I/O subsystem, effective utilization of the SCSI bus resource is a requirement. Since the host adapter is also connected to the host memory, efficient use of the SCSI bus will reflect in efficient use of the host bus. Important in overall SCSI I/O performance is the amount of host involvement supporting the SCSI protocol.

4.4.2 HOST ADAPTER IMPLEMENTATIONS

SCSI host adapters can be implemented in a variety of ways. Hardware component counts can range from as few as six chips for a simple non-intelligent implementation, to 60 or more for a full I/O channel implementation. SCSI users should closely match the host adapter implementation to their particular performance needs and cost goals.

4.4.2.1 SIX CHIP HOST ADAPTER

Many potential SCSI users have been led to believe that an SCSI HBA requires only six chips. In fact, an SCSI HBA can be built with as few as six chips (as shown in Figure 4-11), but its performance is very limited. This HBA will not support arbitration and reconnection. Further, it uses a slow poll loop method to transfer each byte of command, data or status information. (Typical transfer rates range from 20K bytes/second to 50k bytes/second.) The host computer must provide all SCSI protocol support. If the peripherals attached to the SCSI bus do not implement many of the SCSI messages, the host level of support can be reduced, but a certain minimum level is unavoidable. This type of HBA is generally suited to single-user/single task computers, with only one device attached to the SCSI bus.

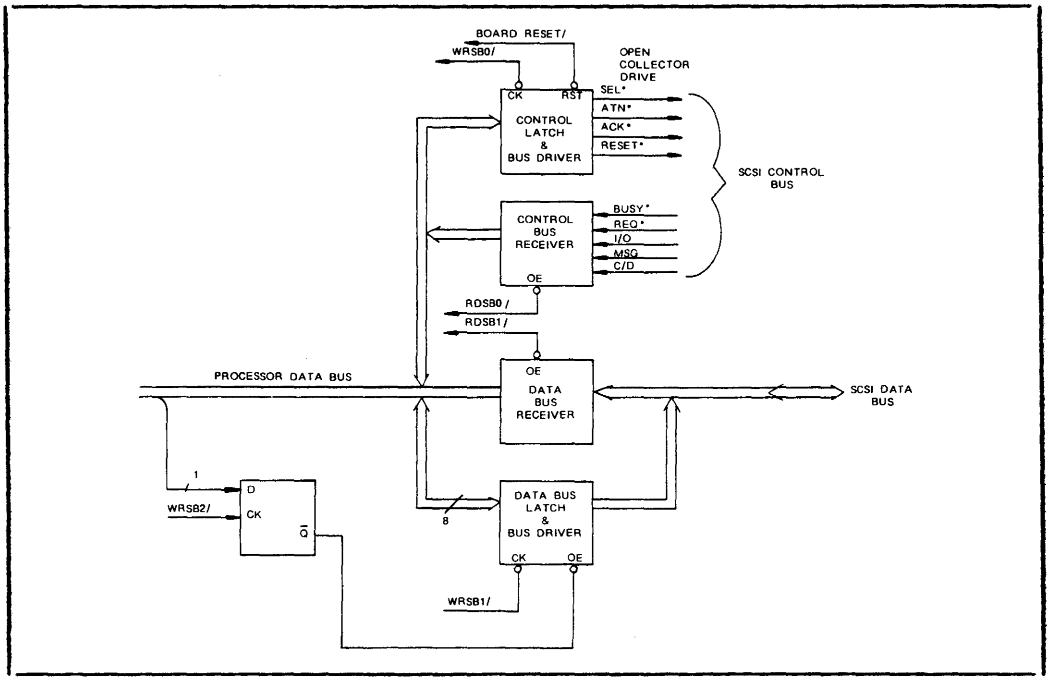


Figure 4-11: Six Chip SCSI Host Adapter Block Diagram

4.4.2.2 PROGRAMMED I/O WITH TRANSMIT/RECEIVE MACHINE

An implementation between the "6 Chip" and full I/O channel is a programmed I/O version which uses a finite state machine to control the SCSI REQUEST/ACKNOWLEDGE handshake. Figure 4-12 shows the block diagram of such a device. The CPU may utilize block I/O move commands with this architecture. The host adapter synchronizes the CPU with the SCSI bus handshake via the PWAIT (processor wait line). If data is not available for input when the processor enters its input cycle, it is placed in a wait state until the SCSI target places data onto the bus. Usually a watch-dog-timer accompanies the T/R machine to guarantee release of PWAIT in the event of a system problem. Data rates obtained from this scheme range from 150K bytes/second up to 400K bytes/second. The adapter shown in Figure 4-12 will also perform arbitration allowing full SCSI disconnect/reconnect. As was the case with the six chip HBA, the host system must handle all of the SCSI protocol logic.

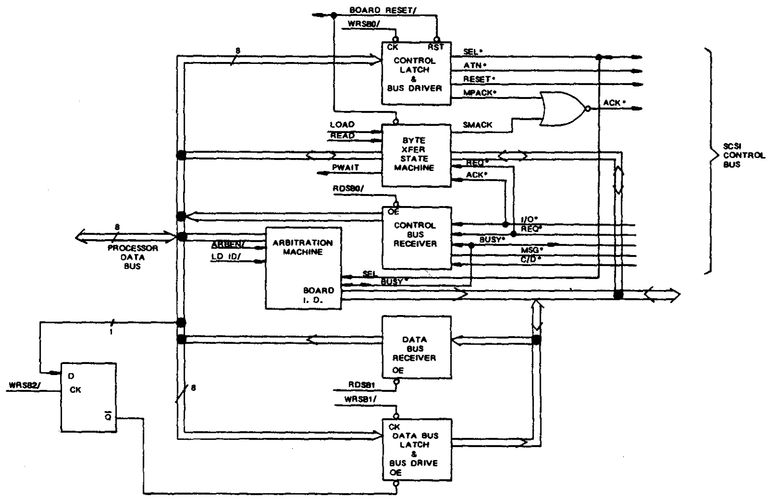


Figure 4-12: Programmed I/O Host Adapter with T/R Machine

4.4.2.3 I/O CHANNEL HOST ADAPTER

The full I/O channel is by far the most sophisticated HBA architecture. Because it handles the data transfer and SCSI bus management tasks, it greatly reduces the host tasks. As shown in Figure 4-13, the I/O channel contains an onboard processor which controls SCSI protocol and the SCSI bus. Moving these functions out of host software into the HBA frees the host from any supervisory responsibilities for the SCSI bus. The host's role is reduced to generating memory based CDB's, informing the host adapter of their locations and checking status upon command completion. The I/O channel's DMA path into memory allows the channel to read the CDB, transfer data and write ending status directly into host memory.

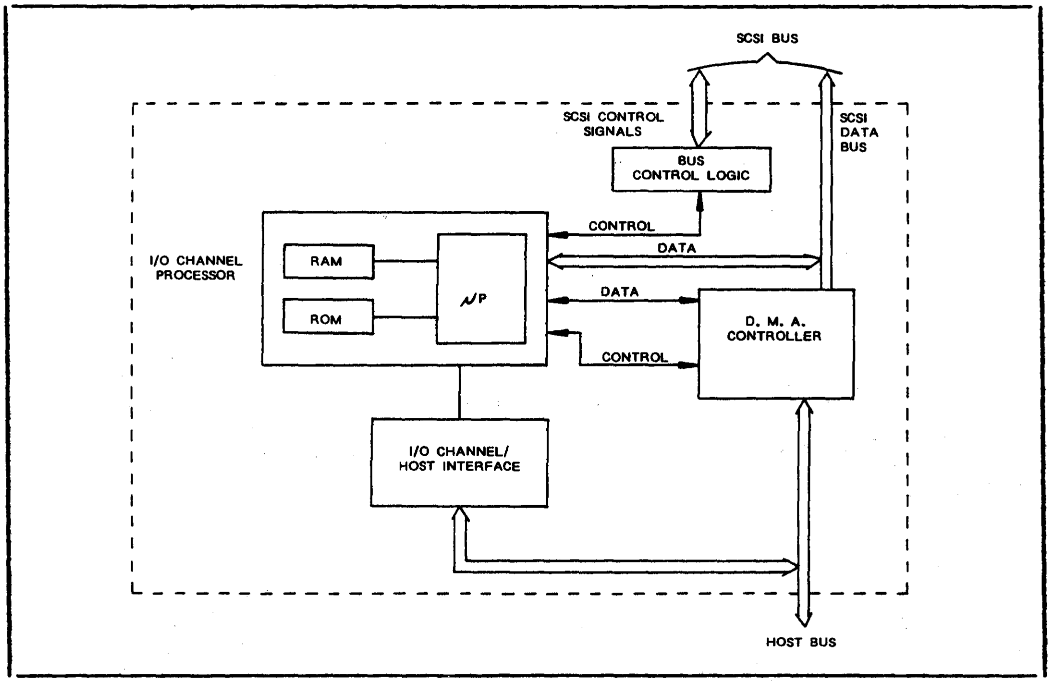


Figure 4-13: I/O Channel Block Diagram

Data rates on the I/O channel range from 900K bytes/second up to 4M bytes/second. These speeds provide rapid movement of large amounts of data on the SCSI bus. Complex SCSI systems should utilize an I/O channel concept in order to minimize SCSI bus loading.

Coupled closely with DMA transfer rate is the host tolerance for a single DMA channel to monopolize the host bus. Some host systems limit the period of time a DMA channel can hold the host bus. In systems with this restriction, the I/O channel must acquire the host bus, transfer a burst of data, relinquish the bus, then repeat the cycle until the full SCSI transfer is complete.

I/O CHANNEL INTERFACE

A major goal of SCSI is to shorten new peripheral integration time. This objective is achieved by the common command set for all devices of the same type and the block addressing architecture of SCSI peripherals. These features also help assure commonality and interchangeability of SCSI devices. In order to have true second sourcing and software interchangeability at the I/O function level, the I/O channel interface to the host bus must be common across all devices of the same type. The only variables should be the I/O channel protocol and the I/O channel port address.

Since the I/O channel is tied very closely to the HBA and factors internal to the system, ANSI has limited the scope of the SCSI specification to the areas of the SCSI bus, CDB definition, and pointer definition. The definition of host adapter configurations has been left to the individual manufacturers and the marketplace.

4.4.3 PROCESSOR DEVICE COMMAND SET

Figure 4-14 presents the command set for SCSI processor devices as defined in the REV 4 B of the SCSI specification. All the group 0 commands for processors except SEND and RECEIVE are common to all SCSI device classes. The group 1 commands for processor devices are the common ones - COMPARE and COPY AND VERIFY.

RECEIVE COMMAND

The RECEIVE command requests the target to transfer to the initiator the number of bytes specified in the CDB.

SEND COMMAND

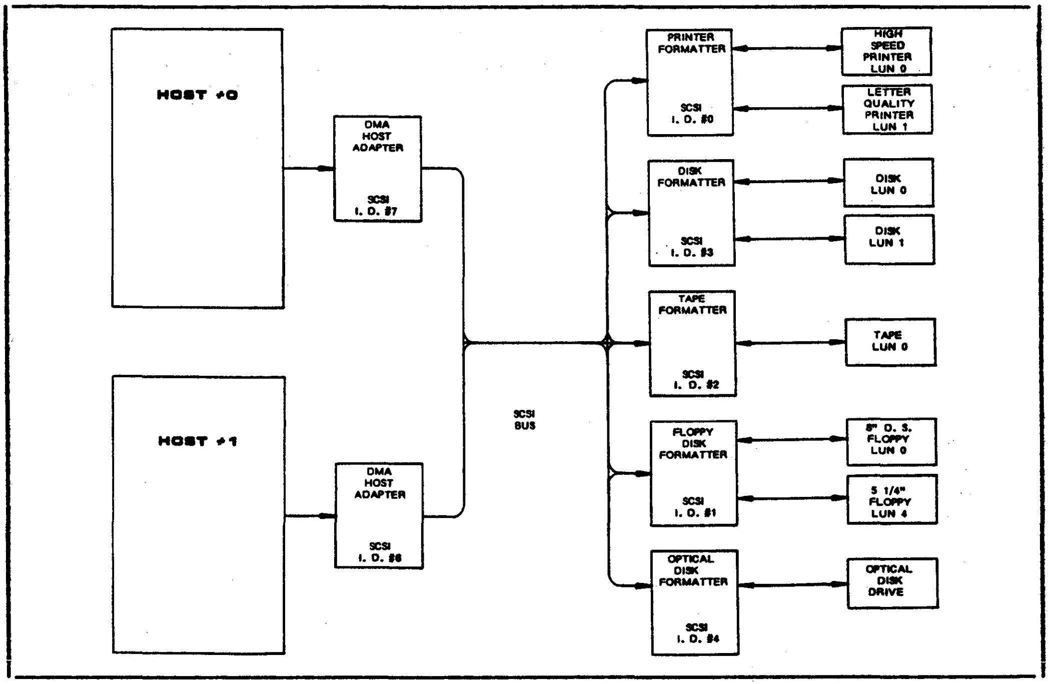
With the SEND command, the initiator transmits the number of bytes specified in the CDB to the target.

Figure 4-14: Group 0 Command Codes for Processor Devices

OP CODE	TYPE	DESCRIPTION	CDB DEFINITION
00	O	Test Unit Ready	A.1.1
01	V		
02	V		
03	M	Request Sense	A.1.3
04	V		
05	V		
06	V		
07	V		
08	O	Receive	A.4.1
09	V		
0A	M	Send	A.4.2
0B	V		
0C	V		
0D	V		
0E	V		
0F	V		
10	V		
11	V		
12	E	Inquiry	A.1.9
13	V		
14	V		
15	V		
16	V		
17	V		
18	O	Copy	A.1.13
19	R		
1A	R		
1B	R		
1C	O	Receive Diagnostic Results	A.1.16
1D	O	Send Diagnostic	A.1.17
1E	R		
1F	R		

TYPE KEY: M = Mandatory (minimum implementation for ANSI compliance.)
E = Extended (Required for SCSI devices that support device independent self-configuring software.)
O = Optional
R = Reserved by ANSI for future standardization.
V = Reserved for Vendor Unique Definition

4.5 WRITE-ONCE READ-MULTIPLE AND READ-ONLY FORMATTERS


HOST #0
HOST #1
DMA HOST ADAPTER SCSI I. D. #7
DMA HOST ADAPTER SCSI I. D. #6
SCSI BUS
PRINTER FORMATTER SCSI I. D. #0
HIGH SPEED PRINTER LUN 0
LETTER QUALITY PRINTER LUN 1
DISK FORMATTER SCSI I. D. #3
DISK LUN 0
DISK LUN 1
TAPE FORMATTER SCSI I. D. #2
TAPE LUN 0
FLOPPY DISK FORMATTER SCSI I. D. #1
8" D. S. FLOPPY LUN 0
5 1/4" FLOPPY LUN 4
OPTICAL DISK FORMATTER SCSI I. D. #4
OPTICAL DISK DRIVE

4.5.1 WRITE-ONCE READ-MULTIPLE DEVICES

Optical recording technology has created a new type of storage device - the write-once read-multiple (WORM) disk. These devices may be ideal for program and data distribution, as well as for applications where permanent records of every transaction are necessary. However, the development cycle has been long for these devices, and they have yet to establish a clear niche in the storage market.

If WORM devices do secure a market niche, SCSI will be ready with a command set specifically for them. Figure 4-15 shows the group 0 commands for WORM devices, and Figure 4-16 shows the group 1 commands for them. This command set is identical to the command set for direct address devices, with the exception that the group 0 READ and WRITE commands, mandatory for direct access devices, are optional for WORM devices, and the reverse is true for the group 1 READ and WRITE commands. Also, some options and attributes defined for direct access devices, such as MODE SELECT and MODE SENSE options, are left reserved, or vendor unique, for WORM devices.

Figure 4-15: Group 0 Command Codes for Write-Once Read-Multiple Devices

OP CODE	TYPE	DESCRIPTION	CDB DEFINITION
00	O	Test Unit Ready	A.1.1
01	O	Rezero Unit	A.1.2
02	V		
03	M	Request Sense	A.1.3
04	R		
05	V		
06	V		
07	O	Re-Assign Blocks	A.1.4
08	O	Read	A.6.1
09	V		
0A	O	Write	A.6.2
0B	O	Seek	A.1.7
0C	V		
0D	V		
0E	V		
0F	V		
10	V		
11	V		
12	E	Inquiry	A.1.8
13	V		
14	V		
15	O	Mode Select	A.6.3
16	O	Reserve	A.1.10
17	O	Release	A.1.11
18	O	Copy	A.1.12
19	V		
1A	O	Mode Sense	A.6.4
1B	O	Start/Stop Unit	A.1.14
1C	O	Receive Diagnostic Results	A.1.16
1D	O	Send Diagnostic	A.1.17
1E	O	Prevent/Allow Medium Rmvl.	
1F	R		

TYPE KEY: M = Mandatory (minimum implementation for ANSI compliance.)
E = Extended (Required for SCSI devices that support device independent self-configuring software.)
O = Optional
R = Reserved by ANSI for future standardization.
V = Reserved for Vendor Unique Definition

Figure 4-16: Group 1 Command Codes for
Write-Once Read-Multiple Devices

OP CODE	TYPE	DESCRIPTION	CDB DEFINITION
20	V		
21	V		
22	V		
23	V		
24	V		
25	E	Read Capacity	A.2.1
26	V		
27	V		
28	M	Read	A.7.1
29	V		
2A	M	Write	A.7.2
2B	O	Seek	
2C	V		
2D	V		
2E	O	Write And Verify	A.7.3
2F	O	Verify	A.7.4
30	O	Search Data High	A.2.7
31	O	Search Data Equal	A.2.7
32	O	Search Data Low	A.2.7
33	O	Set Limits	
34	R		
35	R		
36	R		
37	R		
38	R		
39	O	Compare	A.2.8
3A	O	Copy and Verify	A.2.9
3B	R		
3C	R		
3D	R		
3E	R		
3F	R		

TYPE KEY:		
M	=	Mandatory (minimum implementation for ANSI compliance.)
E	=	Extended (Required for SCSI devices that support device independent self-configuring software.)
O	=	Optional
R	=	Reserved by ANSI for future standardization.
V	=	Reserved for Vendor Unique Definition

4.5.2 READ-ONLY DIRECT ACCESS DEVICES

Another variation of optical disk is a read-only device. As with WORM devices, the market niche and customer acceptance are, as yet, unproven. Figures 4-17 and 4-18 show the group 0 and group 1 commands, respectively, for read-only direct access devices. They are identical to the commands for WORM devices, with the exception that the WRITE and REASSIGN BLOCKS commands have been deleted, and some status information is changed.

Figure 4-17: Group 0 Command Codes for Read-Only Direct Access Devices

OP CODE	TYPE	DESCRIPTION	CDB DEFINITION
00	O	Test Unit Ready	A.1.1
01	O	Rezero Unit	A.1.2
02	V		
03	M	Request Sense	A.1.3
04	R		A.1.4
05	V		
06	V		
07	R		A.1.15
08	O	Read	A.1.5
09	V		
0A	R		A.1.6
0B	O	Seek	A.1.7
0C	V		
0D	V		
0E	V		
0F	V		
10	V		
11	V		
12	E	Inquiry	A.1.8
13	V		
14	V		
15	O	Mode Select	A.1.9
16	O	Reserve	A.1.10
17	O	Release	A.1.11
18	O	Copy	A.1.12
19	V		
1A	O	Mode Sense	A.1.13
1B	O	Start/Stop Unit	A.1.14
1C	O	Receive Diagnostic Results	A.1.16
1D	O	Send Diagnostic	A.1.17
1E	O	Prevent/Allow Medium Rmvl.	
1F	R		

TYPE KEY: M = Mandatory (minimum implementation for ANSI compliance.)
E = Extended (Required for SCSI devices that support device independent self-configuring software.)
O = Optional
R = Reserved by ANSI for future standardization.
V = Reserved for Vendor Unique Definition

Figure 4-18: Group 1 Command Codes for Read-Only Direct Access Devices

OP CODE	TYPE	DESCRIPTION	CDB DEFINITION
20	V		
21	V		
22	V		
23	V		
24	V		
25	E	Read Capacity	A.2.1
26	V		
27	V		
28	M	Read	A.2.2
29	V		
2A	R		A.2.3
2B	O	Seek	
2C	V		
2D	V		
2E	R		A.2.4
2F	O	Verify	A.2.5
30	O	Search Data High	A.2.7
31	O	Search Data Equal	A.2.6
32	O	Search Data Low	A.2.8
33	O	Set Limits	
34	R		
35	R		
36	R		
37	R		
38	R		
39	O	Compare	
3A	O	Copy and Compare	
3B	R		
3C	R		
3D	R		
3E	R		
3F	R		

TYPE KEY:

M =	Mandatory (minimum implementation for ANSI compliance.)
E =	Extended (Required for SCSI devices that support device independent self-configuring software.)
O =	Optional
R =	Reserved by ANSI for future standardization.
V =	Reserved for Vendor Unique Definition

APPENDIX A

COMMAND DESCRIPTOR BLOCK DEFINITION

The command descriptor blocks (CDBs) presented in this appendix are taken directly from the ANSI document X3T9.2/82-2, REV 15 (May 8, 1985). Descriptive material associated with each CDB is, in some cases, abridged from the ANSI document. For commands with extended description, such as COPY and FORMAT, the description is largely omitted, and summarized only.

Revision 15 of the proposed SCSI standard has been reviewed by ANSI Subcommittee X3T9 on I/O interfaces. It is still awaiting approval by the ANSI X3 Committee.

A.1.1 TEST UNIT READY Command

Peripheral Device Type:	All
Operation Code Type:	Optional
Operation Code:	00_H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Reserved							
5	Vendor Unique		Reserved				Flag	Link

The TEST UNIT READY command provides a means to check if the logical unit is ready. This is not a request for a self test. If the logical unit would accept an appropriate medium-access command without returning CHECK CONDITION status, this command shall return a GOOD status.

A.1.2 REZERO UNIT Command

Peripheral Device Type:	Direct Access
Operation Code Type:	Optional
Operation Code:	01H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Reserved							
5	Vendor Unique		Reserved				Flag	Link

The REZERO UNIT command requests that the target set the logical unit to a specific state. See vendor specifications for details.

A.1.3 REQUEST SENSE Command

Peripheral Device Type: All
Operation Code Type: Mandatory
Operation Code: 03H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Allocation Length							
5	Vendor Unique		Reserved				Flag	Link

The REQUEST SENSE command requests that the target transfer sense data to the initiator

The sense data is valid for a CHECK CONDITION status returned on the prior command. The target preserves this sense data until retrieved by the initiator via the REQUEST SENSE command or until the receipt of any other command for the same logical unit from the initiator that issued the command resulting in the CHECK CONDITION status. The target clears the sense data upon receipt of any subsequent command to the logical unit from the initiator receiving the CHECK CONDITION status. In the case of the single initiator option, the target assumes that the REQUEST SENSE command is from the same initiator.

For additional information on sense data, refer to Appendix B.

A.1.4 FORMAT UNIT Command

Peripheral Device Type:	Direct Access
Operation Code Type:	Mandatory
Operation Code:	04_H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			FmtData	CmpLst	Defect List Format		
2	Vendor Unique							
3	Interleave (MSB)							
4	Interleave (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The FORMAT UNIT command ensures that the medium is formatted so that all data blocks can be accessed. There is no guarantee that the medium has or has not been altered. In addition, the medium may be certified and control structures be created for the management of the medium and defects. With the large number of specified options, almost any media verification or test pattern can be implemented without the use of vendor unique options. Implementers and users should study the many options of this command very carefully before creating a vendor unique command to achieve a particular format or media test.

The FORMAT UNIT command is both a mandatory command and an extended command with multiple options.

A format data (FmtData) bit of one indicates that format data is supplied during the DATA OUT phase. The defect list included with this data specifies the defects that shall be entered into the defect map. The format of the defect list is determined by defect list format. A FmtData bit of zero indicates that the DATA OUT phase shall not occur (no defect data shall be supplied by the initiator).

A complete list (CmpLst) bit of one indicates the data supplied is the complete list of known defects. Any previous initiator-specified defect map or defect data shall be erased. The target may add to this list as it formats the medium. The result is to purge any previous initiator-specified defect list and to build a new defect list. A CmpLst bit of zero indicates that the data supplied is in addition to existing defect data using the current format. When using the block format, the defect list refers to the current block length (and not to the new block length, if it is different) and the defect list refers to current logical block addresses (not physical addresses). The target may add to this list as it formats the medium.

The defect list format field specifies additional information related to the defect list.

The interleave field requests that the logical blocks be related in a specific fashion to the physical blocks to facilitate speed matching. An interleave value of zero requests that the target use its default interleave. An interleave value of one requests that consecutive logical blocks be placed in consecutive physical order. Values of two or greater are vendor unique.

Figure A-1: FORMAT UNIT Command Variations

Bit Reference 4 FmtData	3 CmpLst	2 Defect List Format	1	0	Command Type	Comments
0	X	X	X	X	Mandatory	Format with no defect data sent from the initiator to the target.
1	0	0	X	X	Extended	Format adding the defects specified in the defect list to the known defects.
1	1	0	X	X	Optional	Format using defects in the defect list as the full set of known defects.
1	0	1	0	0	Optional	Format adding the defects in the defect list to the known defects.
1	1	1	0	0	Optional	Format using the defects in the defect list as the full set of known defects.
1	0	1	0	1	Optional	Format adding the defects in the defect list to the known defects.
1	1	1	0	1	Optional	Format using the defects in the defect list as the full set of known defects.
1	X	1	1	0	Vendor unique	
1	0	1	1	1	Reserved	
1	1	1	1	1	Reserved	

X = 1 or 0 (i.e., don't care term).

The defect lists shown in Figures A-2, A-3, A-4, and A-5 contain a four-byte header followed by one or more defect descriptors. The length of the defect descriptors varies with the format of the defect list.

The defect list length in each table specifies the total length in bytes of the defect descriptors that follow. In Figure A-2, the defect list length is equal to four times the number of defect descriptors. In Figures A-3 and A-4, the defect list length is equal to eight times the number of defect descriptors.

Figure A-2: Defect List - Block Format

Byte	Defect List Header
0	Reserved
1	Reserved
2	Defect List Length (MSB)
3	Defect List Length (LSB)
	Defect Descriptor(s)
0	Defect Block Address (MSB)
1	Defect Block Address
2	Defect Block Address
3	Defect Block Address (LSB)

Each defect descriptor for the block format specifies a four-byte defect block address that contains the defect. The defect descriptors shall be in ascending order.

Figure A-3: Defect List - Bytes From Index Format

Byte	Defect List Header
0	Reserved
1	Reserved
2	Defect List Length (MSB)
3	Defect List Length (LSB)
	Defect Descriptor(s)
0	Cylinder Number of Defect (MSB)
1	Cylinder Number of Defect
2	Cylinder Number of Defect (LSB)
3	Head Number of Defect
4	Defect Bytes from Index (MSB)
5	Defect Bytes from Index
6	Defect Bytes from Index
7	Defect Bytes from Index (LSB)

Each defect descriptor for the bytes from index format specifies the beginning of an eight-byte defect location on the medium. Each defect descriptor is composed of the cylinder number of defect, the head number of defect, and the defect bytes from index. The defect descriptors shall be in ascending order. For determining ascending order, the cylinder number of defect is considered the most significant part of the address and the defect bytes from index is considered the least significant part of the address.

A defect bytes from index of FFFFFFFF_H indicates that the entire track shall be reassigned.

Figure A-4: Defect List - Physical Sector Format

Byte	Defect List Header
0	Reserved
1	Reserved
2	Defect List Length (MSB)
3	Defect List Length (LSB)
	Defect Descriptor(s)
0	Cylinder Number of Defect (MSB)
1	Cylinder Number of Defect
2	Cylinder Number of Defect (LSB)
3	Head Number of Defect
4	Defect Sector Number (MSB)
5	Defect Sector Number
6	Defect Sector Number
7	Defect Sector Number (LSB)

Each defect descriptor for the physical sector format specifies a sector-size defect location composed of the cylinder number of defect, the head number of defect, and the defect sector number. The defect descriptors shall be in ascending order. For determining ascending order, the cylinder number of defect is considered the most significant part of the address and the defect sector number is considered the least significant part of the address.

A defect sector number of FFFFFFFFH indicates that the entire track shall be reassigned.

A.1.5 REASSIGN BLOCKS Command

Peripheral Device Type: Direct Access and Write-Once Read-Multiple
Operation Code Type: Optional
Operation Code: 07_H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Reserved							
5	Vendor Unique		Reserved				Flag	Link

The REASSIGN BLOCKS command requests the target to reassign the defective logical blocks to an area on the logical unit reserved for this purpose.

The initiator transfers a defect list that contains the logical block addresses to be reassigned. Note that the initiator does not specify the physical medium to which logical blocks are reassigned. The target reassigns the physical medium used for each logical block address in the list. The data contained in the logical blocks specified in the defect list may be altered, but the data in all other logical blocks on the medium shall be preserved.

The effect of specifying a logical block to be reassigned that previously has been reassigned is to reassign the block again. Thus, over the life of the medium, a logical block can be assigned to multiple physical addresses (until no more spare locations remain on the medium).

The REASSIGN BLOCKS defect (Figure A-5) contains a four-byte header followed by one or more defect descriptors. The length of each defect descriptor is four bytes.

The defect list length specifies the total length in bytes of the defect descriptors that follow. The defect list length is equal to four times the number of defect descriptors.

Figure A-5: REASSIGN BLOCKS Defect List

Byte	Defect List Header
0	Reserved
1	Reserved
2	Defect List Length (MSB)
3	Defect List Length (LSB)
	Defect Descriptor(s)
0	Defect Logical Block Address (MSB)
1	Defect Logical Block Address
2	Defect Logical Block Address
3	Defect Logical Block Address (LSB)

The defect descriptor specifies a four-byte defect logical block address that contains the defect. The defect descriptors are listed in ascending order.

If the logical unit has insufficient capacity to reassign all of the defective logical blocks, the command terminates with a CHECK CONDITION status and the sense key is set to MEDIUM ERROR. The logical block address of the first logical block not reassigned is returned in the information bytes of the sense data.

A.1.6 READ Command

Peripheral Device Type: Direct Access
Operation Code Type: Mandatory
Operation Code: 08_H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Logical Block Address (MSB)				
2	Logical Block Address							
3	Logical Block Address (LSB)							
4	Transfer Length							
5	Vendor Unique		Reserved				Flag	Link

The READ command requests that the target transfer data to the initiator.

The logical block address specifies the logical block at which the read operation begins.

The transfer length specifies the number of contiguous logical blocks of data to transferred. A transfer length of zero indicates that 256 logical blocks shall be transferred. Any other value indicates the number of logical blocks to be transferred.

This command is terminated with a RESERVATION CONFLICT status if any reservation access conflict exists, and no data shall be read.

If any of the following conditions occur, this command is terminated with a CHECK CONDITION status, and if extended sense is implemented, the sense key is set as indicated in the following table. Figure A-6 does not provide an exhaustive enumeration of all conditions that may cause the CHECK CONDITION status.

Figure A-6: Selected CHECK CONDITIONS for READ Command

Condition	Sense Key
Invalid logical block address	ILLEGAL REQUEST (see note)
Target reset or medium change since last command from this initiator	UNIT ATTENTION
Unrecoverable read error	MEDIUM ERROR
Recovered read error	RECOVERED ERROR
Overrun or other error that might be resolved by repeating the command	ABORTED COMMAND

NOTE: The extended sense information bytes shall be set to the logical block address of the first invalid address.

A.1.7 WRITE Command

Peripheral Device Type: Direct Access
Operation Code Type: Mandatory
Operation Code: $0A_H$

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Logical Block Address (MSB)				
2	Logical Block Address							
3	Logical Block Address (LSB)							
4	Transfer Length							
5	Vendor Unique		Reserved				Flag	Link

The WRITE command requests that the target write the data transferred by the initiator to the medium.

The logical block address specifies the logical block at which the write operation begins.

The transfer length specifies the number of contiguous logical blocks of data to transferred. A transfer length of zero indicates that 256 logical blocks are to be transferred. Any other value indicates the number of logical blocks to be transferred.

This command shall be terminated with a RESERVATION CONFLICT status if any reservation access conflict exists, and no data shall be written.

If any of the following conditions occur, this command shall be terminated with a CHECK CONDITION status, and if extended sense is implemented, the sense key shall be set as indicated in the following table. Figure A-7 below does not provide an exhaustive enumeration of all conditions that may cause the CHECK CONDITION status.

Figure A-7: Selected CHECK CONDITIONS for WRITE Command

Condition	Sense Key
Invalid logical block address	ILLEGAL REQUEST (see note)
Target reset or medium change since last command from this initiator	UNIT ATTENTION
Overrun or other error that might be resolved by repeating the command	ABORTED COMMAND

NOTE: The extended sense information bytes shall be set to the logical block address of the first invalid address. In this case, no data shall be written on the logical unit.

A.1.8 SEEK Command

Peripheral Device Type:	Direct Access, Write-Once Read-Multiple, and Read-Only Direct Access
Operation Code Type:	Optional
Operation Code:	$0B_H$

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Logical Block Address (MSB)				
2	Logical Block Address							
3	Logical Block Address (LSB)							
4	Reserved							
5	Vendor Unique		Reserved				Flag	Link

The SEEK command requests that the logical unit seek to the specified logical block address.

A.1.9 INQUIRY Command

Peripheral Device Type: All
Operation Code Type: Extended
Operation Code: 12_H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Allocation Length							
5	Vendor Unique		Reserved				Flag	Link

The INQUIRY command requests that information regarding parameters of the target and its attached peripheral device(s) be sent to the initiator.

The allocation length specifies the number of bytes that the initiator has allocated for returned INQUIRY data. An allocation length of zero indicates that no INQUIRY data is to be transferred. This condition is not to be considered as an error. Any other value indicates the maximum number of bytes to be transferred. The target terminates the DATA IN phase when allocation length bytes have been transferred or when all available INQUIRY data have been transferred to the initiator, whichever is less.

A CHECK CONDITION status is reported only when the target cannot return the requested INQUIRY data.

The INQUIRY data, shown in Figure A-8, contains a five byte header, followed by the vendor unique parameters, if any.

Figure A-8: INQUIRY Data

Bit / Byte	7	6	5	4	3	2	1	0
0	Peripheral Device Type							
1	RMB	Device-Type Qualifier						
2	Version							
3	Reserved							
4	Additional Length (n)							
	Vendor Unique Parameters							
5 – n+4	Vendor Unique Parameter Bytes							

The peripheral device type is shown in Figure A-9 below.

Figure A-9: Peripheral Device Type

Code	Description
00_H	Direct-access device (e.g., magnetic disk)
01_H	Sequential-access device (e.g., magnetic tape)
02_H	Printer device
03_H	Processor device
04_H	Write-once read-multiple device (e.g., some optical disks)
05_H	Read-only direct-access device (e.g., some optical disks)
06_H – $7E_H$	Reserved
$7F_H$	Logical unit not present
80_H – FF_H	Vendor unique

A removable medium (RMB) bit of zero indicates that the medium is not removable. A RMB bit of one indicates that the medium is removable.

The device-type qualifier is a seven bit user specified code. This code may be set with switches or by some other means by the target or peripheral device. SCSI devices that do not support this feature must return all zero bits. This feature allows each user to assign unique codes to each specific type of peripheral device that is supported on the system being used. These codes may then be used by self-configuring software to determine what specific peripheral device is at each logical unit number. This is especially valuable for systems that support multiple types of removable medium.

The version is the implemented version of this standard and is defined as shown in the table below.

Figure A-10: Version

Code	Description
00_H	Version is unspecified
01_H	This version. This code shall be used by SCSI devices that claim to comply with this standard (upon ANSI publication)
02_H – FF_H	Reserved

The additional length shall specify the length in bytes of the vendor unique parameters. If the allocation length of the command descriptor block is too small to transfer all of the vendor unique parameters, the additional length shall not be adjusted to reflect the truncation.

A.1.10 MODE SELECT Command

Peripheral Device Type: Direct Access
Operation Code Type: Optional
Operation Code: 15_H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Parameter List Length							
5	Vendor Unique		Reserved				Flag	Link

The MODE SELECT command provides a means for the initiator to specify medium, logical unit, or peripheral device parameters to the target.

The parameter list length specifies the length in bytes of the MODE SELECT parameter list that shall be transferred during the DATA OUT phase. A parameter list length of zero indicates that no data is to be transferred. This condition is not considered as an error.

The MODE SELECT parameter list, shown in Figure A-11, contains a four-byte header, followed by zero or more block descriptors, followed by the vendor unique parameters, if any.

Figure A-11: MODE SELECT Parameter List

Byte	MODE SELECT Header
0	Reserved
1	Medium Type
2	Reserved
3	Block Descriptor Length
	Block Descriptor(s)
0	Density Code
1	Number of Blocks (MSB)
2	Number of Blocks
3	Number of Blocks (LSB)
4	Reserved
5	Block Length (MSB)
6	Block Length
7	Block Length (LSB)
	Vendor Unique Parameter(s)
0 – n	Vendor Unique Parameter Byte(s)

Code values for the medium type field shall be assigned as follows:

00H	Default (currently mounted medium type)
01H	Flexible disk, single-sided diskette
02H	Flexible disk, double-sided diskette
03H-7FH	Reserved
80H-FFH	Vendor unique

The block descriptor length specifies the length in bytes of all the block descriptors. It is equal to the number of block descriptors times eight and does not include the vendor unique parameters, if any. A block descriptor length of zero indicates that no block descriptors are included in the parameter list. This condition is not considered as an error.

Each block descriptor specifies the medium characteristics for all or part of a logical unit. Each block descriptor contains a density code, a number of blocks, and a block length.

The number of blocks field specifies the number of logical blocks on the medium that meet the density code and block length in the block descriptor. A number of blocks of zero indicates that all of the remaining logical blocks of the logical unit shall have the medium characteristics specified by the block descriptor.

The block length specifies the length in bytes of each logical block described by the block descriptor.

A.1.11 RESERVE Command

Peripheral Device Type:	Direct Access, Write-Once Read-Multiple, and Read-Only Direct Access
Operation Code Type:	Optional
Operation Code:	16_H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			3rdPty	Third Party Device ID			Extent
2	Reservation Identification							
3	Extent List Length (MSB)							
4	Extent List Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The RESERVE command is used to reserve logical units, or, if the extent reservation option is implemented, extents within logical units for the use of the initiator.

If the extent bit is one, and the extent reservation option is not implemented, then the RESERVE command shall be rejected with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST. Refer to the ANSI document for further details on the RESERVE Command.

Figure A-12: Data Format of Extent Descriptors

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved					RelAdr	Reservation Type	
1	Number of Blocks (MSB)							
2	Number of Blocks							
3	Number of Blocks (LSB)							
4	Logical Block Address (MSB)							
5	Logical Block Address							
6	Logical Block Address							
7	Logical Block Address (LSB)							

The size of the extent list shall be defined by the extent list length parameter. The extent list shall consist of one or more descriptors as shown in Figure A-12. Each extent descriptor defines an extent beginning at the specified logical block address for the specified number of blocks. If the number of blocks is zero, the extent shall begin at the specified logical block address and continue through the last logical block address on the logical unit.

The reservation type field shall determine the type of reservation to be effected for each extent. The four possible types of reservations are:

DB(1)	DB(0)	Reservation Type
1	0	Read Exclusive
0	1	Write Exclusive
1	1	Exclusive Access
0	0	Read Shared

Read Exclusive. While this reservation is active, no other initiator shall be permitted read operations to the indicated extent. This reservation shall not inhibit write operations from any initiator or conflict with a write exclusive reservation; however, read exclusive, exclusive access, and read shared reservations that overlap this extent shall conflict with this reservation.

Write Exclusive. While this reservation is active, no other initiator shall be permitted write operations to the indicated extent. This reservation shall not inhibit read operations from any initiator or conflict with a read exclusive reservation from any initiator. This reservation shall conflict with write exclusive, exclusive access, and read shared reservations that overlap this extent.

Exclusive Access. While this reservation is active, no other initiator shall be permitted any access to the indicated extent. All reservation types that overlap this extent shall conflict with this reservation.

Read Shared. While this reservation is active, no write operations shall be permitted by any initiator to the indicated extent. This reservation shall not inhibit read operations from any initiator or conflict with a read shared reservation. Read exclusive, write exclusive, and exclusive access reservations that overlap with this extent shall conflict with this reservation.

If the relative address bit is one, the logical block address shall be treated as a two's complement displacement. This displacement shall be added to the logical block address last accessed on the logical unit to form the logical block address for this extent. This feature is only available when linking commands and requires that a previous command in the linked group has accessed a logical block on the logical unit; if not, the RESERVE command shall be terminated with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

If an initiator attempts a command to a logical block that has been reserved and that access is prohibited by the reservation, the command shall not be performed and the command shall be terminated with a RESERVATION CONFLICT status. If a reservation conflict precludes any part of the command, none of the command shall be performed. COPY commands shall be terminated with a CHECK CONDITION status and the sense key shall be set to DATA PROTECT if any part of the copy operation is prohibited by an active reservation. If any extent in a logical unit is reserved in any way, a FORMAT UNIT command shall be rejected with a RESERVATION CONFLICT status.

The ANSI document also defines third party reservation and release operations. The third party reeservation allows an initiator to reserve a logical unit or extents within a logical unit for another SCSI device. This option is intended for use in multiple-initiator systems that use the COPY command.

A.1.12 RELEASE Command

Peripheral Device Type:	Direct Access, Write-Once Read-Multiple, and Read-Only Direct Access
Operation Code Type:	Optional
Operation Code:	17_H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			3rdPty	Third Party Device ID			Extent
2	Reservation Identification							
3	Reserved							
4	Reserved							
5	Vendor Unique		Reserved				Flag	Link

The RELEASE command is used to release previously reserved logical units, or, if the extent release option is implemented, previously reserved extents within logical units. It is not an error for an initiator to attempt to release a reservation that is not currently active.

If the extent bit is zero, the RELEASE command shall cause the target to terminate any reservation that is active from the initiator to the specified logical unit.

If the extent bit is one and the extent release option is not implemented, then the RELEASE command is terminated with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST. This option must be implemented if the extent reservation option is implemented.

If the extent bit is one and the extent release option is implemented, this command causes any reservation from the requesting initiator with a matching reservation identification that is active or queued to be terminated. Other reservations from the requesting initiator shall remain in effect.

If the logical unit queues reservations, then when a RELEASE command is processed, the reservation queue shall be examined on a first-in first-out basis. If there are one or more reservations in the queue that can now be activated, the logical unit shall first disconnect from the initiator. It shall then successively reconnect with each initiator whose queued reservation may now be activated. A queued reservation request shall not be activated if it conflicts with any previously queued reservation. After first granting all possible queued reservations, the unit shall reconnect with the initiator of the RELEASE command.

If a logical unit that queues reservations receives a RELEASE command from a second initiator while it is disconnected during processing of a previous RELEASE command, it shall then disconnect from the second initiator and suspend processing of the second RELEASE until after reconnection with the first initiator, or until it is determined that reconnection has failed.

The third-party release option for the RELEASE command allows an initiator to release a logical unit or extents within a logical unit that were previously reserved using the third-party reservation option. This option must be implemented if the third-party reservation option is implemented. This option is intended for use in multiple-initiator systems that use the COPY command.

A.1.13 COPY Command

Peripheral Device Type: All
Operation Code Type: Optional
Operation Code: 18_H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Parameter List Length (MSB)							
3	Parameter List Length							
4	Parameter List Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The COPY command provides a means to copy data from one logical unit to another or the same logical unit. The logical units may reside on the same SCSI device or different SCSI devices. Some SCSI devices that implement this command may not support copies to or from another SCSI device or third party copies (both logical units reside on other SCSI devices).

The parameter list length specifies the length in bytes of the parameters which will be sent during the DATA OUT phase of the command. A parameter list length of zero indicates that no data shall be transferred.

The COPY parameter list, shown in Figure A-13, begins with a four-byte header that contains the COPY function code and priority. Following the header is one or more segment descriptors.

Figure A-13: COPY Parameter List

Bit / Byte	7	6	5	4	3	2	1	0
0	COPY Function Code					Priority		
1	Vendor Unique							
2	Reserved							
3	Reserved							
	Segment Descriptor(s)							
0 _ xx	Segment Descriptor 0 (See specific table for length.)							
	. . .							
0 _ xx	Segment Descriptor n (See specific table for length.)							

The COPY function code defines a specific format for the segment descriptors.

The priority field of the COPY parameter list establishes the relative priority of this COPY command to other commands being executed by the same target. All other commands are assumed to have a priority of 1. Priority 0 is the highest priority with increasing values indicating lower priorities.

The segment descriptor formats are determined by the COPY function code. The segment descriptor format used for write-once read-multiple devices and for read-only direct-access devices shall be the same as for direct-access devices. The segment descriptor format used for printer devices and for processor devices shall be the same as for sequential-access devices. Thus a COPY from a write-once read-multiple device to a printer device uses the same segment descriptor format as for a COPY from a direct-access device to a sequential-access device. A maximum of 256 segment descriptors are permitted. The segment descriptors are identified by ascending numbers beginning with zero.

Figure A-14: COPY Functions

Peripheral Device Type		COPY Function Code	Segment Descriptor Table	Comment
Source	Destination			
00H	01H	00H	Figure A-15	
00H	02H	00H	Figure A-15	
00H	03H	00H	Figure A-15	
04H	01H	00H	Figure A-15	Direct Access
04H	02H	00H	Figure A-15	to
04H	03H	00H	Figure A-15	Sequential Access
05H	01H	00H	Figure A-15	
05H	02H	00H	Figure A-15	
05H	03H	00H	Figure A-15	
01H	00H	01H	Figure A-15	Sequential Access
01H	04H	01H	Figure A-15	to
03H	00H	01H	Figure A-15	Direct Access
03H	04H	01H	Figure A-15	
00H	00H	02H	Figure A-16	
00H	04H	02H	Figure A-16	Direct Access
04H	00H	02H	Figure A-16	to
04H	04H	02H	Figure A-16	Direct Access
05H	00H	02H	Figure A-16	
05H	04H	02H	Figure A-16	
01H	01H	03H	Figure A-17	
01H	02H	03H	Figure A-17	Sequential Access
01H	03H	03H	Figure A-17	to
03H	01H	03H	Figure A-17	Sequential Access
03H	02H	03H	Figure A-17	
03H	03H	03H	Figure A-17	

Peripheral device type:	00H	Direct-access device
	01H	Sequential-access device
	02H	Printer device
	03H	Processor device
	04H	Write-once read-multiple device
	05H	Read-only direct-access device

COPY function code:	00H	Direct access to sequential access
	01H	Sequential access to direct access
	02H	Direct access to direct access
	03H	Sequential access to sequential access
	04H _ 0FH	Reserved
	10H _ 1FH	Vendor unique

Errors Detected by the Managing SCSI Device. Two classes of unusual conditions may occur during execution of a COPY command. The first class consists of those unusual conditions detected by the SCSI device that received the COPY command and is managing the execution of the command. These conditions include parity errors while transferring the COPY command and status byte, invalid parameters in the COPY command, invalid segment descriptors, and inability of the SCSI device controlling the COPY functions to continue operating. In the event of such an unusual condition, the SCSI device managing the COPY shall:

(1) Terminate the COPY command with a CHECK CONDITION status.

(2) Return the sense data in the extended sense format. The valid bit shall be set to one. The segment number shall contain the number of the segment descriptor being processed at the time the unusual condition is detected. The sense key shall contain the sense key code describing the unusual condition. The information bytes shall contain the difference between the number of blocks field in the segment descriptor being processed at the time of the failure and the number of blocks successfully copied. This number is the residue of unprocessed blocks remaining for the segment descriptor.

Errors Detected by a Target. The second class of errors consists of unusual conditions detected by the SCSI device transferring data at the request of the SCSI device managing the transfer. The SCSI device managing the COPY command detects unusual conditions by receiving a CHECK CONDITION status from one of the SCSI devices it is managing. It then shall recover the sense data associated with the unusual condition.

The SCSI device managing the COPY command may also be the source or destination SCSI device (or both). It shall distinguish between a failure of the management of the COPY and a failure of the data transfer being requested. It shall then create the appropriate sense data internally.

After recovering the sense data associated with the detected error, the SCSI device managing the COPY command shall:

(1) Terminate the COPY command with a CHECK CONDITION status.

(2) Return the sense data in the extended sense format. The valid bit shall be set to one. The segment number shall contain the number of the segment descriptor being processed at the time the unusual condition is detected. The sense key shall be set to COPY ABORTED. The information bytes shall contain the difference between the number of blocks field in the segment descriptor being processed at the time of the failure and the number of blocks successfully copied. This number is the residue of unprocessed blocks remaining for the segment descriptor. The additional sense length shall specify the number of additional sense bytes.

The first additional sense byte specifies the byte number, relative to the first byte of sense data of the beginning of the source logical unit's status byte and sense data. A zero value indicates that no status byte or sense data is being returned for the source logical unit. The first byte of the area pointed to by the first additional sense byte contains the status byte from the source logical unit. The subsequent bytes contain, unchanged, the sense data recovered from the source logical unit.

The second additional sense byte shall specify the byte number, relative to the first byte of sense data of the beginning of the destination logical unit's status byte and sense data. A zero value indicates that no status byte or sense data is being returned for the destination logical unit. The first byte of the area pointed to by the second additional sense byte shall contain the status byte from the destination logical unit. The subsequent bytes shall contain, unchanged, the sense data recovered from the destination logical unit.

COPY Function Code 00H and 01H. The format for the segment descriptors for COPY transfers between direct-access and sequential-access devices is specified in Figure A-15 below. This format is required for COPY function codes 00H or 01H. The segment descriptor may be repeated up to 256 times within the parameter list length specified in the command descriptor block.

Figure A-15: Segment Descriptor for COPY Function Codes 00H and 01H

Bit / Byte	7	6	5	4	3	2	1	0
0	Source Address			Reserved		Source LUN		
1	Destination Address			Reserved		Destination LUN		
2	Sequential-Access Device Block-Length (MSB)							
3	Sequential-Access Device Block-Length (LSB)							
4	Direct-Access Device Number of Blocks (MSB)							
5	Direct-Access Device Number of Blocks							
6	Direct-Access Device Number of Blocks							
7	Direct-Access Device Number of Blocks (LSB)							
8	Direct-Access Device Logical Block Address (MSB)							
9	Direct-Access Device Logical Block Address							
10	Direct-Access Device Logical Block Address							
11	Direct-Access Device Logical Block Address (LSB)							

Source address and destination address fields specify the SCSI devices and the source LUN and destination LUN fields specify the logical units to use for this segment of the COPY command. Some SCSI devices may not support "third-party" COPY in which the copying SCSI device is not the source or destination device. Some SCSI devices only support COPY within the SCSI device and not to other SCSI devices. If an unsupported COPY operation is requested, the command is terminated with a CHECK CONDITION status and the sense key is set to ILLEGAL REQUEST.

The sequential-access device block-length field specifies the block-length to be used on the sequential-access logical unit during this segment of the COPY command. If this block-length is known by the SCSI device managing the COPY to be not supported, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST. If the block-length is found to be invalid while executing a read or write operation to the sequential-access device, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to COPY ABORTED.

The direct-access device number of blocks field specifies the number of blocks in the current segment. A value of zero indicates that no blocks shall be transferred in this segment. The direct-access device logical block address specifies the starting logical block address on the logical unit for this segment.

COPY Function Code 02H. The format for the segment descriptors for COPY transfers among direct-access devices is specified in Figure A-16. This format is required for COPY function code 02_H. The segment descriptor may be repeated up to 256 times within the parameter list length specified in the command descriptor block.

Figure A-16: Segment Descriptor for COPY Function Code 02H

Bit / Byte	7	6	5	4	3	2	1	0
0	Source Address			Reserved		Source LUN		
1	Destination Address			Reserved		Destination LUN		
2	Reserved							
3	Reserved							
4	Source Number of Blocks (MSB)							
5	Source Number of Blocks							
6	Source Number of Blocks							
7	Source Number of Blocks (LSB)							
8	Source Logical Block Address (MSB)							
9	Source Logical Block Address							
10	Source Logical Block Address							
11	Source Logical Block Address (LSB)							
12	Destination Logical Block Address (MSB)							
13	Destination Logical Block Address							
14	Destination Logical Block Address							
15	Destination Logical Block Address (LSB)							

The source address and destination address fields specify the SCSI devices and the source LUN and destination LUN specify the logical units to use for this segment of the COPY command. Some SCSI devices may not support "third-party" COPY in which the copying SCSI device is not the source or destination device. Some SCSI devices only support COPY within the SCSI device and not to other SCSI devices. If an unsupported COPY operation is requested, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

The source number of blocks field specifies the number of blocks to be transferred from the source device during command execution. The source logical block address field specifies the starting logical block address on the source device. The destination logical block address field specifies the starting logical block address on the destination device.

COPY Function Code 03H. The format for the segment descriptors for COPY transfers among sequential-access devices is specified by Figure A.17. This format is required for COPY function code 03_H. The segment descriptor may be repeated up to 256 times within the parameter list length specified in the command descriptor block.

Figure A-17: Segment Descriptor for COPY Function Code 03H

Bit / Byte	7	6	5	4	3	2	1	0
0	Source Address			Reserved		Source LUN		
1	Destination Address			Reserved		Destination LUN		
2	Reserved							
3	Reserved							
4	Source Block Length (MSB)							
5	Source Block Length (LSB)							
6	Destination Block Length (MSB)							
7	Destination Block Length (LSB)							
8	Source Number of Blocks (MSB)							
9	Source Number of Blocks							
10	Source Number of Blocks							
11	Source Number of Blocks (LSB)							

Source address and destination address fields specify the SCSI devices and the source LUN and destination LUN fields specify the logical units to use for this segment of the COPY command. Some SCSI devices may not support "third-party" COPY in which the copying SCSI device is not the source or destination device. Some SCSI devices only support COPY within the SCSI device and not to other SCSI devices. If an unsupported COPY operation is requested, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

The source block-length field specifies the block-length of the source device for this segment of the COPY. A zero in this field indicates variable block-length. For nonzero values, this field shall match the logical unit's actual block-length. If block-length mismatches are detected by the SCSI device managing the COPY, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST. If the mismatches are detected during the read operation by the COPY manager, the command is terminated with a CHECK CONDITION status and the sense key is set to COPY ABORTED.

The destination block-length field specifies the block-length to be used on the destination logical unit during the COPY. Destination block-length mismatches are handled in the same manner as source block-length mismatches.

The source number of blocks field specifies the number of blocks to be transferred from the source device during this segment. A value of zero indicates that no blocks shall be transferred.

A.1.14 MODE SENSE Command

Peripheral Device Type: Direct Access
Operation Code Type: Optional
Operation Code: $1A_H$

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Allocation Length							
5	Vendor Unique		Reserved				Flag	Link

The MODE SENSE command provides a means for a target to report its medium, logical unit, or peripheral device parameters to the initiator. It is a complementary command to the MODE SELECT command for support of medium that may contain multiple block lengths or densities.

The allocation length specifies the number of bytes that the initiator has allocated for returned MODE SENSE data. An allocation length of zero indicates that no MODE SENSE data shall be transferred. This condition is not considered as an error. Any other value indicates the maximum number of bytes that shall be transferred. The target terminates the DATA IN phase when allocation length bytes have been transferred or when all available MODE SENSE data have been transferred to the initiator, whichever is less.

The MODE SENSE data, shown in Figure A-17, contains a four-byte header, followed by zero or more eight-byte block descriptors, followed by the vendor unique parameters, if any.

Figure A-18: MODE SENSE Data

Bit / Byte	7	6	5	4	3	2	1	0
0	Sense Data Length							
1	Medium Type							
2	WP	Reserved						
3	Block Descriptor Length							
	Block Descriptor(s)							
0	Density Code							
1	Number of Blocks (MSB)							
2	Number of Blocks							
3	Number of Blocks (LSB)							
4	Reserved							
5	Block Length (MSB)							
6	Block Length							
7	Block Length (LSB)							
	Vendor Unique Parameter(s)							
0 _ n	Vendor Unique Parameter Byte(s)							

The sense data length specifies the length in bytes of the following MODE SENSE data that is available to be transferred during the DATA IN phase. The sense data length does not include itself.

Code values for the medium type field shall be assigned as follows:

00H	Default (only one medium type supported)
01H	Flexible disk, single-sided diskette
02H	Flexible disk, double-sided diskette
03H-7FH	Reserved
80H-FFH	Vendor unique

A write protected (WP) bit of zero indicates that the medium is write enabled. A WP bit of one indicates that the medium is write protected.

The block descriptor length specifies the length in bytes of all the block descriptors. It is equal to the number of block descriptors times eight and does not include the vendor unique parameters, if any. A block descriptor length of zero indicates that no block descriptors shall be included in the parameter list. This condition shall not be considered as an error.

Each block descriptor specifies the medium characteristics for all or part of a logical unit. Each block descriptor contains a density code, a number of blocks, and a block length.

The number of blocks field specifies the number of logical blocks of the medium that meets the density code and block length in the block descriptor. A number of blocks of zero indicates that all of the remaining logical blocks of the logical unit have the medium characteristics specified by the block descriptor.

The block length specifies the length in bytes of each logical block.

A.1.15 START/STOP UNIT Command

Peripheral Device Type:	Direct Access, Write-Once Read-Multiple, and Read-Only Direct Access
Operation Code Type:	Optional
Operation Code:	$1B_H$

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				Immed
2	Reserved							
3	Reserved							
4	Reserved							Start
5	Vendor Unique		Reserved				Flag	Link

The START/STOP UNIT command requests that the target enable or disable the logical unit for further operations.

An immediate (Immed) bit of one indicates that status shall be returned as soon as the operation is initiated. An Immed bit of zero indicates that status shall be returned after the operation is completed.

A start bit of one requests the logical unit be made ready for use. A start bit of zero requests that the logical unit be stopped.

A.1.16 RECEIVE DIAGNOSTIC RESULTS Command

Peripheral Device Type:	All
Operation Code Type:	Optional
Operation Code:	1CH

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Allocation Length (MSB)							
4	Allocation Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The RECEIVE DIAGNOSTIC RESULTS command requests analysis data be sent to the initiator after completion of a SEND DIAGNOSTIC command (see A.1.17)

The allocation length shall specify the number of bytes that the initiator has allocated for returned diagnostic data. An allocation length of zero indicates that no diagnostic data shall be transferred. Any other value indicates the maximum number of bytes that shall be transferred. The target terminates the DATA IN phase when allocation length bytes have been transferred or when all available diagnostic data have been transferred to the initiator, whichever is less.

The diagnostic data returned is vendor unique.

NOTE: Although diagnostic software is generally device-specific, this command and the SEND DIAGNOSTIC command provide a means to isolate the operating system software from the device-specific diagnostic software. Hence the operating system can remain device-independent. This also allows diagnostic software to be more easily ported to other operating systems.

A.1.17 SEND DIAGNOSTIC Command

Peripheral Device Type: All
Operation Code Type: Optional
Operation Code: $1D_H$

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved		SelfTest	DevOfL	UnitOfL
2	Reserved							
3	Parameter List Length (MSB)							
4	Parameter List Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The SEND DIAGNOSTIC command requests the target to perform diagnostic tests on itself, on the attached peripheral devices, or on both. This command is usually followed by a RECEIVE DIAGNOSTIC RESULTS command, except when the self test (SelfTest) bit is one.

The parameter list length specifies the length in bytes of the parameter list that shall be transferred during the DATA OUT phase. A parameter list length of zero indicates that no data shall be transferred. This condition shall not be considered as an error. The parameter list is vendor unique.

A logical unit off-line (UnitOfL) bit of one enables write operations on user medium or operations that affect user visible medium positioning. An SCSI device off-line (DevOfL) bit of one enables diagnostic operations that may adversely affect operations to other logical units on the same target.

The logical unit off-line and SCSI device off-line bits are generally set by operating system software, while the parameter list is prepared by diagnostic application software. Thus, by preventing operations that are not enabled by these bits, the target assists the operating system in protecting its resources.

A self test bit of one directs the target to complete its default self test. If the self test is requested, the parameter list length shall be set to zero and no data shall be transferred. If the self test successfully passes, the command shall be terminated with a GOOD status; otherwise, the command shall be terminated with a CHECK CONDITION status and, if extended sense is implemented, the sense key shall be set to HARDWARE ERROR.

NOTE: See the note under the RECEIVE DIAGNOSTIC RESULTS command (A.1.16).

A.1.18 PREVENT/ALLOW MEDIUM REMOVAL Command

Peripheral Device Type:	Direct Access, Write-Once Read-Multiple, and Read-Only Direct Access
Operation Code Type:	Optional
Operation Code:	$1E_H$

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Reserved							Prevent
5	Vendor Unique		Reserved				Flag	Link

The PREVENT/ALLOW MEDIUM REMOVAL command requests that the target enable or disable the removal of the medium in the logical unit.

A prevent bit of one shall inhibit mechanisms that normally allow removal of the medium. A prevent bit of zero shall allow removal of the medium.

This prevention of medium removal condition shall terminate upon receipt of a PREVENT/ALLOW MEDIUM REMOVAL command with the prevent bit set to zero, or by the receipt of a BUS DEVICE RESET message from any initiator or by a "hard" RESET condition.

A.2 GROUP 1 COMMANDS FOR ALL DEVICES, AND DIRECT ACCESS DEVICES

A.2.1 READ CAPACITY Command

Peripheral Device Type:	Direct Access, Write-Once Read-Multiple, and Read-Only Direct Access
Operation Code Type:	Extended
Operation Code:	25_H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				RelAdr
2	Logical Block Address (MSB)							
3	Logical Block Address							
4	Logical Block Address							
5	Logical Block Address (LSB)							
6	Reserved							
7	Reserved							
8	Vendor Unique		Reserved					PMI
9	Vendor Unique		Reserved				Flag	Link

The READ CAPACITY command provides a means for the initiator to request information regarding the capacity of the logical unit.

A partial medium indicator (PMI) bit of zero indicates that the information returned in the READ CAPACITY data shall be the logical block address and block length (in bytes) of the last logical block of the logical unit. The logical block address in the command descriptor block shall be to set zero for this option.

A PMI bit of one indicates that the information returned shall be the logical block address and block length (in bytes) of the last logical block address greater than or equal to the logical block address specified in the command descriptor block after which a substantial delay in data transfer will be encountered. (Implementors note: This function is intended to assist storage management software in determining whether there is sufficient space on the current track, cylinder, etc., to contain a frequently accessed data structure such as a file directory or file index.)

The eight bytes of READ CAPACITY data shown in Table 8-24 shall be sent during the DATA IN phase of the command.

Figure A-18: READ CAPACITY Data

Byte	Description
0	Logical Block Address (MSB)
1	Logical Block Address
2	Logical Block Address
3	Logical Block Address (LSB)
4	Block Length (MSB)
5	Block Length
6	Block Length
7	Block Length (LSB)

A.2.2 READ Command

```
Peripheral Device Type:  Direct Access
   Operation Code Type:  Extended
        Operation Code:  28H
```

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				RelAdr
2	Logical Block Address (MSB)							
3	Logical Block Address							
4	Logical Block Address							
5	Logical Block Address (LSB)							
6	Reserved							
7	Transfer Length (MSB)							
8	Transfer Length (LSB)							
9	Vendor Unique		Reserved				Flag	Link

The READ command requests that the target transfer data to the initiator.

The logical block address specifies the logical block at which the read operation shall begin.

The transfer length specifies the number of contiguous logical blocks of data to be transferred. A transfer length of zero indicates that no logical blocks shall be transferred. This condition shall not be considered as an error. Any other value indicates the number of logical blocks that shall be transferred.

The most recent data value written in the addressed logical block shall be returned.

This command shall be terminated with a RESERVATION CONFLICT status if any reservation access conflict exists and no data shall be read.

If any of the following conditions occur, this command shall be terminated with a CHECK CONDITION status and the sense key shall be set as indicated in the following table. This table does not provide an exhaustive enumeration of all conditions that may cause the CHECK CONDITION status.

Condition	Sense Key
Invalid logical block address	ILLEGAL REQUEST (see note)
Target reset or medium change since last command from this initiator	UNIT ATTENTION
Unrecovered read error	MEDIUM ERROR
Recoverable read error	RECOVERED ERROR
Overrun or other error that might be resolved by repeating the command	ABORTED COMMAND

NOTE: The extended sense information bytes shall be set to the logical block address of the first invalid address.

A.2.3 WRITE Command

Peripheral Device Type: Direct Access
Operation Code Type: Extended
Operation Code: $2A_H$

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				RelAdr
2	Logical Block Address (MSB)							
3	Logical Block Address							
4	Logical Block Address							
5	Logical Block Address (LSB)							
6	Reserved							
7	Transfer Length (MSB)							
8	Transfer Length (LSB)							
9	Vendor Unique		Reserved				Flag	Link

The WRITE command requests that the target write the data transferred by the initiator to the medium.

The logical block address specifies the logical block at which the write operation shall begin.

The transfer length specifies the number of contiguous logical blocks of data that shall be transferred. A transfer length of zero indicates that no logical blocks shall be transferred. This condition shall not be considered as an error and no data shall be written. Any other value indicates the number of logical blocks that shall be transferred.

This command shall be terminated with a RESERVATION CONFLICT status if any reservation access conflict exists and no data shall be written.

If any of the following conditions occur, this command shall be terminated with a CHECK CONDITION status and the sense key shall be set as indicated in the following table. This table does not provide an exhaustive enumeration of all conditions that may cause the CHECK CONDITION status.

Condition	Sense Key
Invalid logical block address	ILLEGAL REQUEST (see note)
Target reset or medium change since the last command from this initiator	UNIT ATTENTION
Overrun or other error that might be resolved by repeating the command	ABORTED COMMAND

NOTE: The extended sense information bytes shall be set to the logical block address of the first invalid address. In this case, no data shall be written on the logical unit.

A.2.4 SEEK Command

Peripheral Device Type:	Direct Access, Write-Once Read-Multiple, and Read-Only Direct Access
Operation Code Type:	Optional
Operation Code:	$2B_H$

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Logical Block Address (MSB)							
3	Logical Block Address							
4	Logical Block Address							
5	Logical Block Address (LSB)							
6	Reserved							
7	Reserved							
8	Reserved							
9	Vendor Unique		Reserved				Flag	Link

Tne SEEK command requests that the logical unit seek to the specified logical block address.

A.2.5 WRITE AND VERIFY Command

Peripheral Device Type: Direct Access
Operation Code Type: Optional
Operation Code: 2EH

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved			BytChk	RelAdr
2	Logical Block Address (MSB)							
3	Logical Block Address							
4	Logical Block Address							
5	Logical Block Address (LSB)							
6	Reserved							
7	Transfer Length (MSB)							
8	Transfer Length (LSB)							
9	Vendor Unique		Reserved				Flag	Link

The WRITE AND VERIFY command requests that the target write the data transferred from the initiator to the medium and then verify that the data is correctly written.

A byte check (BytChk) bit of zero causes the verification to be simply a medium verification (CRC, ECC, etc). A BytChk bit of one causes a byte-by-byte compare of data written on the peripheral device and the data transferred from the initiator. If the compare is unsuccessful, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to MISCOMPARE.

The logical block address specifies the logical block at which the write operation shall begin.

The transfer length specifies the number of contiguous logical blocks of data that shall be transferred. A transfer length of zero indicates that no logical blocks shall be transferred. This condition shall not be considered as an error and no data shall be written. Any other value indicates the number of logical blocks that shall be transferred.

A.2.6 VERIFY Command

Peripheral Device Type: Direct Access
Operation Code Type: Optional
Operation Code: $2F_H$

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved			BytChk	RelAdr
2	Logical Block Address (MSB)							
3	Logical Block Address							
4	Logical Block Address							
5	Logical Block Address (LSB)							
6	Reserved							
7	Verification Length (MSB)							
8	Verification Length (LSB)							
9	Vendor Unique		Reserved				Flag	Link

The VERIFY command requests that the target verify the data written on the medium.

A byte check (BytChk) bit of zero causes the verification to be simply a medium verification (CRC, ECC, etc). A BytChk bit of one causes a byte-by-byte compare of data on the medium and the data transferred from the initiator. If the compare is unsuccessful, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to MISCOMPARE.

The logical block address specifies the logical block at which the verify operation shall begin.

The verification length specifies the number of contiguous logical blocks of data that shall be verified. A transfer length of zero indicates that no logical blocks shall be verified. This condition shall not be considered as an error. Any other value indicates the number of logical blocks that shall be verified.

A.2.7 SEARCH DATA Commands

Peripheral Device Type:	Direct Access, Write-Once Read-Multiple, and Read-Only Direct Access
Operation Code Type:	Optional
Operation Code:	30H, 31H, or 32H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Invert	Reserved		SpnDat	RelAdr
2	Logical Block Address (MSB)							
3	Logical Block Address							
4	Logical Block Address							
5	Logical Block Address (LSB)							
6	Reserved							
7	Transfer Length (MSB)							
8	Transfer Length (LSB)							
9	Vendor Unique		Reserved				Flag	Link

The SEARCH DATA commands search one or more logical blocks for equality or inequality to a data pattern. The concept of records within a logical block is used to allow multiple records within a logical block to be searched.

An invert bit of one indicates that the search condition is to be inverted. Thus, a SEARCH DATA EQUAL command (operation code 31H) would search for not-equal data, a SEARCH DATA HIGH command (operation code 30H) would search for less-than-equal or equal data, and a SEARCH DATA LOW command (operation code 32H) would search for greater-than-equal or equal data.

A spanned data (SpnDat) bit of zero indicates that each record shall be wholly contained within a single block. Any space at the end of a block that is smaller than the record length is ignored by the SEARCH DATA commands. A SpnDat bit of one indicates that records span block boundaries. Thus, a record may start in one block and end in the next or a subsequent block.

A transfer length of zero indicates that no data shall be searched. This condition shall be treated the same as an unsatisfied search.

A link bit of zero indicates a nonlinked command and if the search is satisfied, the command shall be terminated with a CONDITION MET status. A REQUEST SENSE command can then be issued to determine the logical block address and record offset of the matching record.

A link bit of one indicates a command is linked to the SEARCH DATA command and if the search is satisfied, the next command is executed. If the RelAdr bit in the next command is one, the logical block address of the next command is used as a displacement from the logical block address at which the search was satisfied. If a linked search is not satisfied, the command is terminated with a CHECK CONDITION status. A REQUEST SENSE command may then be issued.

A REQUEST SENSE command following a satisfied SEARCH DATA command shall:

(1) Return a sense key of EQUAL if the search was satisfied by an exact match. If the search was satisfied by an inequality then a sense key of NO SENSE shall be returned.

(2) Return the valid bit set to one.

(3) Return the logical block address of the logical block containing the first matching record in the information bytes.

(4) Return the record offset of the matching record in the first four bytes of additional sense bytes.

A REQUEST SENSE command following an unsatisfied SEARCH DATA command shall:

(1) Return a sense key of NO SENSE, if no errors occurred during the command execution.

(2) Return the valid bit set to zero.

The SEARCH DATA parameter list (Figure A-20) contains a fourteen-byte header, followed by one or more search argument descriptors.

Figure A-20: SEARCH DATA Parameter List

Byte	Parameter List Header
0	Logical Record Length (MSB)
1	Logical Record Length
2	Logical Record Length
3	Logical Record Length (LSB)
4	First Record Offset (MSB)
5	First Record Offset
6	First Record Offset
7	First Record Offset (LSB)
8	Number of Records (MSB)
9	Number of Records
10	Number of Records
11	Number of Records (LSB)
12	Search Argument Length (MSB)
13	Search Argument Length (LSB)
	Search Argument Descriptor(s)
0	Displacement (MSB)
1	Displacement
2	Displacement
3	Displacement (LSB)
4	Pattern Length (MSB)
5	Pattern Length (LSB)
6 _ n	Pattern

The first record offset field specifies the number of bytes that shall be ignored in the first logical block before the search begins. The value in the first record offset field shall not exceed the length of the logical block. Subsequent logical blocks shall be searched beginning with the first byte in the logical block. This permits one or more records to be skipped initially.

The number of records field specifies the maximum number of records that shall be searched by this command. An unsatisfied search shall terminate when the number of records or the number of blocks (from the command descriptor block) has been exhausted.

The search argument length specifies the length in bytes of all the search argument descriptors that follow. Since the pattern length can vary, there is no fixed multiple of the search argument descriptor to determine the search argument length.

The search argument descriptors specify one or more search conditions to execute within a single record in order to satisfy the search. Each search argument descriptor is made up of a displacement, a pattern length, and a pattern.

The displacement field specifies the displacement in bytes of the first byte of the data to be compared from the start of the logical record.

The pattern length field specifies the length in bytes of the pattern that follows.

The pattern specifies the data to compare to the logical record.

A.2.7.1 SEARCH DATA HIGH Command. The SEARCH DATA HIGH command (operation code 30H) shall be satisfied by the first logical record searched that contains data that satisfies all of the search argument descriptor(s). If the invert bit in the command descriptor block is zero, the search argument descriptor(s) shall be satisfied by data in the logical record being greater than the data in the pattern. If the invert bit is one, the search argument descriptor(s) shall be satisfied by data in the logical record being less than or equal to the data in the pattern. (See A.2.7.)

A.2.7.2 SEARCH DATA EQUAL Command. The SEARCH DATA EQUAL command (operation code 31H) shall be satisfied by the first logical record searched that contains data that satisfies all of the search argument descriptor(s). If the invert bit in the command descriptor block is zero, the search argument descriptor(s) shall be satisfied by data in the logical record being equal to the data in the pattern. If the invert bit is one, the search argument descriptor(s) shall be satisfied by data in the logical record being not equal to the data in the pattern. (See A.2.7.)

A.2.7.3 SEARCH DATA LOW Command. The SEARCH DATA LOW command (operation code 32H) shall be satisfied by the first logical record searched that contains data that satisfies all of the search argument descriptor(s). If the invert bit in the command descriptor block is zero, the search argument descriptor(s) shall be satisfied by data in the logical record being less than the data in the pattern. If the invert bit is one, the search argument descriptor(s) shall be satisfied by data in the logical record being greater than or equal to the data in the pattern. (See A.2.7.)

A.2.8 SET LIMITS Command

Peripheral Device Type:	Direct Access, Write-Once Read-Multiple, and Read-Only Direct Access
Operation Code Type:	Optional
Operation Code:	33_H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved			RdInh	WrInh
2	Logical Block Address (MSB)							
3	Logical Block Address							
4	Logical Block Address							
5	Logical Block Address (LSB)							
6	Reserved							
7	Number of Blocks (MSB)							
8	Number of Blocks (LSB)							
9	Vendor Unique		Reserved				Flag	Link

The SET LIMITS command defines the boundary outside of which any subsequent linked commands may not operate. A second SET LIMITS command may not be linked to a chain of commands in which a SET LIMITS command has already been issued.

A read inhibit (RdInh) bit of one indicates that read operations are inhibited. A write inhibit (WrInh) bit of one indicates that write operations are inhibited.

The logical block address specifies the starting address (lower boundary) for the range.

The number of blocks specifies the range in logical blocks over which the subsequent linked commands may operate. A number of blocks of zero indicates that the range shall extend to the last logical block on the logical unit.

Any attempt to access outside of the restricted range, or any attempt to perform an inhibited operation within the restricted range shall not be performed. The command shall be terminated with CHECK CONDITION status and the sense key shall be set to DATA PROTECT. A second SET LIMITS command within a linked list of commands shall be rejected with CHECK CONDITION status and the sense key shall be set to DATA PROTECT.

A.2.9 COMPARE Command

Peripheral Device Type: All
Operation Code Type: Optional
Operation Code: 39_H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Parameter List Length (MSB)							
4	Parameter List Length							
5	Parameter List Length (LSB)							
6	Reserved							
7	Reserved							
8	Reserved							
9	Vendor Unique		Reserved				Flag	Link

The COMPARE command provides the means to compare data from one logical unit with another or the same logical unit in a manner similar to the COPY command.

This command functions in the same manner as the COPY command, except that the data from the source is compared on a byte-by-byte basis with the data from the destination. The parameter list transferred to the target is the same as for the COPY command. This parameter list contains the information to identify the logical units involved in the comparison and the length of the comparison.

If the comparison is unsuccessful, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to MISCOMPARE. The remaining fields in the extended sense shall be set as documented in the COPY command.

A.2.10 COPY AND VERIFY Command

```
Peripheral Device Type:  All
   Operation Code Type:  Optional
        Operation Code:  3AH
```

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved			BytChk	Reserved
2	Reserved							
3	Parameter List Length (MSB)							
4	Parameter List Length							
5	Parameter List Length (LSB)							
6	Reserved							
7	Reserved							
8	Reserved							
9	Vendor Unique		Reserved				Flag	Link

The COPY AND VERIFY command performs the same function as the COPY command, except that a verification of the data written to the destination logical unit is performed after the data is written. The parameter list transferred to the target is the same as for the COPY command. This parameter list contains the information to identify the logical units involved in the copy and the length of the copy. (See A.1.13 for additional information about the COPY command.)

A byte check (BytChk) bit of zero causes the verification to be simply a medium verification (CRC, ECC, etc). A BytChk bit of one causes a byte-by-byte comparison of data written to the destination logical unit and the data read from the source logical unit.

If the comparison is unsuccessful, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to MISCOMPARE. The remaining fields in the extended sense shall be set as documented in the COPY command.

A.3 GROUP 0 COMMANDS FOR SEQUENTIAL-ACCESS DEVICES

A.3.1 REWIND Command

Peripheral Device Type: Sequential Access
Operation Code Type: Mandatory
Operation Code: 01_H

Bit Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				Immed
2	Reserved							
3	Reserved							
4	Reserved							
5	Vendor Unique		Reserved				Flag	Link

The REWIND command requests that the target rewind the logical unit to the beginning-of-medium or load-point.

An immediate (Immed) bit of one indicates that status shall be returned as soon as the operation is initiated. An Immed bit of zero indicates that status shall be returned after the operation is completed.

A.3.2 READ BLOCK LIMITS Command

Peripheral Device Type: Sequential Access
Operation Code Type: Extended
Operation Code: 05_H

Bit Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Reserved							
5	Vendor Unique		Reserved				Flag	Link

The READ BLOCK LIMITS command requests that the block length limits be returned for the logical unit. The READ BLOCK LIMITS data shown in Figure A-20 shall be sent during the DATA IN phase of the command.

Figure A-21: READ BLOCK LIMITS Data

Byte	Description
0	Reserved
1	Maximum Block Length (MSB)
2	Maximum Block Length
3	Maximum Block Length (LSB)
4	Minimum Block Length (MSB)
5	Minimum Block Length (LSB)

If the maximum block length equals the minimum block length, fixed-length blocks are specified. Otherwise, variable-length blocks are specified. For variable-length blocks, if the maximum block length equals zero, no upper limit is specified.

A.3.3 READ Command

Peripheral Device Type: Sequential Access
Operation Code Type: Mandatory
Operation Code: 08_H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				Fixed
2	Transfer Length (MSB)							
3	Transfer Length							
4	Transfer Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The READ command transfers one or more block(s) to the initiator beginning with the next block on the logical unit. The fixed bit specifies both the meaning of the transfer length field and whether fixed-length or variable-length block(s) are to be transferred.

If the fixed bit is zero, a single block shall be transferred with the transfer length specifying the maximum number of bytes the initiator has allocated for the returned data. If the actual block length is different from the specified transfer length, a CHECK CONDITION status shall be sent to the initiator and the incorrect length indicator (ILI) bit and valid bit in extended sense shall be set to one. The information bytes in extended sense shall be set to the difference (residue) between the requested transfer length and the actual block length. Targets that do not support negative residues shall set the ILI bit to one and the residue to zero when the actual block length is larger than the transfer length. In any case, no more than transfer length bytes shall be transferred to the initiator and the medium shall be positioned after the block (end-of-medium side).

If the fixed bit is one, the transfer length specifies the number of blocks to be transferred to the initiator. This form of the READ command is valid only if the logical unit is currently operating in fixed block mode. A logical unit is in fixed block mode when either of the following conditions is true:

(1) The logical unit reports the same value for minimum block length and maximum block length in response to the READ BLOCK LIMITS command. In this case, the current block length is the value returned.

(2) The logical unit has been instructed to use fixed-length blocks with the MODE SELECT command. In this case, the current block length is the block length defined in the MODE SELECT command.

Otherwise, the logical unit is in variable block mode. The target may implement fixed block mode, variable block mode, or both modes. If the fixed bit does not match the current mode, or the mode indicated by the fixed bit is not implemented, the target shall reject the command by returning a CHECK CONDITION status and by setting the sense key to ILLEGAL REQUEST.

A successful READ command with the fixed bit equal to one transfers the current block length times the transfer length bytes of data to the initiator. Upon termination of the READ command, the medium is positioned after the last block transferred (end-of-medium side).

If the fixed bit is one and if a block is read that is larger or smaller than the current block length, a CHECK CONDITION status shall be returned to the initiator. The ILI bit and the valid bit in extended sense shall be set to one. The information bytes shall be set to the difference (residue) between the requested transfer length and the actual number of blocks read (not including the incorrect length block). Upon termination, the medium shall be positioned after the incorrect length block (end-of-medium side).

If a logical unit reads a filemark during a READ command, it sends a CHECK CONDITION status to the initiator and sets the filemark bit in extended sense. Upon termination, the medium shall be positioned after the filemark (end-of-medium side). If the fixed bit is one, the target sets the valid bit to one and the information bytes are set to the difference (residue) between the requested transfer length and the actual number of blocks read (not including the filemark).

If a logical unit encounters the physical end-of-medium during a READ command, the target shall return a CHECK CONDITION status to the initiator and sets the end-of-medium (EOM) bit to one in extended sense. The sense key is set to MEDIUM ERROR. If the fixed bit is one, the target sets the valid bit to one and the information bytes to the difference (residue) between the requested transfer length and the actual number of blocks successfully read. The medium position following this condition is not defined.

When the transfer length is zero, no data is transferred and the current position on the logical unit does not change. This condition is not considered as an error.

A.3.4 WRITE Command

Peripheral Device Type: Sequential Access
Operation Code Type: Mandatory
Operation Code: $0A_H$

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				Fixed
2	Transfer Length (MSB)							
3	Transfer Length							
4	Transfer Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The WRITE command transfers one or more block(s) from the initiator to the current position on the logical unit. The fixed bit specifies both the meaning of the transfer length field and whether fixed-length or variable-length block(s) are to be transferred.

If the fixed bit is zero, a single block is be transferred from the initiator and written to the logical unit beginning at the current medium position. The transfer length specifies the length of the block to be written (in bytes). The requested block length shall be within the minimum and maximum block length range. If this condition is not met, a CHECK CONDITION status shall be returned and the sense key shall be set to ILLEGAL REQUEST and no data shall be written. Upon successful termination, the medium shall be positioned after the block written by this command (end-of-medium side).

If the fixed bit is one, the transfer length field specifies the number of block(s) to be transferred to the logical unit beginning at the current medium position. This form of the WRITE command is valid only if the logical unit is currently operating in fixed block mode. At the conclusion of the WRITE Command, the medium shall be positioned after the block(s) written by this command (end-of-medium side).

The target may implement fixed block mode, variable block mode, or both modes. If the fixed bit does not match the current mode, or the mode indicated by the fixed bit is not implemented, the target shall reject the command by returning a CHECK CONDITION status and by setting the sense key to ILLEGAL REQUEST.

If the early warning end-of-medium condition is encountered while writing, an attempt to finish writing any buffered data may be made. The command shall terminate with a CHECK CONDITION status and the EOM bit in extended sense shall be set to one. If any data remains in the target's buffer, then the sense key shall be set to VOLUME OVERFLOW. If the fixed bit is one and the logical unit is not buffered (buffered mode of the MODE SENSE command is zero), then the valid bit in extended sense shall be set to one and the information bytes shall be set to the difference (residue) between the requested transfer length and the actual number of blocks written to the medium. If the fixed bit is one and the logical unit is buffered (buffered mode of the MODE SENSE command is one), then the valid bit shall be set to one and the information bytes shall be set to the total number of blocks not written (the number of blocks not transferred from the initiator plus the number of blocks remaining in the target's buffer). Note that in this case it is possible for the value in the information bytes to exceed the transfer length.

When the transfer length is zero, no data is transferred and the current position on the logical unit shall not be changed. This condition is not considered as an error.

A.3.5 TRACK SELECT Command

Peripheral Device Type: Sequential Access
Operation Code Type: Optional
Operation Code: 0BH

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Track Value							
5	Vendor Unique		Reserved				Flag	Link

The TRACK SELECT command requests that the track specified in the track value field be selected.

A.3.6 READ REVERSE Command

Peripheral Device Type: Sequential Access
Operation Code Type: Optional
Operation Code: 0FH

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				Fixed
2	Transfer Length (MSB)							
3	Transfer Length							
4	Transfer Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The READ REVERSE command functions identically to the READ command except that medium motion is in the reverse direction. Thus, the block(s) and bytes within the block(s) are transferred in the reverse order and the medium position upon termination is before the last block read (beginning-of-medium side). This command terminates with a CHECK CONDITION status and the EOM bit in extended sense is set to one if beginning-of-medium or load-point is encountered. The sense key is set to NO SENSE. If the fixed bit is one, then the valid bit is set to one and the information bytes shall contain the difference (residue) of the requested transfer length and the actual number of blocks transferred before beginning-of-medium or load-point was encountered.

Filemark handling is the same as in the READ command except that the medium position upon command termination shall be before the filemark (beginning-of-medium side).

If the transfer length is zero, no data shall be transferred and the current position on the logical unit shall not be changed. This condition is not considered as an error.

The target may implement fixed block mode, variable block mode, or both modes. If the fixed bit does not match the current mode, or the mode indicated by the fixed bit is not implemented, the target shall reject the command by returning a CHECK CONDITION status and by setting the sense key to ILLEGAL REQUEST.

A.3.7 WRITE FILEMARKS Command

Peripheral Device Type: Sequential Access
Operation Code Type: Mandatory
Operation Code: 10H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Number of Filemarks (MSB)							
3	Number of Filemarks							
4	Number of Filemarks (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The WRITE FILEMARKS command causes the specified number of filemarks to be written beginning at the current medium position on the logical unit. A zero in this field indicates that no filemarks are to be written.

This command is also used to force any buffered data to be written. This command shall not return a GOOD status unless all buffered data blocks and the filemarks (if any) are correctly written on the medium.

If the early warning end-of-medium condition is encountered while writing, an attempt to finish writing any buffered data may be made. The command shall terminate with a CHECK CONDITION status and the EOM bit in extended sense shall be set to one. If any filemarks remain to be written, then the sense key shall be set to VOLUME OVERFLOW. If the logical unit is not buffered (buffered mode of the MODE SENSE command is zero), then the valid bit in extended sense shall be set to one and the information bytes shall be set to the number of unwritten filemarks. If the logical unit is buffered (buffered mode of the MODE SENSE command is one), then the valid bit shall be set to one and the information bytes shall be set to the total number of blocks not written (the number of unwritten filemarks plus the number of blocks remaining in the target's buffer). Note that in this case it is possible for the value in the information bytes to exceed the transfer length.

A.3.8 SPACE Command

Peripheral Device Type: Sequential Access
Operation Code Type: Optional
Operation Code: 11_H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved			Code	
2	Count (MSB)							
3	Count							
4	Count (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The SPACE command provides a variety of positioning functions that are determined by the code and count. Both forward (toward end-of-medium) and reverse (toward beginning-of-medium) positioning are provided, although some SCSI devices may only support a subset of this command. Such SCSI devices shall return a CHECK CONDITION status and set the sense key to ILLEGAL REQUEST in response to any attempt to invoke a function that is not supported.

The code is defined as follows:

DB(1)	DB(0)	Description
0	0	Blocks
0	1	Filemarks
1	0	Sequential Filemarks
1	1	Physical End-of-Data

When spacing over blocks or filemarks, the count field specifies the number of blocks or filemarks to be spaced over. A positive value N in the count field shall cause forward medium movement over N blocks or filemarks ending on the end-of-medium side of the last block or filemark. A zero value in the count field shall cause no medium movement. A negative value -N (2's complement notation) in the count field shall cause reverse medium movement over N blocks or filemarks ending on the beginning-of-medium side of the last block or filemark.

If a filemark is encountered while spacing over blocks, medium movement shall be stopped. The medium shall be positioned on the end-of-medium side of the filemark if movement was in the forward direction and on the beginning-of-medium side of the filemark if movement was in the reverse direction. A CHECK CONDITION status shall be sent to the initiator and the filemark and valid bits in extended sense shall be set to one. The information bytes shall be set to the difference (residue) in the requested count and the actual number of blocks spaced over (not including the filemark).

If the physical end-of-medium is encountered while spacing forward over blocks or filemarks, the target shall return a CHECK CONDITION status to the initiator and shall set the end-of-medium (EOM) bit in extended sense to one. The sense key shall be set to MEDIUM ERROR. The target shall set the valid bit to one and the information bytes to the difference (residue) between the requested count and the actual number of blocks or filemarks spaced over.

If beginning-of-medium or load-point is encountered while spacing over blocks or filemarks in the reverse direction, the target shall return a CHECK CONDITION status to the initiator and shall set the end-of-medium (EOM) bit in extended sense to one. The sense key shall be set to NO SENSE. The target shall set the valid bit to one and the information bytes to the difference (residue) between the requested count and the actual number of blocks or filemarks spaced over.

When spacing over sequential filemarks, the count field is interpreted as follows:

(1) A positive value N shall cause forward medium movement to the first occurrence of N or more consecutive filemarks stopping after the Nth filemark.

(2) A zero value shall cause no medium movement.

(3) A negative value -N (2's complement notation) shall cause reverse medium movement to the first occurrence of N or more consecutive filemarks stopping on the beginning-of-medium side of the Nth filemark.

When spacing to physical end-of-data, the count field is ignored. Forward medium movement shall occur until the logical unit encounters physical end-of-data as defined by the sequential-access device. Some sequential-access devices define physical end-of-data as an erased area on the medium; however, other definitions are not precluded. Targets that implement this function shall leave the medium positioned such that a subsequent WRITE command would append data to the last recorded information on the medium.

A.3.9 VERIFY Command

Peripheral Device Type: Sequential Access
Operation Code Type: Optional
Operation Code: 13_H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved			BytCmp	Fixed
2	Verification Length (MSB)							
3	Verification Length							
4	Verification Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The VERIFY command verifies one or more block(s) beginning with the next block on the logical unit. The fixed bit specifies both the meaning of the verification length field and whether fixed-length or variable-length block(s) are to be verified.

A byte compare (BytCmp) bit of zero indicates that the verification shall be simply a medium verification (CRC, ECC, etc). No data shall be transferred between the initiator and target. A byte compare bit of one indicates that a byte-by-byte compare of the data on the medium, and the data transferred from the initiator shall be performed by the target. Data shall be transferred from the initiator to the target as in a WRITE command.

A fixed bit of zero requests that the next block of the logical unit be verified. The verification length specifies the number of bytes to verify. A fixed bit of one requests verification length blocks be verified beginning with the next logical block on the logical unit. This form of the VERIFY command is only valid if the logical unit is currently in fixed block mode as defined in the READ command. If the data does not compare (byte compare bit equals one), the command shall terminate with a CHECK CONDITION status and the sense key shall be set to MISCOMPARE. If the fixed bit is one, the valid bit shall be set to one and the information bytes shall be set to the difference (residue) between the verification length and the actual number of blocks successfully verified. The medium shall be positioned after the block containing the miscompare (end-of-medium side).

The target may implement fixed block mode, variable block mode, or both modes. If the fixed bit does not match the current mode, or the mode indicated by the fixed bit is not implemented, the target shall reject the command by returning a CHECK CONDITION status and by setting the sense key to ILLEGAL REQUEST.

The VERIFY command shall terminate when the verification length has been satisfied, when a filemark is encountered, or when physical end-of-medium is encountered. The status and sense data for each of these conditions are handled the same as in the READ command. Upon completion of the VERIFY command, the medium shall be positioned after the last block from which data was verified or after the filemark, if encountered.

When the verification length is zero, no data is verified and the current position on the logical unit shall not be changed. This condition is not considered as an error.

A.3.10 RECOVER BUFFERED DATA Command

Peripheral Device Type:	Sequential Access
Operation Code Type:	Optional
Operation Code:	14_H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				Fixed
2	Transfer Length (MSB)							
3	Transfer Length							
4	Transfer Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The RECOVER BUFFERED DATA command is used to read data that has been transferred to an SCSI device buffer but has not been written on the medium. It is normally only used to recover from error or exception conditions that make it impossible to write the buffered data on the medium.

This command functions like the READ command except that the data is transferred from the SCSI device buffer instead of the medium. The order in which block(s) are transferred is the same as if they had been transferred to the medium. One or more RECOVER BUFFERED DATA commands may be used to read the unwritten buffered data.

The target may implement fixed block mode, variable block mode, or both modes. If the fixed bit does not match the current mode, or the mode indicated by the fixed bit is not implemented, the target shall reject the command by returning a CHECK CONDITION status and by setting the sense key to ILLEGAL REQUEST.

If an attempt is made to recover more logical blocks of data than are contained in the SCSI device buffer, the command shall be terminated with a CHECK CONDITION status. The EOM bit in extended sense shall be set to one. If the fixed bit is one, the valid bit shall be set to one and the information bytes shall be set to the difference (residue) between the requested transfer length and the actual number of blocks transferred.

The transfer length specifies the number of contiguous logical blocks of data to be transferred. A transfer length of zero indicates that no data shall be transferred. This condition shall not be considered as an error.

A.3.11 MODE SELECT Command

Peripheral Device Type: Sequential Access
Operation Code Type: Optional
Operation Code: 15H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Parameter List Length							
5	Vendor Unique		Reserved				Flag	Link

The MODE SELECT command provides a means for the initiator to specify medium, logical unit, or peripheral device parameters to the target.

The parameter list length specifies the length in bytes of the MODE SELECT parameter list that shall be transferred during the DATA OUT phase. A zero parameter list length indicates that no data shall be transferred. This condition is not considered as an error.

The MODE SELECT parameter list shown in Figure A-21 contains a four-byte header, followed by zero or more eight-byte block descriptors, followed by the vendor unique parameters, if any.

Figure A-22: MODE SELECT Parameter List

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	Reserved							
2	Reserved	Buffered Mode			Speed			
3	Block Descriptor Length							
	Block Descriptor(s)							
0	Density Code							
1	Number of Blocks (MSB)							
2	Number of Blocks							
3	Number of Blocks (LSB)							
4	Reserved							
5	Block Length (MSB)							
6	Block Length							
7	Block Length (LSB)							
	Vendor Unique Parameter(s)							
0 _ n	Vendor Unique Parameter Byte(s)							

A buffered mode of zero indicates that the target shall not report a GOOD status on WRITE commands until the data blocks are actually written on the medium. A buffered mode of one indicates that the target may report a GOOD status on WRITE commands as soon as the data block has been transferred to the SCSI device buffer. One or more blocks may be buffered prior to writing the block(s) to the medium. Buffered modes of 2H through 7H are reserved.

Code values for the speed field shall be assigned as follows:

0H	Default (Use the peripheral device's default speed).
1H	Use the peripheral device's lowest speed.
2H _ FH	Use increasing peripheral device speeds.

The block descriptor length specifies the length in bytes of all the block descriptors. It is equal to the number of block descriptors times eight and does not include the vendor unique parameters, if any. A block descriptor length of zero indicates that no block descriptors are included in the parameter list. This condition is not considered as an error.

Each block descriptor specifies the medium characteristics for all or part of a logical unit. Each block descriptor contains a density code, a number of blocks, and a block length.

Code values for the density code field shall be assigned as follows:

00H	Default (peripheral device's default density)
01H	X3.22-1983 (800 CPI, NRZI)
02H	X3.39-1973 (1600 CPI, PE)
03H	X3.54-1976 (6250 CPI, GCR)
04H	1/4 inch cartridge, QIC-11 format
05H	1/4 inch cartridge, QIC-24 format
06H	X385/85-13 (Project Number 0391-D) (3200 CPI, PE)
07H _ 7FH	Reserved
80H _ FFH	Vendor unique

The number of blocks field specifies the number of logical blocks on the medium that meet the density code and block length in the block descriptor. A number of blocks of zero indicates that all of the remaining logical blocks of the logical unit shall have the medium characteristics specified by the block descriptor.

The block length specifies the length in bytes of each logical block described by the block descriptor. A block length of zero indicates that the length shall be variable.

A.3.12 RESERVE UNIT and RELEASE UNIT Commands

Peripheral Device Type: Sequential Access
Operation Code Type: Optional
Operation Code: 16_H and 17H, respectively

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			3rdPty	Third Party Device ID			Reserved
2	Reserved							
3	Reserved							
4	Reserved							
5	Vendor Unique		Reserved				Flag	Link

The RESERVE UNIT and RELEASE UNIT commands both use the command descriptor block shown above.

A.3.12.1 RESERVE UNIT Command. The RESERVE UNIT command operation code 16_H) shall reserve the specified logical unit for the exclusive use by the requesting initiator.

The reservation remains in effect until a RELEASE UNIT command is received from the same initiator, or a BUS DEVICE RESET message is received from any initiator, or a "hard" RESET condition occurs. The occurrence of the last two conditions is indicated by a sense key of UNIT ATTENTION on the next command following the condition. It is not an error to issue this command to a logical unit that is currently reserved to the requesting initiator.

If the logical unit is previously reserved for another initiator, then the target shall respond by either:

(1) returning a RESERVATION CONFLICT status; or

(2) queuing the reservation request and then disconnecting until all previously queued reservations have been released and the logical unit is available, then reconnecting to perform the reservation.

If, after honoring the reservation, any other initiator then subsequently attempts to perform any command on the reserved logical unit other than a RESERVE UNIT command, which may be queued, then the command shall be rejected with a RESERVATION CONFLICT status.

The third-party reservation option for the RESERVE UNIT command allows an initiator to reserve a logical unit for another SCSI device. This option is intended for use in multiple-initiator systems that use the COPY command. Any target that implements the third-party reservation option shall also implement the third-party release option.

If the third-party (3rdPty) bit is zero, then the third-party reservation option is not requested. If the 3rdPty bit is one and the third-party reservation option is implemented, then the RESERVE UNIT command shall reserve the specified logical unit for the SCSI device specified in the third-party device ID field.

A.3.12.2 RELEASE UNIT Command. The RELEASE UNIT command operation code 17_H) releases the logical unit if it is currently reserved to the requesting initiator.

It is not an error to attempt to release a logical unit that is not currently reserved to the requesting initiator. However, it shall not be released if it is reserved by another initiator.

If the 3rdPty bit is one and the target does not implement the third-party release option, then the target shall terminate the command with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

A.3.13 ERASE Command

Peripheral Device Type: Sequential Access
Operation Code Type: Optional
Operation Code: 19_H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				Long
2	Reserved							
3	Reserved							
4	Reserved							
5	Vendor Unique		Reserved				Flag	Link

The ERASE command causes part or all of the remaining medium to be erased beginning from the current medium position. As used here, "erased" means either the medium shall be erased or a pattern shall be written on the medium that appears as gap to the target.

The distance to be erased is controlled by the long bit. A long bit of one indicates that all remaining medium on the logical unit shall be erased. A long bit of zero indicates that a peripheral device specified portion of the medium shall be erased. Normally, short erases are used to create an extended gap for software controlled error recovery or for support of "update in place" functions. The medium position following an ERASE command with a long bit of one is not defined by the standard.

NOTE: Some targets may reject ERASE commands with the long bit set to one if the medium is not positioned at the beginning-of-medium.

A.3.14 MODE SENSE Command

Peripheral Device Type: Sequential Access
Operation Code Type: Optional
Operation Code: $1A_H$

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Allocation Length							
5	Vendor Unique		Reserved				Flag	Link

The MODE SENSE command provides a means for a target to report its medium, logical unit, or peripheral device parameters to the initiator. It is a complementary command to the MODE SELECT command (see A.3.11) for support of a medium that may contain different densities, such as half-inch tapes.

The allocation length specifies the number of bytes that the initiator has allocated for returned MODE SENSE data. An allocation length of zero indicates that no MODE SENSE data shall be transferred. This condition shall not be considered as an error. Any other value indicates the maximum number of bytes that shall be transferred. The target shall terminate the DATA IN phase when allocation length bytes have been transferred or when all available MODE SENSE data have been transferred to the initiator, whichever is less.

The MODE SENSE data, shown in Figure A-22, contains a four-byte header, followed by zero or more eight-byte block descriptors, followed by the vendor unique parameters, if any.

Figure A-23 MODE SENSE Data

Bit / Byte	7	6	5	4	3	2	1	0
0	Sense Data Length							
1	Medium Type							
2	WP	Buffered Mode			Speed			
3	Block Descriptor Length							
	Block Descriptor(s)							
0	Density Code							
1	Number of Blocks (MSB)							
2	Number of Blocks							
3	Number of Blocks (LSB)							
4	Reserved							
5	Block Length (MSB)							
6	Block Length							
7	Block Length (LSB)							
	Vendor Unique Parameter(s)							
0 _ n	Vendor Unique Parameter Byte(s)							

The sense data length specifies the length in bytes of the following mode sense data that is available to be transferred during the DATA IN phase. The sense data length does not include itself.

Code values for the medium type field shall be assigned as follows:

00H	Default (only one medium type supported)
01H _ 7FH	Reserved
80H _ FFH	Vendor unique

A write protected (WP) bit of zero indicates that the medium is write enabled. A WP bit of one indicates that the medium is write protected.

A buffered mode of zero indicates that the target does not report a GOOD status on WRITE commands until the data blocks are actually written on the medium. A buffered mode of one indicates that the target may report a GOOD status on WRITE commands as soon as the data block has been transferred to the SCSI device buffer. One or more blocks may be buffered prior to writing the block(s) to the medium. Buffered modes of 2H through 7H are reserved.

Code values for the speed field shall be assigned as follows:

0H	Default (only one speed supported)
1H	Lowest peripheral device speed
2H _ FH	Increasing peripheral device speeds

The block descriptor length specifies the length in bytes of all the block descriptors. It is equal to the number of block descriptors times eight and does not include the vendor unique parameters, if any. A block descriptor length of zero indicates that no block descriptors shall be included in the parameter list. This condition shall not be considered as an error.

Each block descriptor specifies the medium characteristics for all or part of a logical unit. Each block descriptor contains a density code, a number of blocks, and a block length.

Code values for the density code field shall be assigned as follows:

00H	Default (peripheral device's default density)
01H	X3.22-1983 (800 CPI, NRZI)
02H	X3.39-1973 (1600 CPI, PE)
03H	X3.54-1976 (6250 CPI, GCR)
04H	1/4 inch cartridge, QIC-11 format
05H	1/4 inch cartridge, QIC-24 format
06H	X385/85-13 (Project Number 0391-D) (3200 CPI, PE)
07H _ 7FH	Reserved
80H _ FFH	Vendor unique

The number of blocks field specifies the number of logical blocks on the medium that meet the density code and block length in the block descriptor. A number of blocks of zero indicates that all of the remaining logical blocks of the logical unit have the medium characteristics specified by the block descriptor.

The block length specifies the length in bytes of each logical block described by the block descriptor. A block length of zero indicates that the length is variable.

A.3.15 LOAD/UNLOAD Command

Peripheral Device Type:	Sequential Access
Operation Code Type:	Optional
Operation Code:	$1B_H$

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				Immed
2	Reserved							
3	Reserved							
4	Reserved						Re-Ten	Load
5	Vendor Unique		Reserved				Flag	Link

The LOAD/UNLOAD command requests that the target enable or disable the logical unit for further operations. This command may also be used to request the retension function on peripheral devices that support this function.

A load bit of one indicates that the medium on the logical unit shall be loaded and positioned to the beginning-of-medium or load-point as determined by the peripheral device. A load bit of zero indicates that the medium on the logical unit shall be positioned for removal from the peripheral device.

Status shall be returned after the medium is positioned unless the immediate (Immed) bit is one. If the Immed bit is one, status may be returned as soon as the command has been accepted.

A re-tension (Re-Ten) bit of one indicates that the medium on the addressed logical unit shall be correctly tensioned before the LOAD/UNLOAD command is completed. This is an optional function intended for use by those peripheral devices that support the re-tension function.

A.3.16 PREVENT/ALLOW MEDIUM REMOVAL Command

Peripheral Device Type:	Sequential Access
Operation Code Type:	Optional
Operation Code:	1EH

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Reserved							Prevent
5	Vendor Unique		Reserved				Flag	Link

The PREVENT/ALLOW MEDIUM REMOVAL command requests that the target enable or disable the removal of the medium in the logical unit.

A prevent bit of one shall inhibit mechanisms that normally allow removal of the medium. A prevent bit of zero shall allow removal of the medium.

This prevention of medium removal condition shall terminate upon receipt of a PREVENT/ALLOW MEDIUM REMOVAL command with the prevent bit set to zero, or by the receipt of a BUS DEVICE RESET message from any initiator or by a "hard" RESET condition.

A.4 GROUP 0 COMMANDS FOR PRINTER DEVICES

A.4.1 FORMAT Command

Peripheral Device Type: Printer
Operation Code Type: Optional
Operation Code: 04_H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved			Format Type	
2	Transfer Length (MSB)							
3	Transfer Length							
4	Transfer Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The FORMAT command provides a means for the initiator to specify forms or fonts to printers that support programmable forms or fonts. The format information sent is vendor unique since it is peripheral-device specific.

The format type field specifies the type of format information to be transferred from the initiator to the target. This field is defined as follows:

DB(1)	DB(0)	Format Type
0	0	Set Form
0	1	Set Font
1	0	Vendor Unique
1	1	Reserved

The transfer length specifies the length in bytes of format information that shall be sent during the DATA OUT phase. A transfer length of zero indicates that no format information shall be sent. This condition shall not be considered as an error.

A.4.2 PRINT Command

Peripheral Device Type: Printer
Operation Code Type: Mandatory
Operation Code: $0A_H$

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Transfer Length (MSB)							
3	Transfer Length							
4	Transfer Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The PRINT command transfers the specified number of bytes from the initiator to the target to be printed. The data sent is application dependent.

The transfer length specifies the length in bytes of data that shall be sent during the DATA OUT phase. A transfer length of zero indicates that no data shall be sent. This condition is not considered as an error.

A.4.3 SLEW AND PRINT Command

Peripheral Device Type: Printer
Operation Code Type: Optional
Operation Code: $0B_H$

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				Channel
2	Slew Value							
3	Transfer Length (MSB)							
4	Transfer Length (LSB)							
5	Vendor Unique		.Reserved				Flag	Link

The SLEW AND PRINT command transfers the specified number of bytes from the initiator to the target to be printed. The data sent is application dependent. This command is provided for printers that do not support forms control information imbedded within the print data.

The transfer length specifies the length in bytes of data that shall be sent during the DATA OUT phase. A transfer length of zero indicates that no data shall be sent. This condition shall not be considered as an error.

If the channel bit is zero, the slew value specifies the number of lines the form shall be advanced before printing. A value of 255 indicates that the form shall be advanced to the first line of the next form before printing. If the channel bit is one, the slew value specifies the forms control channel number to which the form shall be advanced prior to printing the data.

If the channel bit is one, and the channel option is not implemented, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

A.4.4 FLUSH BUFFER Command

Peripheral Device Type:	Printer
Operation Code Type:	Optional
Operation Code:	10H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Reserved							
5	Vendor Unique		Reserved				Flag	Link

The FLUSH BUFFER command provides a means for an initiator to ensure that the data have been successfully printed prior to releasing the peripheral device. This is useful for applications that wish to handle any error or exception conditions (e.g., end-of-medium) prior to termination of the application.

When all buffered data are actually printed the command shall be terminated with a GOOD status. If it is not possible to finish printing all of the buffered data (due to an error or exception condition on the peripheral device), then this command shall be terminated with a CHECK CONDITION status and the appropriate sense key.

A.4.5 RECOVER BUFFERED DATA Command

Peripheral Device Type: Printer
Operation Code Type: Optional
Operation Code: 14_H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Transfer Length (MSB)							
3	Transfer Length							
4	Transfer Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The RECOVER BUFFERED DATA command returns to the initiator the data that has been sent to the target, but not yet printed.

This command is normally used only to recover from error or exception conditions that make it impossible to print the buffered data. The order in which the data is transferred from the target to the initiator is the same as it was when the data was previously transferred using the PRINT command or SLEW AND PRINT command. Data that is transferred by this command is deleted from the target data buffer. One or more RECOVER BUFFERED DATA commands may be used to return the unprinted buffered data.

If an attempt is made to recover more data than is contained in the buffer, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to NO SENSE. In addition, the EOM, the valid, and the ILI bits in extended sense shall be set to one. The information bytes shall be set to the difference (residue) between the transfer length and the actual number of bytes returned.

The transfer length specifies the maximum length in bytes of data that shall be transferred during the DATA IN phase. A transfer length of zero indicates that no data shall be transferred. This condition is not considered as an error.

A.4.6 MODE SELECT Command

Peripheral Device Type: Printer
Operation Code Type: Optional
Operation Code: 15H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Parameter List Length							
5	Vendor Unique		Reserved				Flag	Link

The MODE SELECT command provides a means for the initiator to specify medium, logical unit, or peripheral device parameters to the target.

The parameter list length specifies the length in bytes of the MODE SELECT parameter list that shall be transferred during the DATA OUT phase. A parameter list length of zero indicates that no data shall be transferred. This condition shall not be considered as an error. The MODE SELECT parameter list (Figure A-23) contains a four-byte header, followed by the vendor unique parameters, if any.

Figure A-24: MODE SELECT Parameter List

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	Reserved							
2	Reserved	Buffered Mode			Reserved			
3	Reserved							
	Vendor Unique Parameter(s)							
0 _ n	Vendor Unique Parameter Byte(s)							

A buffered mode of zero indicates that the target shall not report a GOOD status on PRINT commands or SLEW AND PRINT commands until the data are actually printed. A buffered mode of one indicates that the target may report a GOOD status on PRINT commands or SLEW AND PRINT commands as soon as the data have been transferred to the SCSI device buffer. The data from one or more commands may be buffered prior to printing. Buffered modes of 2_H through 7H are reserved.

A.4.7 RESERVE UNIT and RELEASE UNIT Commands

Peripheral Device Type:	Printer
Operation Code Type:	Optional
Operation Code:	16H and 17HG, respectively

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			3rdPty	Third Party Device ID			Reserved
2	Reserved							
3	Reserved							
4	Reserved							
5	Vendor Unique		Reserved				Flag	Link

The RESERVE UNIT and RELEASE UNIT commands both use the command descriptor block shown above.

A.4.7.1 RESERVE UNIT Command. The RESERVE UNIT command (operation code 16H) shall reserve the specified logical unit for the exclusive use by the requesting initiator.

The reservation shall remain in effect until a RELEASE UNIT command is received from the same initiator, or a BUS DEVICE RESET message is received from any initiator, or a "hard" RESET condition occurs. The occurrence of the last two conditions is indicated by a sense key of UNIT ATTENTION on the next command following the condition. It is not an error to issue this command to a logical unit that is currently reserved to the requesting initiator.

If the logical unit is previously reserved for another initiator, then the target shall respond by either:

(1) returning a RESERVATION CONFLICT status; or

(2) queuing the reservation request and then disconnecting until all previously queued reservations have been released and the logical unit is available, then reconnecting to perform the reservation.

If, after honoring the reservation, any other initiator then subsequently attempts to perform any command on the reserved logical unit other than a RESERVE UNIT command, which may be queued, then the command shall be rejected with RESERVATION CONFLICT status.

The third-party reservation option for the RESERVE UNIT command allows an initiator to reserve a logical unit for another SCSI device. This option is intended for use in multiple-initiator systems that use the COPY command. Any target that implements the third-party reservation option shall also implement the third-party release option (see 10.7.2).

If the third-party (3rdPty) bit is zero, then the third-party reservation option is not requested. If the 3rdPty bit is one and the third-party reservation option is implemented, then the RESERVE UNIT command shall reserve the specified logical unit for the SCSI device specified in the third-party device ID field. The target shall preserve the reservation until it is released by the same initiator (or by a BUS DEVICE RESET message from any initiator or a "hard" RESET condition). The target shall ignore any attempt to release the reservation made by any other initiator.

If the 3rdPty bit is one and the third-party reservation option is not implemented, then the target shall reject the RESERVE UNIT command with a CHECK CONDITION status and a sense key of ILLEGAL REQUEST.

A.4.7.2 RELEASE UNIT Command. The RELEASE UNIT command (operation code 17H) shall release the logical unit if it is currently reserved to the requesting initiator.

It is not an error to attempt to release a logical unit that is not currently reserved to the requesting initiator. However, it shall not be released if it is reserved by another initiator.

The third-party release option for the RELEASE UNIT command allows an initiator to release a logical unit that was previously reserved using the third-party reservation option (see 10.7.1). This option shall be implemented if the third-party reservation option is implemented. This option is intended for use in multiple-initiator systems that use the COPY command.

If the third-party (3rdPty) bit is zero, then the third-party release option is not requested. If the 3rdPty bit is one and the target implements the third-party release option, then the target shall release the specified logical unit, but only if the reservation was made using the third-party reservation option by the same initiator for the same SCSI device as specified in the third-party device ID field.

If the 3rdPty bit is one and the target does not implement the third-party release option, then the target shall terminate the command with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

A.4.8 MODE SENSE Command

Peripheral Device Type: Printer
Operation Code Type: Optional
Operation Code: $1A_H$

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Allocation Length							
5	Vendor Unique		Reserved				Flag	Link

The MODE SENSE command provides a means for a target to report its medium, logical unit, or peripheral device parameters to the initiator. It is a complementary command to the MODE SELECT command.

The allocation length specifies the number of bytes that the initiator has allocated for returned MODE SENSE data. An allocation length of zero indicates that no MODE SENSE data shall be transferred. This condition shall not be considered as an error. Any other value indicates the maximum number of bytes that shall be transferred. The target shall terminate the DATA IN phase when allocation length bytes have been transferred or when all available MODE SENSE data have been transferred to the initiator, whichever is less.

The MODE SENSE data (Figure A-25) contains a four-byte header, followed by the vendor unique parameters, if any.

Figure A-25: MODE SENSE Data

Bit / Byte	7	6	5	4	3	2	1	0
0	Sense Data Length							
1	Reserved							
2	Reserved	Buffered Mode			Reserved			
3	Reserved							
	Vendor Unique Parameter(s)							
0 _ n	Vendor Unique Parameter Byte(s)							

The sense data length specifies the length in bytes of the following MODE SENSE data that is available to be transferred during the DATA IN phase. The sense data length does not include itself.

A buffered mode of zero indicates that the target does not report a GOOD status on PRINT commands or SLEW AND PRINT commands until the data are actually printed. A buffered mode of one indicates that the target may report a GOOD status on PRINT commands or SLEW AND PRINT commands as soon as the data have been transferred to the SCSI device buffer. The data from one or more commands may be buffered prior to printing. Buffered modes of 2H through 7H are reserved.

A.4.9 STOP PRINT Command

Peripheral Device Type:	Printer
Operation Code Type:	Optional
Operation Code:	$1B_H$

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				Retain
2	Vendor Unique							
3	Reserved							
4	Reserved							
5	Vendor Unique		Reserved				Flag	Link

The STOP PRINT command is used to halt printing on buffered devices in an orderly fashion.

A retain bit of zero requests that the target data buffer be discarded; otherwise, the unprinted data is retained. The unprinted data may be recovered by use of the RECOVER BUFFERED DATA command, if supported. A subsequent PRINT command or SLEW AND PRINT command shall cause the remaining unprinted and unrecovered data to be printed followed by the data transferred by the subsequent command. The point at which printing is suspended by this command is peripheral-device specific and is not defined by this standard.

A.5 GROUP 0 COMMANDS FOR PROCESSOR DEVICES

A.5.1 RECEIVE Command

Peripheral Device Type:	Processor Devices
Operation Code Type:	Optional
Operation Code:	08_H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Allocation Length (MSB)							
3	Allocation Length							
4	Allocation Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The RECEIVE command transfers data from the target to the initiator.

The allocation length specifies the number of bytes that the initiator has allocated for the returned data. An allocation length of zero indicates that no data shall be transferred. This condition shall not be considered as an error. Any other value indicates the maximum number of bytes that shall be transferred. The target shall terminate the DATA IN phase when allocation length bytes have been transferred or when all available data have been transferred to the initiator, whichever is less.

A.5.2 SEND Command

Peripheral Device Type: Processor Devices
Operation Code Type: Mandatory
Operation Code: $0A_H$

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Transfer Length (MSB)							
3	Transfer Length							
4	Transfer Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The SEND command transfers data from the initiator to the target.

The transfer length specifies the length in bytes of data that shall be sent during the DATA OUT phase. A transfer length of zero indicates that no data shall be sent. This condition shall not be considered as an error.

A.6 GROUP 0 COMMANDS FOR WRITE-ONCE READ-MULTIPLE DEVICES

A.6.1 READ Command

Peripheral Device Type:	Write-Once Read-Multiple and Read-Only Direct Access
Operation Code Type:	Optional
Operation Code:	08_H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Logical Block Address (MSB)				
2	Logical Block Address							
3	Logical Block Address (LSB)							
4	Transfer Length							
5	Vendor Unique		Reserved				Flag	Link

The READ command requests that the target transfer data to the initiator.

The logical block address specifies the logical block at which the read operation shall begin.

The transfer length specifies the number of contiguous logical blocks of data to be transferred. A Transfer Length of zero indicates that 256 logical blocks shall be transferred. Any other value indicates the number of logical blocks that shall be transferred.

This command shall be terminated with a status of RESERVATION CONFLICT if any reservation access conflict exists and no data shall be transferred.

If any of the following conditions occur, this command shall be terminated with a CHECK CONDITION status, and if extended sense is implemented, the sense key shall be set as indicated in the following table. This table does not provide an exhaustive enumeration of all conditions that may cause the CHECK CONDITION status.

Condition	Sense Key
Invalid logical block address	ILLEGAL REQUEST (see note 1)
Target reset or medium change since the last command from this initiator	UNIT ATTENTION
Unrecoverable read error	MEDIUM ERROR
Overrun or other error that might be resolved by repeating the command	ABORTED COMMAND
Attempt to read a blank or previously unwritten block	BLANK CHECK (see note 2)

NOTES:
(1) The extended sense information bytes shall be set to the logical block address of the first invalid address.

(2) The extended sense information bytes shall be set to the logical block address of the first blank block encountered. The data read up to that block shall be transferred.

A.6.2 WRITE Command

Peripheral Device Type: Write-Once Read-Multiple
Operation Code Type: Optional
Operation Code: $0A_H$

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Logical Block Address (MSB)				
2	Logical Block Address							
3	Logical Block Address (LSB)							
4	Transfer Length							
5	Vendor Unique		Reserved				Flag	Link

The WRITE command requests that the target write the data transferred from the initiator to the medium.

The logical block address specifies the logical block at which the write operation shall begin.

The transfer length specifies the number of contiguous logical blocks of data that shall be written. A transfer length of zero indicates that 256 logical blocks shall be written. Any other value indicates the number of logical blocks that shall be written.

This command shall be terminated with a status of RESERVATION CONFLICT if any reservation access conflict exists and no data shall be written.

If any of the following conditions occur, this command shall be terminated with a CHECK CONDITION status, and if extended sense is implemented, the sense key shall be set as indicated in the following table. This table does not provide an exhaustive enumeration of all conditions that may cause the CHECK CONDITION status.

Condition	Sense Key
Invalid logical block address	ILLEGAL REQUEST (see note 1)
Target reset or medium change since the last command from this initiator	UNIT ATTENTION
Overrun or other error that might be resolved by repeating the command	ABORTED COMMAND
Attempt to write a previously written block and blank checking is enabled	BLANK CHECK (see note 2)

NOTES:
(1) The extended sense information bytes shall be set to the logical block address of the first invalid address.

(2) The extended sense information bytes shall be set to the logical block address of the first non-blank block encountered.

A.6.3 MODE SELECT Command

Peripheral Device Type:	Write-Once Read-Multiple and Read-Only Direct Access
Operation Code Type:	Optional
Operation Code:	15_H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Parameter List Length							
5	Vendor Unique		Reserved				Flag	Link

The MODE SELECT command provides a means for the initiator to specify medium, logical unit, or peripheral device parameters to the target.

The parameter list length specifies the length in bytes of the MODE SELECT parameter list that shall be transferred from the initiator to the target. A parameter list length of zero indicates that no data shall be transferred. This condition shall not be considered as an error.

The MODE SELECT parameter list (Figure A-25) contains a four-byte header, followed by zero or more eight-byte block descriptors, followed by the vendor unique parameters, if any.

Figure A-26 MODE SELECT Parameter List

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	Reserved							
2	Reserved							EBC
3	Block Descriptor Length							
	Block Descriptor(s)							
0	Reserved							
1	Number of Blocks (MSB)							
2	Number of Blocks							
3	Number of Blocks (LSB)							
4	Reserved							
5	Block Length (MSB)							
6	Block Length							
7	Block Length (LSB)							
	Vendor Unique Parameter(s)							
0 _ n	Vendor Unique Parameter Byte(s)							

An enable blank check (EBC) bit of zero disables blank checking of the medium during write operations. An EBC bit of one enables blank checking. If a non-blank block is found during a write operation, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to BLANK CHECK. For read-only direct-access devices, the EBC bit is reserved.

The block descriptor length specifies the length in bytes of all the block descriptors. It is equal to the number of block descriptors times eight and does not include the vendor unique parameters, if any. A block descriptor length of zero indicates that no block descriptors shall be included in the parameter list. This condition shall not be considered as an error.

Each block descriptor specifies the medium characteristics for all or part of a logical unit. Each block descriptor contains a number of blocks and a block length. The number of blocks field specifies the number of logical blocks to be formatted with the block length specified in the block descriptor. The block length field specifies the length in bytes of the logical block to be formatted.

A.6.4 MODE SENSE Command

Peripheral Device Type:	Write-Once Read-Multiple and Read-Only Direct Access
Operation Code Type:	Optional
Operation Code:	$1A_H$

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Allocation Length							
5	Vendor Unique		Reserved				Flag	Link

The MODE SENSE command provides a means for a target to report its medium, logical unit, or peripheral device parameters to the initiator. It is a complementary command to the MODE SELECT command for support of medium that may contain multiple block lengths.

The allocation length specifies the number of bytes that the initiator has allocated for returned MODE SENSE Data. An allocation length of zero indicates that no MODE SENSE data shall be transferred. This condition shall not be considered as an error. Any other value indicates the maximum number of bytes that shall be transferred. The target shall terminate the DATA IN phase when allocation length bytes have been transferred or when all available MODE SENSE data have been transferred to the initiator, whichever is less.

The MODE SENSE data (Figure A-27) contains a four-byte header, followed by zero or more eight-byte block descriptors, followed by the vendor unique parameters, if any.

Figure A-27: MODE SENSE Data

Bit / Byte	7	6	5	4	3	2	1	0
0	Sense Data Length							
1	Medium Type							
2	WP	Reserved						EBC
3	Block Descriptor Length							
	Block Descriptor(s)							
0	Reserved							
1	Number of Blocks (MSB)							
2	Number of Blocks							
3	Number of Blocks (LSB)							
4	Reserved							
5	Block Length (MSB)							
6	Block Length							
7	Block Length (LSB)							
	Vendor Unique Parameter(s)							
0 _ n	Vendor Unique Parameter Byte(s)							

The sense data length specifies the length in bytes of the following MODE SENSE data that is available to be transferred during the DATA IN phase. The sense data length does not include itself.

Code values for the medium type field shall be assigned as follows:

00H	Default (only one medium type supported)
01H _ 7FH	Reserved
80H _ FFH	Vendor unique

An enable blank check (EBC) bit of zero indicates that blank checking of the medium during write operations is disabled. An EBC bit of one indicates that blank checking during write operations is enabled. For read-only direct-access devices, the EBC bit is reserved.

A write protected (WP) bit of zero indicates that the medium is write enabled. A WP bit of one indicates that the medium is write protected. For read-only direct-access devices, the WP bit is reserved.

The block descriptor length specifies the length in bytes of all the block descriptors. It is equal to the number of block descriptors times eight and does not include the vendor unique parameters, if any. A block descriptor length of zero indicates that no block descriptors shall be included in the parameter list. This condition shall not be considered as an error.

Each block descriptor specifies the medium characteristics for all or part of a logical unit. Each block descriptor contains a number of blocks and a block length.

The number of blocks field indicates the number of logical blocks that have the block length specified in the block descriptor. The block length field indicates the length in bytes of each logical block.

A.7 GROUP 1 COMMANDS FOR WRITE-ONCE READ-MULTIPLE DEVICES

A.7.1 READ Command

Peripheral Device Type:	Write-Once Read-Multiple and Read-Only Direct Access
Operation Code Type:	Mandatory
Operation Code:	28_H

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				RelAdr
2	Logical Block Address (MSB)							
3	Logical Block Address							
4	Logical Block Address							
5	Logical Block Address (LSB)							
6	Reserved							
7	Transfer Length (MSB)							
8	Transfer Length (LSB)							
9	Vendor Unique		Reserved				Flag	Link

The READ command requests that the target transfer data to the initiator from the medium.

The logical block address specifies the logical block at which the read operation shall begin.

The transfer length specifies the number of contiguous logical blocks of data that shall be transferred. A transfer length of zero indicates that no data shall be transferred. This condition shall not be considered as an error. Any other value indicates the number of logical blocks that shall be transferred.

This command shall be terminated with a status of RESERVATION CONFLICT if any reservation access conflict exists and no data shall be transferred.

If any of the following conditions occur, this command shall be terminated with a CHECK CONDITION status and, if extended sense is implemented, the sense key shall be set as indicated in the following table. This table does not provide an exhaustive enumeration of all conditions that may cause the CHECK CONDITION status.

Condition	Sense Key
Invalid logical block address	ILLEGAL REQUEST (see note 1)
Target reset or medium change since the last command from this initiator	UNIT ATTENTION
Overrun or other error that might be resolved by repeating the command	ABORTED COMMAND
Attempt to read a blank or previously unwritten block	BLANK CHECK (see note 2)

NOTES:
(1) The extended sense information bytes shall be set to the logical block address of the first invalid address.

(2) The extended sense information bytes shall be set to the logical block address of the first blank block encountered. The data read up to that block shall be transferred.

A.7.2 WRITE Command

Peripheral Device Type: Write-Once Read-Multiple
Operation Code Type: Mandatory
Operation Code: $2A_H$

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				RelAdr
2	Logical Block Address (MSB)							
3	Logical Block Address							
4	Logical Block Address							
5	Logical Block Address (LSB)							
6	Reserved							
7	Transfer Length (MSB)							
8	Transfer Length (LSB)							
9	Vendor Unique		Reserved				Flag	Link

The WRITE command requests that the target write the data transferred from the initiator to the medium.

The logical block address specifies the logical block at which the write operation shall begin.

The transfer length specifies the number of contiguous logical blocks of data that shall be transferred. A transfer length of zero indicates that no data shall be transferred. This condition shall not be considered as an error and no data shall be written. Any other value indicates the number of logical blocks that shall be transferred.

This command shall be terminated with a status of RESERVATION CONFLICT if any reservation access conflict exists and no data shall be written.

If any of the following conditions occur, this command shall be terminated with a CHECK CONDITION status and, if extended sense is implemented, the sense key shall be set as indicated in the following table. This table does not provide an exhaustive enumeration of all conditions that may cause the CHECK CONDITION status.

Condition	Sense Key
Invalid logical block address	ILLEGAL REQUEST (see note 1)
Target reset or medium change since the last command from this initiator	UNIT ATTENTION
Overrun or other error that might be resolved by repeating the command	ABORTED COMMAND
Attempt to write a previously written block and blank checking is enabled (see 12.1.3)	BLANK CHECK (see note 2)

NOTES:
(1) The extended sense information bytes shall be set to the logical block address of the first invalid address.

(2) The extended sense information bytes shall be set to the logical block address of the first non-blank block encountered.

A.7.3 WRITE AND VERIFY Command

Peripheral Device Type: Write-Once Read-Multiple
Operation Code Type: Optional
Operation Code: $2E_H$

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved			BytChk	RelAdr
2	Logical Block Address (MSB)							
3	Logical Block Address							
4	Logical Block Address							
5	Logical Block Address (LSB)							
6	Reserved							
7	Transfer Length (MSB)							
8	Transfer Length (LSB)							
9	Vendor Unique		Reserved				Flag	Link

The WRITE AND VERIFY command requests that the target write the data transferred from the initiator to the medium and then verify that the data is correctly written.

A byte check (BytChk) bit of zero causes the verification to be simply a medium verification (CRC, ECC, etc). A BytChk bit of one causes a byte-by-byte compare of data written to the peripheral device and the data transferred from the initiator. If the compare is unsuccessful, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to MISCOMPARE.

The logical block address specifies the logical block at which the write operation shall begin.

The transfer length specifies the number of contiguous logical blocks of data that shall be transferred. A transfer length of zero indicates that no logical blocks shall be transferred. This condition shall not be considered as an error and no data shall be written. Any other value indicates the number of logical blocks that shall be transferred.

A.7.4 VERIFY Command

Peripheral Device Type: Write-Once Read-Multiple and Read-Only Direct Access
Operation Code Type: Optional
Operation Code: $2F_H$

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved		BlkVfy	BytChk	RelAdr
2	Logical Block Address (MSB)							
3	Logical Block Address							
4	Logical Block Address							
5	Logical Block Address (LSB)							
6	Reserved							
7	Verification Length (MSB)							
8	Verification Length (LSB)							
9	Vendor Unique		Reserved				Flag	Link

The VERIFY command requests that the target verify the data on the medium.

A byte check (BytChk) bit of zero causes the verification to be simply a medium verification (CRC, ECC, etc). A BytChk bit of one causes a byte-by-byte compare of the data on the medium and the data transferred from the initiator. The data shall be transferred as it would be for a WRITE command. If the compare is unsuccessful, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to MISCOMPARE.

A blank verify (BlkVfy) bit of one causes a verification that the blocks are blank.

The logical block address specifies the logical block at which the verify operation shall begin.

The verification length specifies the number of contiguous logical blocks of data or blanks that shall be verified. A verification length of zero indicates that no logical blocks shall be verified. This condition shall not be considered as an error. Any other value indicates the number of logical blocks that shall be verified.

A.8 GROUP 0 COMMANDS FOR READ-ONLY DIRECT ACCESS DEVICES

All group 0 commands for Read-Only Direct-Access devices are included in sections A.1 and A.6.

A.9 GROUP 1 COMMANDS FOR READ-ONLY DIRECT-ACCESS DEVICES

All group 1 commands for Read-Only Direct-Access devices are included in sections A.2 and A.7.

APPENDIX B

RETURN STATUS BLOCK

and

SENSE DATA BLOCK

DEFINITION

The Return Status Block and Sense Data Block presented in this appendix are taken directly from ANSI working document ANSI X3T9/84-40 REV 15, dated May 8, 1985.

B.1 STATUS BYTE

A status byte shall be sent from the target to the initiator during the STATUS phase at the termination of each command as specified in Figures B-1 and B-2 unless the command is cleared by an ABORT message, by a BUS DEVICE RESET message, or by a "hard" RESET condition.

Figure B-1: Status Byte

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved	Vendor Unique		Status Byte Code				V

Figure B-2: Status Byte Code Bit Values

Bits of Status Byte								
7	6	5	4	3	2	1	0	Status(es) Represented
R	V	V	0	0	0	0	V	GOOD
R	V	V	0	0	0	1	V	CHECK CONDITION
R	V	V	0	0	1	0	V	CONDITION MET/GOOD
R	V	V	0	0	1	1	V	Reserved
R	V	V	0	1	0	0	V	BUSY
R	V	V	0	1	0	1	V	Reserved
R	V	V	0	1	1	0	V	Reserved
R	V	V	0	1	1	1	V	Reserved
R	V	V	1	0	0	0	V	INTERMEDIATE/GOOD
R	V	V	1	0	0	1	V	Reserved
R	V	V	1	0	1	0	V	INTERMEDIATE/CONDITION MET/GOOD
R	V	V	1	0	1	1	V	Reserved
R	V	V	1	1	0	0	V	RESERVATION CONFLICT
R	V	V	1	1	0	1	V	Reserved
R	V	V	1	1	1	0	V	Reserved
R	V	V	1	1	1	1	V	Reserved

Key: R - Reserved bit
V - Vendor unique bit

APPENDIX B

A description of the status byte codes is given below:

GOOD. This status indicates that the target has successfully completed the command.

CHECK CONDITION. Any error, exception, or abnormal condition that causes sense data to be set, shall cause a CHECK CONDITION status. The REQUEST SENSE command should be issued following a CHECK CONDITION status, to determine the nature of the condition.

CONDITION MET. The SEARCH DATA commands shall send this status whenever a search condition is satisfied. This status does not break a chain of linked commands. The logical block address of the logical block that satisfies the search may be determined with a REQUEST SENSE command.

BUSY. The target is busy. This status shall be sent whenever a target is unable to accept a command from an initiator.

INTERMEDIATE. This status shall be sent for every command in a series of linked commands (except the last command), unless an error, exception, or abnormal condition causes a CHECK CONDITION status or a RESERVATION CONFLICT status to be set. If this status is not sent, the chain of linked commands is broken; no further commands in the series are executed.

RESERVATION CONFLICT. This status shall be sent whenever an SCSI device attempts to access a logical unit or an extent within a logical unit that is reserved for that type of access to another SCSI device.

B.2 SENSE DATA

In response to the REQUEST SENSE command sent from an initiator, the target shall send sense data, as described in Figures B-3 and B-4.

Figure B-3: Nonextended Sense Data Format

Bit / Byte	7	6	5	4	3	2	1	0
0	AdValid	Error Class			Error Code			
1	Vendor Unique			Logical Block Address (MSB)				
2	Logical Block Address							
3	Logical Block Address (LSB)							

The address valid (AdValid) bit indicates that the logical block address field contains valid information related to the error code.

The error class specifies a class of errors, with error classes 0 through 6 being vendor unique. For these classes, the error code is vendor unique.

Error class 7 specifies extended sense. Error code zero specifies the extended sense data format. Error code F_H specifies a vendor unique data format for extended sense. Error codes 1H through EH are reserved.

The extended sense data format is shown below.

Figure B-4: Extended Sense Data Format

Bit / Byte	7	6	5	4	3	2	1	0
0	Valid	Error Class			Error Code			
1	Segment Number							
2	Filemark	EOM	ILI	Reserved	Sense Key			
3	Information Byte (MSB)							
4	Information Byte							
5	Information Byte							
6	Information Byte (LSB)							
7	Additional Sense Length (n)							
8 – n+7	Additional Sense Bytes							

The information bytes are not defined if the valid bit is zero. If the valid bit is one, the information bytes contain valid information as follows:

(1) The unsigned logical block address associated with the sense key, for direct-access devices (Type 0), write-once read-multiple devices (Type 4), and read-only direct-access devices (Type 5).

(2) The difference (residue) of the requested length and the actual length in either bytes or blocks, as determined by the command, for sequential-access devices (Type 1), printer devices (Type 2), and processor devices (Type 3). (Negative values are indicated by two's complement notation.)

(3) The difference (residue) of the requested number of blocks and the actual number of blocks copied or compared for the current segment descriptor of a COPY, COMPARE, or COPY AND VERIFY command.

The segment number contains the number of the current segment descriptor if the extended sense is in response to a COPY, COMPARE, or COPY AND VERIFY command. Up to 256 segments are supported beginning with segment zero.

The filemark bit indicates that the current command has read a filemark. This bit is only used for sequential-access devices.

The end-of-medium (EOM) bit indicates that an end-of-medium (end-of-tape, beginning-of-tape, out-of-paper, etc) has occurred on a sequential access device or printer device. For sequential-access devices, this bit indicates that the unit is at or past the early warning end-of-tape if the direction was forward or that the command could not be completed because beginning-of-tape was encountered if the direction was reverse. Direct-access devices shall not use this bit; instead, these devices shall report attempts to access beyond the end-of-medium as ILLEGAL REQUEST sense key.

The incorrect length indicator (ILI) bit indicates that the requested logical block length did not match the logical block length of the data on the medium.

The additional sense length specifies the number of additional sense bytes to follow. If the allocation length of the command descriptor block is too small to transfer all of the additional sense bytes, the additional sense length is not adjusted to reflect the truncation.

The additional sense bytes contain command-specific, peripheral-device-specific data, or both kinds of data that further define the nature of the CHECK CONDITION status. The COPY, COMPARE, COPY AND VERIFY, and SEARCH DATA commands define a standard purpose for some of these bytes. Except as described in these commands, the additional sense bytes are vendor unique.

Figure B-5: Sense Key (0H-7H) Descriptions

Sense Key	Description
0H	NO SENSE. Indicates that there is no specific sense key information to be reported for the designated logical unit. This would be the case for a successful command or a command that received a CHECK CONDITION status because one of the filemark, EOM, or ILI bits is set to one.
1H	RECOVERED ERROR. Indicates that the last command completed successfully with some recovery action performed by the target. Details may be determinable by examining the additional sense bytes and the information bytes.
2H	NOT READY. Indicates that the logical unit addressed cannot be accessed. Operator intervention may be required to correct this condition.
3H	MEDIUM ERROR. Indicates that the command terminated with a nonrecovered error condition that was probably caused by a flaw in the medium or an error in the recorded data.
4H	HARDWARE ERROR. Indicates that the target detected a nonrecoverable hardware failure (for example, controller failure, device failure, parity error, etc.) while performing the command or during a self test.

Figure B-5: Sense Key (8H-FH) Descriptions, cont'd

Sense Key	Description
5H	ILLEGAL REQUEST. Indicates that there was an illegal parameter in the command descriptor block or in the additional parameters supplied as data for some commands (FORMAT UNIT, SEARCH DATA, etc). If the target detects an invalid parameter in the command descriptor block, then it shall terminate the command without altering the medium. The target may have already altered the medium when it detects an invalid parameter in the data.
6H	UNIT ATTENTION. Indicates that the removable medium may have been changed or the target has been reset. A unit attention condition shall begin for each initiator whenever the removable medium may have been changed or the target has been reset. The unit attention condition shall persist for each initiator until that initiator issues a command other than REQUEST SENSE or INQUIRY for which the target shall return CHECK CONDITION status. If the next command from that initiator is REQUEST SENSE, and if the target supports extended sense, then the UNIT ATTENTION sense key shall be returned. (If any other command is received, the unit attention condition is lost.)
7H	DATA PROTECT. Indicates that a command that reads or writes the medium was attempted on a block that is protected from this operation. The read or write operation is not performed.
8H	BLANK CHECK. Indicates that a write-once read-multiple device or a sequential-access device encountered a blank block while reading or a write-once read-multiple device encountered a nonblank block while writing.
9H	Vendor unique. Available for reporting vendor unique conditions.
AH	COPY ABORTED. Indicates a COPY, COMPARE, or COPY AND VERIFY command was aborted due to an error condition on the source device, the destination device, or both. (See 7.1.4.2 for additional information about this sense key.)
BH	ABORTED COMMAND. Indicates that the target aborted the command. The initiator may be able to recover by trying the command again.
CH	EQUAL. Indicates a SEARCH DATA command has satisfied an equal comparison.
DH	VOLUME OVERFLOW. Indicates that a buffered peripheral device has reached the end-of-medium and data remains in the buffer that has not been written to the medium. A RECOVER BUFFERED DATA command(s) may be issued to read the unwritten data from the buffer.
EH	MISCOMPARE. Indicates that the source data did not match the data read from the medium.
FH	This sense key is reserved.

APPENDIX C

GLOSSARY OF TERMS

arbitration winner. The arbitrating SCSI device which has the highest SCSI address.

byte. In this standard, this term indicates an 8-bit (octet) byte.

command descriptor block (CDB). The structure used to communicate requests from an initiator to a target.

connect. The function that occurs when an initiator selects a target to start an operation.

disconnect. The function that occurs when a target releases control of the SCSI bus, allowing it to go to the BUS FREE phase.

initiator. An SCSI device (usually a host system) that requests an operation to be performed by another SCSI device.

INTERMEDIATE status. A status code sent from a target to an initiator upon completion of each command in a set of linked commands except the last command in the set.

logical thread. The logical path which exists between an initiator's memory and a bus device LUN even though the physical path may be disconnected.

logical unit. A physical or virtual device addressable through a target.

logical unit number. An encoded three-bit identifier for the logical unit.

(LSB). Least significant byte.

LUN. Logical unit number.

mm. Millimeter.

ms. Millisecond.

(MSB). Most significant byte.

ns. Nanosecond.

one. A true signal value.

peripheral device. A peripheral that can be attached to an SCSI device (e.g., magnetic-disk, printer, optical-disk, or magnetic-tape).

reconnect. The function that occurs when a target selects an initiator to continue an operation after a disconnect.

---

1 Available from the Electronic Industries Association, 2001 Eye Street NW, Washington, D.C. 20006.

reserved. The term used for bits, bytes, fields, and code values that are set aside for future standardization.

SCSI address. The octal representation of the unique address (0-7) assigned to an SCSI device. This address would normally be assigned and set in the SCSI device during system installation.

SCSI **ID.** The bit-significant representation of the SCSI address referring to one of the signal lines DB(7-0).

SCSI **device.** A host computer adapter or a peripheral controller or an intelligent peripheral that can be attached to the SCSI bus.

signal assertion. The act of driving a signal to the true state.

signal deassertion. The act of driving a signal to the false state or allowing the cable terminators to bias the signal to the false state (by placing the driver in the high impedance condition).

signal release. The act of allowing the cable terminators to bias the signal to the false state (by placing the driver in the high impedance condition).

status. One byte of information sent from a target to an initiator upon completion of each command.

target. An SCSI device that performs an operation requested by an initiator.

us. Microsecond.

vendor unique. In this standard, this term indicates bits, fields, or code values that are vendor specific.

xxH. Numbers followed by capital H are hexadecimal values. All other numbers are decimal values.

zero. A false signal value.

NOTES:

NOTES:

NOTES:

ADSI ADAPTIVE DATA SYSTEMS INC.

who brought you the First Family of SCSI products . . .

SABER - Disk Controller

PYTHON - Tape Controller

IBM PC Host Adapter

is bringing you the Next Generation
based on our proprietary VLSI chip technology . . .

SCSI DISK CONTROLLERS

- ST506/412
- ESDI
- 3-1/2" or 5-1/4" Form Factor
- Variable Sector, Size to 4096 bytes
- 64KB Data Buffer
- 4 Drive Capability
- ADSI ASIC Chip Sets
- High Reliability
- Bus Parity Checking
- 48 Bit ECC
- SCSI Rev 16 Compatibility
- Low Power Consumption
- 1-to-1 Minimum Interleave
- Sector Skewing
- Copy Command

SCSI TAPE CONTROLLERS

- Industry Standard QIC-36 Interface
- QIC-11 or QIC-24 Selectable Media Formats
- Full SCSI Disconnect/Arbitration/ Reselect Capability
- SCSI Bus and Buffer Parity (Optional)
- 16K On-Board Data Buffer
- Automatic Cartridge Type Detection
- Read-After Write Data Check
- 16 Bit CRC Error Detection Coding
- 5-1/4" Form Factor
- ADSI ASIC Chip Sets
- Copy Command

SCSI COMBO TAPE/DISK

- ST506 and QIC-36 or ESDI and QIC-36
- 64K Data Buffer
- Variable Sector Size to 4096 Bytes
- 1-to-1 Minimum Interleave
- 5-1/4" Form Factor
- 48 Bit ECC
- Copy Command
- 2.0 MB/Sec Host Transfer Rate
- 2 Disk 1 Tape Support
- ADSI ASIC Chip Sets

also . . . VLSI CUSTOM CHIP SETS

available for your products

DISK CONTROLLER SERDES ADS 1000

- ST506, ST412, ST412 HP, ESDI, SMD, ESMD
- Variable Sector Sizes to 4096 bytes
- Generic Microprocessor Interface
- 24 MHZ NRZ Data, 24 MHZ Clock
- 48 Bit ECC
- MOS Technology

TAPE SERDES ADS-4360

- QIC-11 or QIC-24, Tape Format
- 8 Bit Microprocessor Interface
- 16 Bit CRC
- Read After Write Verify
- GCR Data
- Bit by Bit Read Compare
- 6 MHZ Clock
- MOS Technology

BUFFER MANAGER ADS-3570

- 4 Ports
- Priority Resolution
- 256K Addressing
- DRAM Control
- SCSI Bus Support
- 8 Bit Parallel Data
- 15 MHZ Clock
- MOS Technology

SCSI BUS TRANSCEIVER ADS-5050

- Direct Interface to SCSI Bus
- Single Ended Bus
- Data/Control Mode Select
- 9 Bit Wide Data Path
- Power Up/Down Protection
- Bi-Polar Technology

ADSI ADAPTIVE DATA SYSTEMS INC.

ADAPTIVE DATA SYSTEMS, INC.
2627 POMONA BOULEVARD
POMONA, CALIFORNIA 91768
(714) 594-5858
TWX: 910 581 1211