REFERENCE
NBS
PUBLICATIONS

ANSI X3.131-1986

American National Standard

Adopted for Use by the Federal Government

FIPS PUB 131

See Notice on Inside Front Cover

for information systems –

small computer system interface (SCSI)

ANSI X3.131-1986

american national standards institute, inc.
1430 broadway, new york, new york 10018

JK
468
.A8A3
No.131
1987

INTEL-1033
8,010,740

This standard has been adopted for Federal Government use.

Details concerning its use within the Federal Government are contained in Federal Information Processing Standards Publication 131, Small Computer System Interface (SCSI). For a complete list of the publications available in the Federal Information Processing Standards Series, write to the Standards Processing Coordinator (ADP), Institute for Computer Sciences and Technology, National Bureau of Standards, Gaithersburg, MD 20899.

ANSI®
X3.131-1986

American National Standard for Information Systems –

Small Computer System Interface (SCSI)

Secretariat

Computer and Business Equipment Manufacturers Association

Approved June 23, 1986

American National Standards Institute, Inc

Abstract

This standard defines mechanical, electrical, and functional requirements for attaching small computers with each other and with low- to medium-performance intelligent peripherals such as rigid disks, flexible disks, magnetic tape devices, printers, and optical disks. The resulting interface facilitates the interconnection of small computers and intelligent peripherals and thus provides a common interface specification for both systems integrators and suppliers of intelligent peripherals.

American National Standard

Approval of an American National Standard requires verification by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer.

Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that a concerted effort be made toward their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and will in no circumstances give an interpretation of any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretations should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

CAUTION NOTICE: This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken to reaffirm, revise, or withdraw this standard no later than five years from the date of approval. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

Published by

American National Standards Institute
1430 Broadway, New York, New York 10018

Copyright © 1986 by American National Standards Institute, Inc
All rights reserved.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Printed in the United States of America

PC1100588/25

FOREWORD (This Foreword is not part of American National Standard X3.131-1986.)

This standard specifies the mechanical, electrical, and functional requirements for a small computer input/output bus interface, and command sets for peripheral device types, particularly storage devices, commonly used with small computers.

The major goal of this standard is to facilitate the integration of physically small computers and intelligent peripheral devices, particularly storage devices, into computer systems. The interface is suitable for implementation in LSI circuitry, and uses inexpensive drivers, receivers, cables, and connectors, but operates over moderate distances (up to 25 meters) and moderately high data rates (up to 4 megabytes per second). The command sets that are a part of this standard are "device independent"; that is, they largely hide the internal structure of the device (e.g., cylinders, heads, sectors, and the like) from the interface, facilitating the development of "generic" device driver software that can control all devices of a particular type.

The development of comparatively inexpensive VLSI device controllers have recently changed the economics of interfaces for small system storage devices. Where expensive controller logic was once shared among as many devices as possible, in many cases it now makes economic sense to build a controller in each device. This is particularly true for high-performance storage devices, where the intimate interactions of the recording medium, the recording mechanism, and the recording code cause intersymbol interference and error recovery problems that are highly specific to the chosen technologies, and are best resolved within the device itself.

Moreover, the number of types of storage devices for small computers, and the industry that builds them, have grown dramatically in the past few years. In particular, the emergence of physically small, but comparatively high-capacity and high-performance fixed-medium magnetic disk devices (virtually non-existent in the late 1970s, but a multibillion dollar business in the mid-1980s) has driven the development of small computer systems and caused a need for other classes of devices, such as streaming cartridge tape drives, for backup and data interchange.

Because device interfaces are very specific to device types, many device level interface standards would be required to service all small computer device types. Because backplane buses reside at the center of computers, and have dramatic performance effects, many different ones are needed for different system requirements. To connect every backplane bus to every device interface through a controller would require an almost unbounded number of specific controller products.

In addition, in many systems today, it is not the computer which is "central", it is the storage facility. That is, one or two large capacity storage subsystems serve several computers. An interface adapted to this reality was needed.

By 1982, all the needs given above were widely recognized in the industry and by the members of X3T9 and its Task Groups. A commercial small system parallel bus, the Shugart Associates System Interface (SASI), generally met

the small system requirements for a device-independent peripheral or system bus and had enjoyed significant market success. It was offered to X3T9.2 as the basis for a standard. X3T9.2 chose the name Small Computer System Interface (SCSI) for that standard and began work at its April 1982 meeting. The present SCSI ANS is a formalization and extension of the SASI. Many existing SASI devices are SCSI compatible.

Since April 1982, X3T9.2 has held plenary sessions, at two month intervals, plus numerous informal working meetings. The original SASI has been extended in a number of ways, including:

(1) A differential electrical option has been added to allow use of longer cables (up to 25 meters) in environments where common mode noise is a concern.

(2) A synchronous transfer option has been developed allowing maximum transfer rates in the 3 to 4 megabyte per second range.

(3) An optional "extended" command set has been added, allowing for very large capacity storage devices (a block address space of 2^{32} blocks versus 2^{21} for SASI), and Inquiry commands that allow self-configuring driver software.

(4) Command sets for magnetic tape (both start/stop and streaming), printers, processors, optical disks, and read-only optical disks have been added to the proposed SCSI standard, in addition to those for magnetic disks.

Although it might have been premature in April 1982 to claim that SASI was then a de facto standard, this surely is the case for SASI/SCSI today. SCSI compatible host adapters, controllers, and peripheral devices are now widely manufactured around the world. Host adapters are available for most small computers with accessible backplane buses. SCSI controllers are widely available for all the de jure and de facto standard magnetic disk and magnetic tape device interfaces. Small high-capacity fixed-medium magnetic disks, rigid removable-medium magnetic disks, high-capacity Bernoulli-effect flexible disks, and other closed-loop high-capacity flexible disk products, as well as optical disk products, are all available with integral SCSI controllers. SCSI subsystems that integrate both a rigid-disk and a streaming-tape drive into a single package are also available. SCSI interface chips are available, and some disk controller chip sets also provide SCSI support.

Since a large number of companies have implemented and tested the SCSI during the development of the proposed standard, no separate test program has been deemed necessary. At the February 1984 meeting of X3T9.2, representatives of the following companies stated, for the record, that their companies had implemented and tested SCSI: Adaptec, Adaptive Data, Data Technology Corp., Fujitsu America, Inc., NCR Corp., and Shugart Corp. This is not a complete list of companies implementing products using SCSI nor is it a promise by these companies to offer SCSI products.

The SCSI fills an urgent need, provides for the future, and is consonant with actual commercial practice. Most important, the timing is right. SCSI catches the floodtide of new, high-performance storage devices for small systems, and promises to bring a measure of needed order to what would otherwise be a chaotic and fragmented market.

Suggestions for improvement of this standard will be welcome. They should be sent to the Computer and Business Equipment Manufacturers Association, 311 First Street, NW, Suite 500, Washington, DC 20001.

This standard was processed and approved for submittal to ANSI by Accredited Standards Committee on Information Processing Systems, X3. Committee approval of the standard does not imply that all committee members voted for its approval. At the time it approved this standard, the X3 Committee had the following members:

Edward Lohse, Chair
Richard Gibson, Vice-Chair
Catherine A. Kachurik, Administrative Secretary

Organization Represented	Name of Representative
American Express	D. L. Seigal Lucille Durfee (Alt)
American Library Association	Paul Peters
American Nuclear Society	Geraldine C. Main D. R. Vondy (Alt)
AMP Incorporated	Patrick E. Lannan Edward Kelly (Alt)
Association for Computer Machinery	Kenneth Magel Jon A. Meads (Alt)
Association of the Institute for Certification of Computer Professionals	Thomas M. Kurihara
AT&T Communications	Henry L. Marchese Richard Gibson (Alt)
AT&T Technologies	Herbert V. Bertine Paul D. Bartoli (Alt) Stuart M. Garland (Alt)
Burroughs Corporation	Stanley Fenner
Control Data Corporation	Charles E. Cooper Keith Lucke (Alt)
Cooperating Users of Burroughs Equipment	Thomas Easterday Donald Miller (Alt)
Data General Corporation	John Pilat Lyman Chapin (Alt)
Data Processing Management Association	Christian G. Meyer Ward Arrington (Alt) Terrance H. Felker (Alt)
Digital Equipment Computer Users Society	William Hancock Dennis Perry (Alt)
Digital Equipment Corporation	Gary S. Robinson Delbert L. Shoemaker (Alt)
Eastman Kodak	Gary Haines Charleton C. Bard (Alt)
General Electric Company	Richard W. Signor William R. Kruesi (Alt)

Organization Represented	Name of Representative
General Services Administration	William C. Rinehuls Larry L. Jackson (Alt)
GUIDE International	Frank Kirshenbaum Sandra Swartz Abraham (Alt)
Harris Corporation	Walter G. Fredrickson Rajiv Sinha (Alt)
Hewlett-Packard	Donald C. Loughry
Honeywell Information Systems	Thomas J. McNamara David M. Taylor (Alt)
IBM Corporation	Mary Anne Gray Robert H. Follett (Alt)
IEEE Computer Society	Sava I. Sherr Thomas M. Kurihara (Alt) Thomas A. Varetoni (Alt)
Lawrence Berkeley Laboratory	David F. Stevens Robert L. Fink (Alt)
Moore Business Forms	Delmer H. Oddy
National Bureau of Standards	Robert E. Rountree James H. Burrows (Alt)
National Communications System	George W. White
NCR Corporation	Thomas W. Kern A. Raymond Daniels (Alt)
Prime Computer, Inc.	Joseph Schmidt John McHugh (Alt)
Railinc Corporation	R. A. Petrash
Recognition Technology Users Association	Herbert F. Schantz G. W. Wetzel (Alt)
Scientific Computer Systems Corporation	James A. Baker Carl Haberland (Alt)
SHARE, Inc	Thomas B. Steel Robert A. Rannie (Alt)
Sperry Corporation	Marvin W. Bass Jean G. Smith (Alt)
Texas Instruments, Inc	Presley Smith Richard F. Trow, Jr (Alt)
3M Company	Paul D. Jahnke J. Wade Van Valkenburg (Alt)
Travelers Insurance Companies, Inc	Joseph T. Brophy
U.S. Department of Defense	Fred Virtue Belkis Leong-Hong (Alt)
VIM	Chris Tanner Madeline Sparks (Alt)
VISA U.S.A.	Jean T. McKenna Susan Crawford (Alt)
Wang Laboratories, Inc	Marsha Hayek Joseph St. Amand (Alt)
Xerox Corporation	John L. Wheeler Roy Pierce (Alt)

Subcommittee X3T9 on I/O interfaces, which reviewed this standard, had the following members:

Delbert L. Shoemaker, Chair
William E. Burr, Vice-Chair
Ron Tranquilli, Vice-Chair

Dennis Appleyard
James Barnette
Duane Barney
Bob Bender
John Blagaila
Fred Ciechowski
Steve Cooper
Louis C. Domshy
Robert Dugan
Ross Jaibaji
Reinhard Knerr
Pat Lannan
John B. Lohmeyer
John McCool
Gene Milligan
Tom Morrow
Mike Newton
Ted Petrowich
Gary S. Robinson
Arnold J. Roccati
Floyd Ross
Holly S. White

Robert B. Anthony (Alt)
G. Atterbury (Alt)
Robert L. Bergey (Alt)
Charles Brill (Alt)
Roger Cormier (Alt)
Hank Dorris (Alt)
Thomas A. Fiers (Alt)
Henry Ginter (Alt)
Mark Hammang (Alt)
Sunil Joshi (Alt)
Kris Kowal (Alt)
Dennis Krob (Alt)
William J. McClain (Alt)
William Mosenthal, Jr (Alt)
Kirk Moulton (Alt)

Task Group X3T9.2 on Lower-Level Interfaces, which developed this standard, had the following members:

William E. Burr, Chair
John B. Lohmeyer, Vice-Chair

Ezera Alculdia
I. Dal Allan
Dennis Appleyard
J. L. Armstutz
Robert Barter
Robert Bender
Bruce Bonner
Larry Boucher
Tom Brigs
Charles Brill
Claude R. Cardot
William J. Casey
Joseph Chen
David Cornaby
David F. Craft, Jr
Al Crowell
Jay Cuningham
Willard S. Davidson
Terry Dawson
Marvin R. DeForest
Phil Devin
Mario DeFede
Louis C. Domshy
Norm Dornseif
Alan Ebright
Jesse Eckelkamp
P. Allen England
David Filpus
Stephen Fitzgerald
Greg Floryance
Mark Forbes
Anita Freeman
John Frissell
Abe Gindi
Steve Goldman
Bill Grace
Greg Hite
William A. Horton
John Hoy
Sam Karunanithi
John A. Klonick
Robert Knight
Anthony Kozlowski
Curtiss Krueger
Franc Krulc
Lawrence J. Lamers
Pat Lannan
Rick Lehrbaum
Robert Liu
Daniel Loski
Mike Machado
Bob Masterson
William C. Mavity
Hank Meyer
Gene Milligan
Bob Mortensen
Jim Murphy
Paul Nitza
Ted Petrowich
Joseph Piacentino
Ken Post
Paul Rikkonen
Gary S. Robinson
D. Michael Robinson
Arnold J. Roccati
Don Rodgers
Tony Salthouse
Jack Schiffhauer
James Schuessler
Ralph Schultz
Moshe Segal
Robert N. Snively
Jeff Stai
Adrienne Turenne
Harvey Waltersdorf
William Watson
Norm Zimmerman
Karen Ammeberg (Alt)
Robert B. Anthony (Alt)
Ralph H. Bolen (Alt)
Bill Bunker (Alt)
E. W. Calkins (Alt)
Ed Cieniawa (Alt)
George Clark (Alt)
Paul Clement (Alt)
Steve Cooper (Alt)
Stephen Cornaby (Alt)
Steve Crompton (Alt)
Gary Crowell (Alt)
Charles DiChiara (Alt)
William Dollar (Alt)
Marty Francis (Alt)
Edward A. Gardner (Alt)
Jon Ericson (Alt)
Tom Fiers (Alt)
Paul K. Garretson (Alt)
Frank Haas (Alt)
William Homans (Alt)
Gerald Houlder (Alt)
J. V. Howell (Alt)
Robert S Jerritts (Alt)
Skip Jones (Alt)
Mort Kalet (Alt)
Larry Kendall (Alt)
Skip Kilsdonk (Alt)
Jim Korpi (Alt)
Dennis Krob (Alt)
Marshal Lee (Alt)
Tony Lo (Alt)
Lisa Long (Alt)
Rino Mazzucco (Alt)
Dave McIntyre (Alt)
Jon Milledge (Alt)
Allan Miller (Alt)
Daniel E. Moczarny (Alt)
Joanne Monaco (Alt)
Keith Mueller (Alt)
Don Nanneman (Alt)
Mike Newton (Alt)
Doug Nolff (Alt)
Michael O'Donnell (Alt)
Giles A. Pauli (Alt)
Paul Phillips (Alt)
Johhny Pierce (Alt)
Guy Prival (Alt)
Bob Reid (Alt)
Richard Reiser (Alt)
Curt Ridgeway (Alt)
Floyd Ross (Alt)
Jay Seashore (Alt)
James J. Semenak (Alt)
Del Shoemaker (Alt)
Terry Smudz (Alt)
David Snook (Alt)
Chuck Spatafore (Alt)
D. W. Spence (Alt)
Paul Stavish (Alt)
Ron Sulyma (Alt)
Tom Treadway (Alt)
Hobie Weaver (Alt)

TABLE OF CONTENTS

1. Scope....12

2. Referenced Standard....14

3. Glossary and Conventions....14
   3.1 Glossary....14
   3.2 Editorial Conventions....15

4. Physical Characteristics....15
   4.1 Physical Description....16
   4.2 Cable Requirements....16
   4.3 Connector Requirements....16
   4.4 Electrical Description....22
   4.5 SCSI Bus....25
   4.6 SCSI Bus Signals....27
   4.7 SCSI Bus Timing....30

5. Logical Characteristics....31
   5.1 SCSI Bus Phases....31
   5.2 SCSI Bus Conditions....39
   5.3 SCSI Bus Phase Sequences....40
   5.4 SCSI Pointers....43
   5.5 Message System Specification....43

6. SCSI Commands....51
   6.1 Command Implementation Requirements....52
   6.2 Command Descriptor Block....53
   6.3 Command Examples....58

7. Command Descriptions for All Device Types....61
   7.1 Group 0 Commands for All Device Types....61
   7.2 Group 1 Commands for All Device Types....82
   7.3 Group 2 Commands for All Device Types....85
   7.4 Group 3 Commands for All Device Types....85
   7.5 Group 4 Commands for All Device Types....85
   7.6 Group 5 Commands for All Device Types....85
   7.7 Group 6 Commands for All Device Types....85
   7.8 Group 7 Commands for All Device Types....85

8. Command Descriptions for Direct-Access Devices....86
   8.1 Group 0 Commands for Direct-Access Devices....86
   8.2 Group 1 Commands for Direct-Access Devices....113

9. Group 0 Command Descriptions for Sequential-Access Devices....127

10. Group 0 Command Descriptions for Printer Devices....151

11. Group 0 Command Descriptions for Processor Devices....164

12. Command Descriptions for Write-Once Read-Multiple Devices....167
   12.1 Group 0 Commands for Write-Once Read-Multiple Devices....167
   12.2 Group 1 Commands for Write-Once Read-Multiple Devices....176

13. Command Descriptions for Read-Only Direct-Access Devices.....183
13.1 Group 0 Commands for Read-Only Direct-Access Devices.....183
13.2 Group 1 Commands for Read-Only Direct-Access Devices.....184

14. Status 185

LIST OF FIGURES

4-1 Nonshielded SCSI Device Connector.....18
4-2 Nonshielded Cable Connector.....19
4-3 Termination for Single-Ended Devices.....24
4-4 Termination for Differential Devices.....24
4-5 Differential Driver Protection Circuit (Optional).....24
4-6 SCSI ID Bits.....25
4-7 Sample SCSI Configurations.....26

5-1 Phase Sequences without Arbitration.....42
5-2 Phase Sequences with Arbitration.....42
5-3 Simplified SCSI System.....43

Appendixes

Appendix A.....187

SCSI Signal Sequence Example

Appendix A Figures
A1: SCSI Signal Sequence Example.....188

Appendix B.....192

Typical Bus Phase Sequence

Appendix C.....194

SCSI System Operation

C1. Host Memory / Host Adapter / SCSI Controller Relationship.....194
C2. SCSI READ Command Example.....195
C3. I/O Channel Concept.....196

Appendix C Figures
C1: Snapshot Prior to Initial Selection.....198
C2: Snapshot Prior to Data Transfer.....199

Appendix D.....200

Recommended Shielded Connectors

D1. Shielded Connector, Alternative 1.....200
D2. Shielded Connector, Alternative 2.....200
D3. EUROCARD Boxes.....200

Appendix D Figures
D1: Female Shielded SCSI Cable Connector, Alternative 1.....201
D2: Male Shielded SCSI Device Connector, Alternative 1.....202
D3: Shielded SCSI Device Connector, Alternative 2.....203
D4: Shielded SCSI Cable Connector, Alternative 2.....204

Appendix E.....207

Conformance

E1. Alternatives.....207
E2. Levels of Conformance.....207
E3. Options.....208
E4. Statement of Conformance.....208

Appendix F.....210

Additional Medium Type and Density Code Standards

Appendix G.....212

Future Standardization

American National Standard for Information Systems -

Small Computer System Interface (SCSI)

1. Scope

This American National Standard provides the mechanical, electrical, and functional requirements for a small computer input/output bus and command sets for peripheral device types commonly used with small computers.

The small computer system interface, described in this standard, is a local I/O bus that can be operated at data rates up to 4 megabytes per second depending upon circuit implementation choices. The primary objective of the interface is to provide host computers with device independence within a class of devices. Thus, different disk drives, tape drives, printers, and even communication devices can be added to the host computer(s) without requiring modifications to generic system hardware or software. Provision is made for the addition of nongeneric features and functions through vendor unique fields and codes.

The interface uses logical rather than physical addressing for all data blocks. For direct access devices, each logical unit may be interrogated to determine how many blocks it contains. A logical unit may coincide with all or part of a peripheral device.

Provision is made for cable lengths up to 25 meters using differential drivers and receivers. A single-ended driver and receiver configuration is defined for cable lengths of up to 6 meters and is primarily intended for applications within a cabinet.

The interface protocol includes provision for the connection of multiple initiators (SCSI devices capable of initiating an operation) and multiple targets (SCSI devices capable of responding to a request to perform an operation). Optional distributed arbitration (i.e., bus-contention logic) is built into the architecture of SCSI. A priority system awards interface control to the highest priority SCSI device that is contending for use of the bus. The time to complete arbitration is independent of the number of devices that are contending and can be completed in less than 10 microseconds.

The physical characteristics are described in Section 4. There are two electrical alternatives: single-ended and differential. Single-ended and differential devices are electrically different and shall not be mixed on the same bus. In addition, there are several options: shielded or unshielded connectors may be used and parity may or may not be implemented.

Section 5 describes the logical characteristics of the interface. An

arbitration option is defined to permit multiple initiators and to permit concurrent I/O operations. All SCSI devices are required to be capable of operating with the defined asynchronous transfer protocol. In addition, an optional synchronous transfer protocol is defined. Section 5 also specifies a message protocol for control of the interface. In most cases, messages are not directly apparent to the host computer software. Only one message, COMMAND COMPLETE, is mandatory; all others are optional and are not necessarily implemented. Note that some options (e.g., synchronous transfer) require the implementation of certain messages.

The SCSI command structure is specified in Section 6. Commands are classified as mandatory (M), extended (E), optional (O), or vendor unique (V). SCSI devices shall implement all mandatory commands defined for the appropriate device type and may implement other commands as well. Extended SCSI devices shall implement all extended plus all mandatory commands and may implement other commands as well. Extended SCSI devices contain commands that facilitate the writing of self-configuring software drivers that can "discover" all necessary attributes without prior knowledge of specific peripheral characteristics (such as storage capacity). Extended commands for direct access devices also implement a very large logical block address space (2^{32} blocks), although mandatory commands for direct access devices implement a somewhat smaller logical block address space (2^{21} blocks).

Section 7 specifies those commands that have a consistent meaning for all device types.

Sections 8 through 13 contain commands for direct-access (e.g., magnetic disk), sequential-access (e.g., magnetic tape), printer, processor, write-once-read-multiple (e.g., optical disk), and read-only direct-access devices, respectively. The commands in each of these sections are unique to the device type, or they have interpretations, fields, or features that are specific for the device type. Thus, for example, although the WRITE command is used for several device types, it has a somewhat different form for each type, with different parameters and meanings. Therefore, it is specified separately for each device type.

Section 14 describes the status byte for all device types. Status is returned by targets at the end of each command.

Appendixes A through C provide examples of SCSI signal sequences, timing, and phase sequences. Appendix D contains information on recommended shielded connectors. Appendix E contains information on conformance statements. Appendix F contains information on other standards related to medium types and density codes for flexible disks and magnetic tapes. Appendix G contains information on future extensions to SCSI that are being considered by X3T9.2 However, the Appendixes are not part of this standard.

2. Referenced Standard

This standard is intended for use in conjunction with the Standard for Electrical Characteristics of Generators and Receivers for use in Balanced Digital Multipoint Systems, EIA RS-485-1983.[1]

3. Glossary and Conventions

3.1 Glossary

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 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).

---

[1] Available from the Electronic Industries Association, 2001 Eye Street NW, Washington, D.C. 20006.

reconnect. The function that occurs when a target selects an initiator to continue an operation after a disconnect.

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 negation. 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 and are not defined by this standard.

xxh. Numbers followed by lowercase h are hexadecimal values. All other numbers are decimal values.

zero. A false signal value.

3.2 Editorial Conventions. Certain words, terms, and phrases used in this standard have a specific meaning beyond the normal English meaning. These words, terms, and phrases are defined either in the glossary (see 3.1) or in the text where they first appear (e.g., Arbitration Delay, see 4.7.1). Names of signals, phases, conditions, messages, commands, statuses, and sense keys are in all uppercase (e.g., REQUEST SENSE). Lowercase is used for words having the normal English meaning.

4. Physical Characteristics

This section contains the physical definition of the SCSI. The connectors, cables, signals, terminators, and bus timing needed to implement SCSI are covered.

4.1 Physical Description. SCSI devices are daisy-chained together using a common cable. Both ends of the cable are terminated. All signals are common between all SCSI devices. Two driver/receiver alternatives are available:

(1) Single-ended drivers and receivers, which allow a maximum cable length of six meters (primarily for connection within a cabinet)

(2) Differential drivers and receivers, which allow a maximum cable length of 25 meters (primarily for connection outside of a cabinet)

4.2 Cable Requirements. An ideal impedance match with cable terminators implies a cable characteristic impedance of 132 ohms (singled-ended option) or 122 ohms (differential option). In general, cables with this high of a characteristic impedance are not available; however, impedances that are somewhat lower are satisfactory. A characteristic impedance of 100 ohms ± 10% is recommended for unshielded flat or twisted pair ribbon cable. A characteristic impedance greater than 90 ohms is preferred for shielded cables; however, most available cables have a somewhat lower characteristic impedance. To minimize discontinuities and signal reflections, cables of different impedances should not be used in the same bus. Implementations may require trade-offs in shielding effectiveness, cable length, the number of loads, transfer rates, and cost to achieve satisfactory system operation.

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

4.2.1 Single-Ended Cable. A 50-conductor flat cable or 25-signal twisted-pair cable shall be used. The maximum cable length shall be 6.0 meters.

A stub length of no more than 0.1 meters is allowed off the mainline interconnection within any connected equipment.

SCSI bus termination may be internal to the SCSI devices that are at the ends of the cable.

4.2.2 Differential Cable. A 50-conductor cable or 25-signal twisted-pair cable shall be used. The maximum cable length shall be 25 meters.

A stub length of no more than 0.2 meters is allowed off the mainline interconnection within any connected equipment.

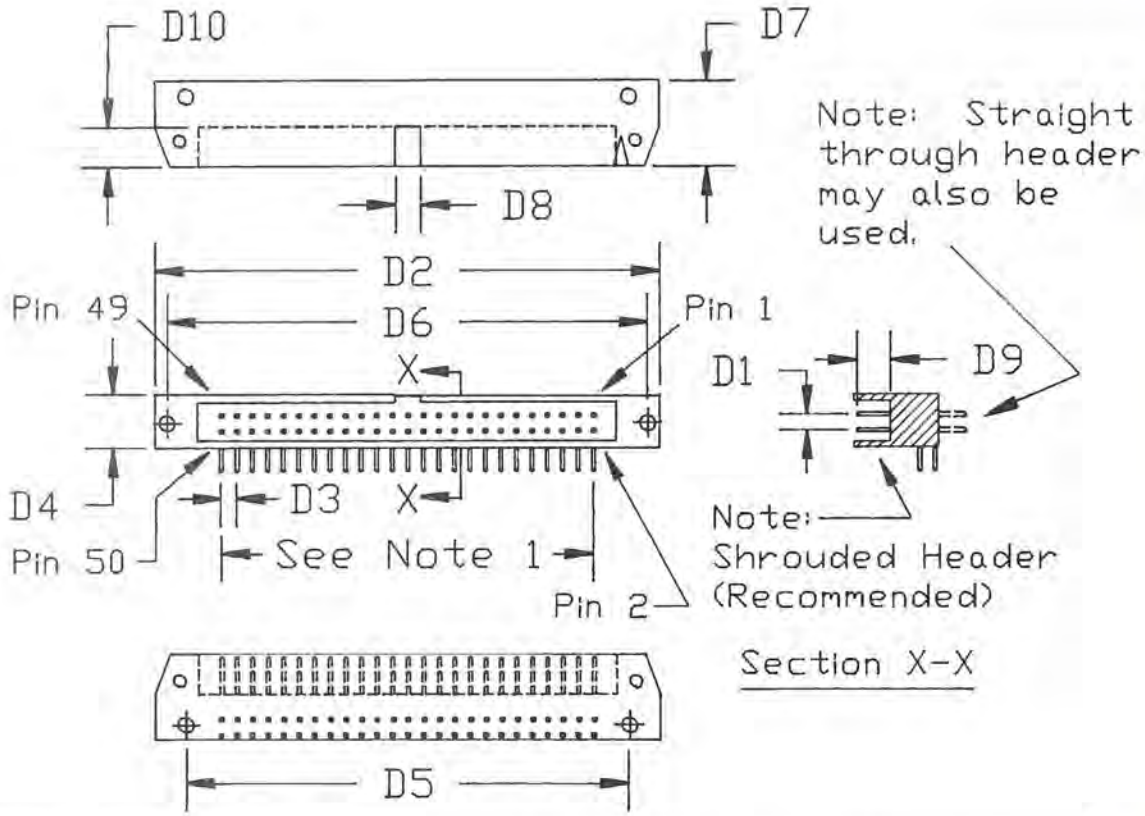
SCSI bus termination may be internal to the SCSI devices that are at the ends of the cable.

4.3 Connector Requirements. Nonshielded connectors are specified. The nonshielded connectors are typically used for in-cabinet applications. Appendix D defines recommended shielded connectors and their pin assignments. These connectors are typically used for external applications where electromagnetic compatibility (EMC) and electrostatic discharge (ESD) protection may be required. Either type of connector may be used with the single-ended or differential drivers.

The nonshielded SCSI device connector (Figure 4-1) shall be a 50-conductor connector consisting of two rows of 25 male pins with adjacent pins 2.54 mm (0.1 in) apart. A shroud and header body should be used. The nonmating portion of the connector is shown for reference only.

The nonshielded cable connector (Figure 4-2) shall be a 50-conductor connector consisting of two rows of 25 female contacts with adjacent contacts 2.54 mm (0.1 in) apart. It is recommended that keyed connectors be used.

The unshielded connector pin assignments shall be as shown in Table 4-1 for single-ended drivers and as shown in Table 4-2 for differential drivers.



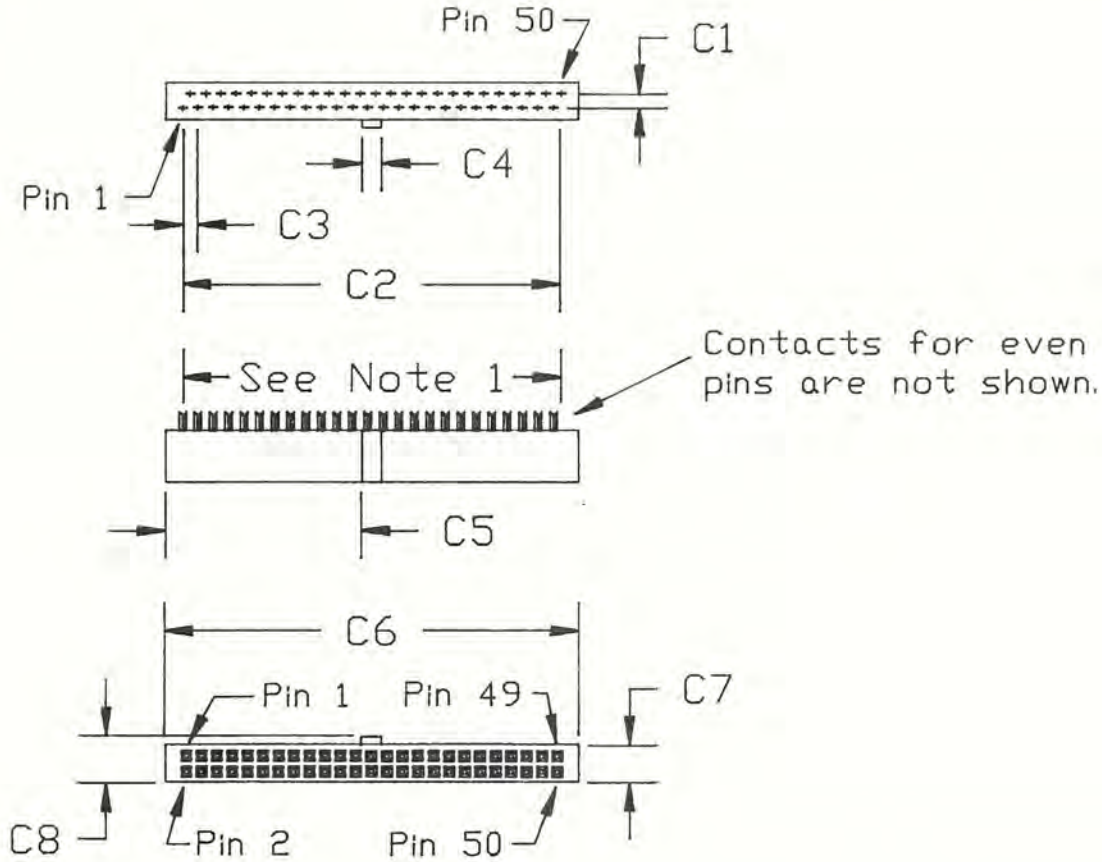
Dimension	Millimeters	Inches	Comments
D1	2.54	0.100	
D2	82.80	3.260	Reference Only
D3	2.54	0.100	
D4	8.89	0.350	Reference Only
D5	72.64	2.860	Reference Only
D6	78.74	3.100	Reference Only
D7	13.94	0.549	Reference Only
D8	4.19±0.25	0.165±0.01	
D9	6.09	0.240	
D10	6.60	0.260	Reference Only

NOTES:

(1) Two rows of twenty five contacts on 2.54-mm (0.100-inch) spacing = 60.96 mm (2.400 inch).

(2) Tolerances ±0.127 mm (0.005 inch) noncumulative, unless specified otherwise.

Figure 4-1. Nonshielded SCSI Device Connector



Dimensions	Millimeters	Inches	Comments
C1	2.540	0.100	
C2	60.960	2.400	
C3	2.540	0.100	
C4	3.302	0.130	
C5	32.385	1.275	
C6	68.072	2.680	
C7	6.096	0.240	
C8	7.620	0.300	Maximum

NOTES:

(1) Fifty contacts on 1.27-mm (0.05-inch) staggered spacing = 62.23 mm (2.450 inch) [reference only].

(2) Tolerances ±0.127 mm (0.005) noncumulative, unless specified otherwise.

(3) Connector cover and strain relief are optional.

Figure 4-2. Nonshielded Cable Connector

Table 4-1
Single-Ended Pin Assignments

Signal	Pin Number
-DB(0)	2
-DB(1)	4
-DB(2)	6
-DB(3)	8
-DB(4)	10
-DB(5)	12
-DB(6)	14
-DB(7)	16
-DB(P)	18
GROUND	20
GROUND	22
GROUND	24
TERMPWR	26
GROUND	28
GROUND	30
-ATN	32
GROUND	34
-BSY	36
-ACK	38
-RST	40
-MSG	42
-SEL	44
-C/D	46
-REQ	48
-I/O	50

NOTES:

(1) All odd pins except pin 25 shall be connected to ground. Pin 25 should be left open. Some products designed prior to the generation of this standard connected this pin to ground.

(2) The minus sign next to the signals indicates active low.

Table 4-2
Differential Pin Assignments

Signal Name	Pin Number		Signal Name
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

NOTE: SHIELD GROUND is optional on some cables. (Implementors note: Some shielded flat ribbon cables use pin 1 as a connection to the shield.)

4.4 Electrical Description

NOTE: For these measurements, SCSI bus termination is assumed to be external to the SCSI device. An SCSI device may have the provision for allowing optional internal termination.

4.4.1 Single-Ended Alternative. All assigned signals shall be terminated with 220 ohms to +5 volts (nominal) and 330 ohms to ground at each end of the cable. (See Figure 4-3.) All signals shall use open-collector or three-state drivers.

4.4.1.1 Output Characteristics. Each signal driven by an SCSI device shall have the following output characteristics when measured at the SCSI device's connector:

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

4.4.1.2 Input Characteristics. Each signal received by an SCSI device shall have the following input characteristics when measured at the SCSI device's connector:

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

4.4.2 Differential Alternative. All signals consist of two lines denoted +SIGNAL and -SIGNAL. A signal is true when +SIGNAL is more positive than -SIGNAL, and a signal is false when -SIGNAL is more positive than +SIGNAL. All assigned signals shall be terminated at each end of the cable as shown in Figure 4-4.

NOTE: As an option, the DIFFSENS signal of the connector is reserved for an active high enable for the differential drivers. If a single-ended device or terminator is inadvertently connected, this signal is grounded, disabling the drivers. (See Figure 4-5.)

4.4.2.1 Output Characteristics. Each signal driven by an SCSI device shall have the following output characteristics when measured at the SCSI device's connector:

V_{OL} (Low-level output voltage) = 2.0 V maximum at I_{OL} (Low-level output current) = 55 milliamps.
V_{OH} (High-level output voltage) = 3.0 V minimum at I_{OH} (High-level output current) = -55 milliamps.
V_{OD} (Differential voltage) = 1.0 V minimum with common-mode voltage ranges from -7 volts dc to +12 volts dc.

V_{OL} and V_{OH} shall be as measured between the output terminal and the SCSI device's logic ground reference.

The output characteristics shall additionally conform to EIA RS-485-1983.

4.4.2.2 Input Characteristics. Each signal received by an SCSI device shall have the following input characteristics when measured at the SCSI device's connector:

I_I (Input current on either input) = $\pm$ 2.0 milliamps maximum.

NOTE: These characteristics include both receivers and passive drivers.

This requirement shall be met with the input voltage varying between -7 volts dc and +12 volts dc, with power on or off, and with the hysteresis equaling 35 millivolts, minimum.

The input characteristics shall additionally conform to EIA RS-485-1983.

4.4.3 Terminator Power (Optional). Single-ended SCSI devices providing terminator power (TERMPWR) shall have the following characteristics:

V_{Term} = 4.0 volts dc to 5.25 volts dc
800 milliamps minimum source drive capability
1.0 milliamp maximum sink capability (except for the purposes of providing power to an internal terminator) with 1.0 amp recommended current limiting (e.g., a fuse).

Differential SCSI devices providing terminator power (TERMPWR) shall have the following characteristics:

V_{Term} = 4.0 volts dc to 5.25 volts dc
600 milliamps minimum source drive capability
1.0 milliamp maximum sink capability (except for the purposes of providing power to an internal terminator) with 1.0 amp recommended current limiting (e.g., a fuse).

The use of keyed connectors is recommended in SCSI devices that provide terminator power to prevent accidental grounding or misconnection of terminator power.

SCSI devices that supply terminator power shall do so through a diode or similar semiconductor that prevents the backflow of power to the SCSI device.

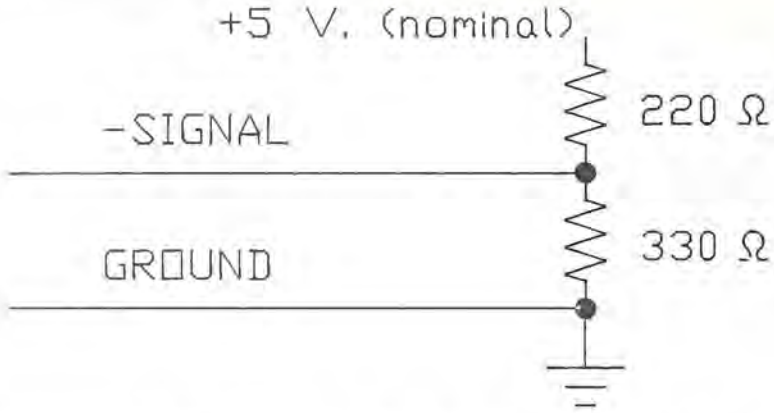


Figure 4-3. Termination for Single-Ended Devices

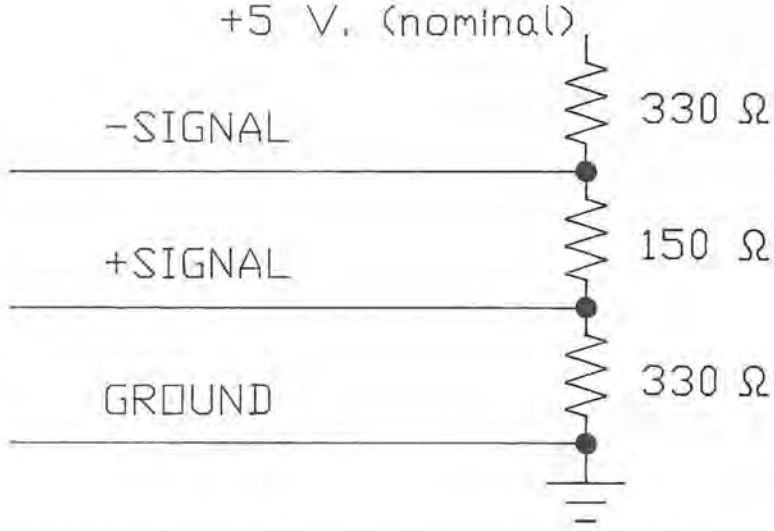


Figure 4-4. Termination for Differential Devices

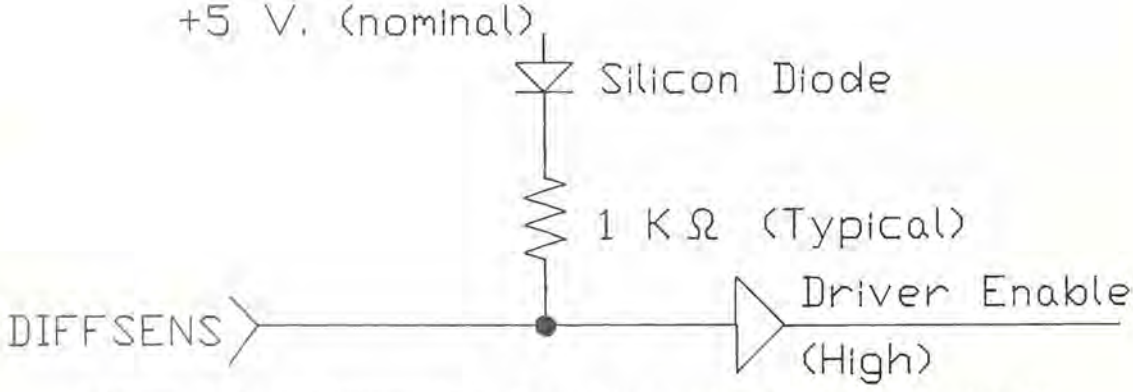


Figure 4-5. Differential Driver Protection Circuit (Optional)

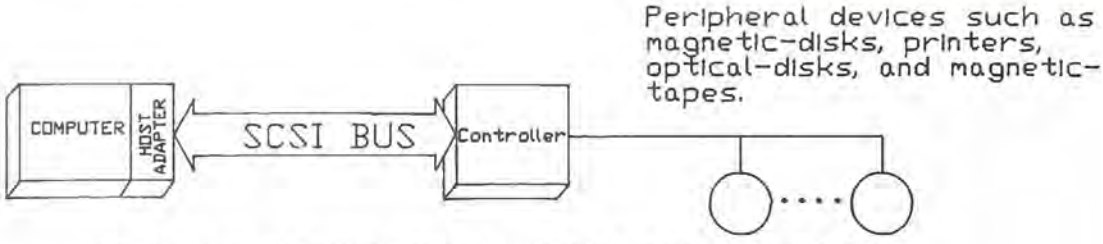
4.5 SCSI Bus. Communication on the SCSI bus is allowed between only two SCSI devices at any given time. There is a maximum of eight SCSI devices. Each SCSI device has an SCSI ID bit assigned as shown in Figure 4-6.

When two SCSI devices communicate on the SCSI bus, one acts as an initiator and the other acts as a target. The initiator originates an operation and the target performs the operation. An SCSI device usually has a fixed role as an initiator or target, but some devices may be able to assume either role.

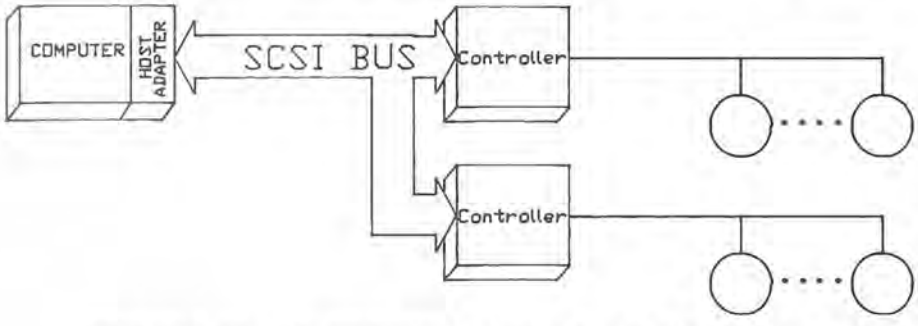
An initiator may address up to eight peripheral devices that are connected to a target. An option allows the addressing of up to 2,048 peripheral devices per target using extended messages. Three sample system configurations are shown in Figure 4-7.

```
DB(7) DB(6) DB(5) DB(4) DB(3) DB(2) DB(1) DB(0)  <-- DATA BUS
 :     :     :     :     :     :     :     :
 :     :     :     :     :     :     :     SCSI ID = 0
 :     :     :     :     :     :     :
 :     :     :     :     :     :     SCSI ID = 1
 :     :     :     :     :     :
 :     :     :     :     :     SCSI ID = 2
 :     :     :     :     :
 :     :     :     :     SCSI ID = 3
 :     :     :     :
 :     :     :     SCSI ID = 4
 :     :     :
 :     :     SCSI ID = 5
 :     :
 :     SCSI ID = 6
 :
SCSI ID = 7
```

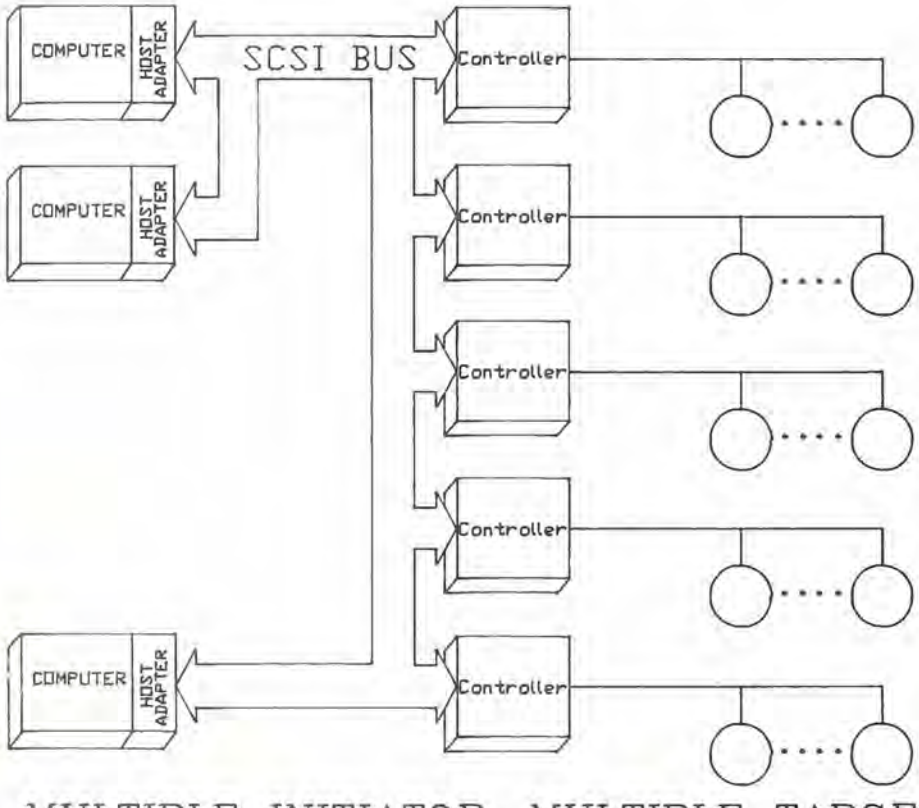
Figure 4-6. SCSI ID Bits



SINGLE INITIATOR, SINGLE TARGET



SINGLE INITIATOR, MULTIPLE TARGET



MULTIPLE INITIATOR, MULTIPLE TARGET

Figure 4-7. Sample SCSI Configurations

Up to eight SCSI devices can be supported on the SCSI bus. They can be any combination of initiators and targets.

Certain SCSI bus functions are assigned to the initiator and certain SCSI bus functions are assigned to the target. The initiator may arbitrate for the SCSI bus and select a particular target. The target may request the transfer of COMMAND, DATA, STATUS, or other information on the DATA BUS, and in some cases it may arbitrate for the SCSI bus and reselect an initiator for the purpose of continuing an operation.

Information transfers on the DATA BUS are asynchronous and follow a defined REQ/ACK handshake protocol. One byte of information may be transferred with each handshake. An option is defined for synchronous data transfer.

4.6 SCSI Bus Signals. There are a total of eighteen signals. Nine are used for control and nine are used for data. (Data signals include the parity signal option). These signals are described as follows:

BSY (BUSY). An "OR-tied" signal that indicates that the bus is being used.

SEL (SELECT). A signal used by an initiator to select a target or by a target to reselect an initiator.

C/D (CONTROL/DATA). A signal driven by a target that indicates whether CONTROL or DATA information is on the DATA BUS. True indicates CONTROL.

I/O (INPUT/OUTPUT). A signal driven by a target that controls the direction of data movement on the DATA BUS with respect to an initiator. True indicates input to the initiator. This signal is also used to distinguish between SELECTION and RESELECTION phases.

MSG (MESSAGE). A signal driven by a target during the MESSAGE phase.

REQ (REQUEST). A signal driven by a target to indicate a request for a REQ/ACK data transfer handshake.

ACK (ACKNOWLEDGE). A signal driven by an initiator to indicate an acknowledgment for a REQ/ACK data transfer handshake.

ATN (ATTENTION). A signal driven by an initiator to indicate the ATTENTION condition.

RST (RESET). An "OR-tied" signal that indicates the RESET condition.

DB(7-0,P) (DATA BUS). Eight data-bit signals, plus a parity-bit signal that form a DATA BUS. DB(7) is the most significant bit and has the highest priority during the ARBITRATION phase. Bit number, significance, and priority decrease downward to DB(0). A data bit is defined as one when the signal value is true and is defined as zero when the signal value is false.

Data parity DB(P) is odd. The use of parity is a system option (i.e., a system is configured so that all SCSI devices on a bus generate parity and have parity detection enabled, or all SCSI devices have parity detection disabled or not implemented). Parity is not valid during the ARBITRATION phase.

4.6.1 Signal Values. Signals may assume true or false values. There are two methods of driving these signals. In both cases, the signal shall be actively driven true, or asserted. In the case of OR-tied drivers, the driver does not drive the signal to the false state, rather the bias circuitry of the bus terminators pulls the signal false whenever it is released by the drivers at every SCSI device. If any driver is asserted, then the signal is true. In the case of non-OR-tied drivers, the signal may be actively driven false, or negated. In this standard, wherever the term negated is used, it means that the signal may be actively driven false, or may be simply released (in which case the bias circuitry pulls it false), at the option of the implementor. The advantage to actively drive signals false is that the transition from true to false occurs more quickly, and noise margins may be somewhat improved; this may permit somewhat faster data transfer.

4.6.2 OR-Tied Signals. The BSY and RST signals shall be OR-tied only. In the ordinary operation of the bus, these signals are simultaneously driven true by several drivers. No signals other than BSY, RST, and DB(P) are simultaneously driven by two or more drivers, and any signal other than BSY and RST may employ OR-tied or non-OR-tied drivers. DB(P) shall not be driven false during the ARBITRATION phase. There is no operational problem in mixing OR-tied and non-OR-tied drivers on signals other than BSY and RST.

4.6.3 Signal Sources. Table 4-3 indicates which type of SCSI device is allowed to source each signal. No attempt is made to show if the source is driving asserted, driving negated, or is passive. All SCSI device drivers that are not active sources shall be in the passive state. Note that the RST signal may be sourced by any SCSI device at any time.

Table 4-3
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	Initiator	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.

4.7 SCSI Bus Timing. Unless otherwise indicated, the delay-time measurements for each SCSI device, shown in 4.7.1 through 4.7.14, shall be calculated from signal conditions existing at that SCSI device's own SCSI bus connection. Thus, these measurements (except cable skew delay) can be made without considering delays in the cable.

4.7.1 Arbitration Delay (2.2 microseconds). The minimum time an SCSI device shall wait from asserting BSY for arbitration until the DATA BUS can be examined to see if arbitration has been won. There is no maximum time.

4.7.2 Assertion Period (90 nanoseconds). The minimum time that a target shall assert REQ while using synchronous data transfers. Also, the minimum time that an initiator shall assert ACK while using synchronous data transfers.

4.7.3 Bus Clear Delay (800 nanoseconds). The maximum time for an SCSI device to stop driving all bus signals after:

(1) The BUS FREE phase is detected (BSY and SEL both false for a bus settle delay)
(2) SEL is received from another SCSI device during the ARBITRATION phase
(3) The transition of RST to true.

NOTE: For the first condition above, the maximum time for an SCSI device to clear the bus is 1200 nanoseconds from BSY and SEL first becoming both false. If an SCSI device requires more than a bus settle delay to detect BUS FREE phase, it shall clear the bus within a bus clear delay minus the excess time.

4.7.4 Bus Free Delay (800 nanoseconds). The minimum time that an SCSI device shall wait from its detection of the BUS FREE phase (BSY and SEL both false for a bus settle delay) until its assertion of BSY when going to the ARBITRATION phase.

4.7.5 Bus Set Delay (1.8 microseconds). The maximum time for an SCSI device to assert BSY and its SCSI ID bit on the DATA BUS after it detects BUS FREE phase (BSY and SEL both false for a bus settle delay) for the purpose of entering the ARBITRATION phase.

4.7.6 Bus Settle Delay (400 nanoseconds). The time to wait for the bus to settle after changing certain control signals as called out in the protocol definitions.

4.7.7 Cable Skew Delay (10 nanoseconds). The maximum difference in propagation time allowed between any two SCSI bus signals when measured between any two SCSI devices.

4.7.8 Data Release Delay (400 nanoseconds). The maximum time for an initiator to release the DATA BUS signals following the transition of the I/O signal from false to true.

4.7.9 Deskew Delay (45 nanoseconds). The minimum time required for deskew of certain signals.

4.7.10 Hold Time (45 nanoseconds). The minimum time added between the

assertion of REQ or ACK and the changing of the data lines to provide hold time in the initiator or target, respectively, while using synchronous data transfers.

4.7.11 Negation Period (90 nanoseconds). The minimum time that a target shall negate REQ while using synchronous data transfers. Also, the minimum time that an initiator shall negate ACK while using synchronous data transfers.

4.7.12 Reset Hold Time (25 microseconds). The minimum time for which RST is asserted. There is no maximum time.

4.7.13 Selection Abort Time (200 microseconds). The maximum time that a target (or initiator) shall take from its most recent detection of being selected (or reselected) until asserting a BSY response. This timeout is required to ensure that a target (or initiator) does not assert BSY after a SELECTION (or RESELECTION) phase has been aborted. This is not the selection timeout period; see 5.1.3.5 and 5.1.4.2 for a complete description.

4.7.14 Selection Timeout Delay (250 milliseconds, recommended). The minimum time that an initiator (or target) should wait for a BSY response during the SELECTION (or RESELECTION) phase before starting the timeout procedure. Note that this is only a recommended time period. The specifications for the peripheral devices shall be consulted for the actual timing requirements.

4.7.15 Transfer Period (set during a MESSAGE phase). The Transfer Period specifies the minimum time allowed between the leading edges of successive REQ pulses and of successive ACK pulses while using synchronous data transfers. (See 5.1.5.2 and 5.5.5.)

5. Logical Characteristics

5.1 SCSI Bus Phases. The SCSI architecture includes eight distinct phases:

```
BUS FREE phase
ARBITRATION phase
SELECTION phase
RESELECTION phase
COMMAND phase  \
DATA phase      \   These phases are collectively termed the
STATUS phase    /   information transfer phases.
MESSAGE phase  /
```

The SCSI bus can never be in more than one phase at any given time. Unless otherwise noted in the following descriptions, signals that are not mentioned shall not be asserted.

5.1.1 BUS FREE Phase. The BUS FREE phase is used to indicate that no SCSI device is actively using the SCSI bus and that it is available for subsequent users.

SCSI devices shall detect the BUS FREE phase after SEL and BSY are both false for at least a bus settle delay. SCSI devices shall release all SCSI

bus signals within a bus clear delay after BSY and SEL become continuously false for a bus settle delay. If an SCSI device requires more than a bus settle delay to detect the BUS FREE phase then it shall release all SCSI bus signals within a bus clear delay minus the excess time to detect the BUS FREE phase. The total time to clear the SCSI bus shall not exceed a bus settle delay plus a bus clear delay.

5.1.2 ARBITRATION Phase. The ARBITRATION phase allows one SCSI device to gain control of the SCSI bus so that it can assume the role of an initiator or target.

NOTE: Implementation of the ARBITRATION phase is a system option. Systems that do not implement this option can have only one initiator. The ARBITRATION phase is required for systems that use the RESELECTION phase.

The procedure for an SCSI device to obtain control of the SCSI bus is as follows:

(1) The SCSI device shall first wait for the BUS FREE phase to occur. The BUS FREE phase is detected whenever both BSY and SEL are simultaneously and continuously false for a minimum of a bus settle delay. (Implementors Note: This bus settle delay is necessary because a transmission line phenomenon known as a "wire-OR glitch" may cause BSY to briefly appear false, even though it is being driven true.)

(2) The SCSI device shall wait a minimum of a bus free delay after detection of the BUS FREE phase (i.e. after BSY and SEL are both false for a bus settle delay) before driving any signal.

(3) Following the bus free delay in Step (2), the SCSI device may arbitrate for the SCSI bus by asserting both BSY and its own SCSI ID, however the SCSI device shall not arbitrate (i.e. assert BSY and its SCSI ID) if more than a bus set delay has passed since the BUS FREE phase was last observed. (Implementors Note: There is no maximum delay before asserting BSY and the SCSI ID following the bus free delay in Step (2) as long as the bus remains in the BUS FREE phase. However, SCSI devices that delay longer than a bus settle delay plus a bus set delay from the time when BSY and SEL first become false may fail to participate in arbitration when competing with faster SCSI devices.)

(4) After waiting at least an arbitration delay (measured from its assertion of BSY) the SCSI device shall examine the DATA BUS. If a higher priority SCSI ID bit is true on the DATA BUS (DB(7) is the highest), then the SCSI device has lost the arbitration and the SCSI device may release its signals and return to Step (1). If no higher priority SCSI ID bit is true on the DATA BUS, then the SCSI device has won the arbitration and it shall assert SEL. Any other SCSI device that is participating in the ARBITRATION phase has lost the arbitration and shall release BSY and its SCSI ID bit within a bus clear delay after SEL becomes true. An SCSI device that loses arbitration may return to Step (1).

(5) The SCSI device that wins arbitration shall wait at least a bus clear delay plus a bus settle delay after asserting SEL before changing any signals.

NOTE: The SCSI ID bit is a single bit on the DATA BUS that corresponds to the SCSI device's unique SCSI address. All other seven DATA BUS bits shall be released by the SCSI device. Parity is not valid during the ARBITRATION phase. During the ARBITRATION phase, DB(P) may be undriven or driven to the true state, but shall not be driven to the false state.

5.1.3 SELECTION Phase. The SELECTION phase allows an initiator to select a target for the purpose of initiating some target function (e.g., READ or WRITE command).

NOTE: During the SELECTION phase the I/O signal shall be negated so that this phase can be distinguished from the RESELECTION phase.

5.1.3.1 Nonarbitrating Systems. In systems with the ARBITRATION phase not implemented, the initiator shall first detect the BUS FREE phase and then wait a minimum of a bus clear delay. Then, except in certain single initiator environments with initiators employing the single initiator option (see 5.1.3.4), the initiator shall assert the desired target's SCSI ID and its own initiator SCSI ID on the DATA BUS. After two deskew delays the initiator shall assert SEL.

5.1.3.2 Arbitrating Systems. In systems with ARBITRATION phase implemented, the SCSI device that won the arbitration has both BSY and SEL asserted and has delayed at least a bus clear delay plus a bus settle delay before ending the ARBITRATION phase. The SCSI device that won the arbitration becomes an initiator by not asserting the I/O signal. Except in certain single initiator environments with initiators employing the single initiator option (see 5.1.3.4), the initiator shall set the DATA BUS to a value which is the OR of its SCSI ID bit and the target's SCSI ID bit. The initiator shall then wait at least two deskew delays and release BSY. The initiator shall then wait at least a bus settle delay before looking for a response from the target.

5.1.3.3 All Systems. In all systems, the target shall determine that it is selected when SEL and its SCSI ID bit are true and BSY and I/O are false for at least a bus settle delay. The selected target may examine the DATA BUS in order to determine the SCSI ID of the selecting initiator unless the initiator employed the single initiator option (see 5.1.3.4). The selected target shall then assert BSY within a selection abort time of its most recent detection of being selected; this is required for correct operation of the timeout procedure. In systems with parity implemented, the target shall not respond to a selection if bad parity is detected. Also, if more than two SCSI ID bits are on the DATA BUS, the target shall not respond to selection.

At least two deskew delays after the initiator detects BSY is true, it shall release SEL and may change the DATA BUS.

5.1.3.4 Single Initiator Option. Initiators that do not implement the RESELECTION phase and do not operate in the multiple initiator environment are allowed to set only the target's SCSI ID bit during the SELECTION phase. This makes it impossible for the target to determine the initiator's SCSI ID.

5.1.3.5 Selection Timeout Procedure. Two optional selection timeout procedures are specified for clearing the SCSI bus if the initiator waits a

minimum of a selection timeout delay and there has been no BSY response from the target:

(1) Optionally, the initiator shall assert the RST signal (see 5.2.2).
(2) Optionally, the initiator shall continue asserting SEL and shall release the DATA BUS. If the initiator has not detected BSY to be true after at least a selection abort time plus two deskew delays, the initiator shall release SEL allowing the SCSI bus to go to the BUS FREE phase. SCSI devices shall ensure that when responding to selection that the selection was still valid within a selection abort time of their assertion of BSY. Failure to comply with this requirement could result in an improper selection (two targets connected to the same initiator, wrong target connected to an initiator, or a target connected to no initiator).

5.1.4 RESELECTION Phase (Optional). RESELECTION is an optional phase that allows a target to reconnect to an initiator for the purpose of continuing some operation that was previously started by the initiator but was suspended by the target, (i.e., the target disconnected by allowing a BUS FREE phase to occur before the operation was complete).

5.1.4.1 RESELECTION. RESELECTION can only be used in systems that have ARBITRATION phase implemented.

Upon completing the ARBITRATION phase, the winning SCSI device has both BSY and SEL asserted and has delayed at least a bus clear delay plus a bus settle delay. The winning SCSI device becomes a target by asserting the I/O signal. The winning SCSI device shall also set the DATA BUS to a value that is the OR of its SCSI ID bit and the initiator's SCSI ID bit. The target shall wait at least two deskew delays and release BSY. The target shall then wait at least a bus settle delay before looking for a response from the initiator.

The initiator shall determine that it is reselected when SEL, I/O, and its SCSI ID bit are true and BSY is false for at least a bus settle delay. The reselected initiator may examine the DATA BUS in order to determine the SCSI ID of the reselecting target. The reselected initiator shall then assert BSY within a selection abort time of its most recent detection of being reselected; this is required for correct operation of the timeout procedure. In systems with parity implemented, the initiator shall not respond to a RESELECTION if bad parity is detected. Also, the initiator shall not respond to a RESELECTION if more than two SCSI ID bits are on the DATA BUS.

After the target detects BSY, it shall also assert BSY and wait at least two deskew delays and then release SEL. The target may then change the I/O signal and the DATA BUS. After the reselected initiator detects SEL false, it shall release BSY. The target shall continue asserting BSY until the target is ready to relinquish the SCSI bus.

NOTE: When the target is asserting BSY, a transmission line phenomenon known as a "wire-OR glitch" may cause BSY to appear false for up to a round-trip propagation delay following the release of BSY by the initiator. This is the reason why the BUS FREE phase is recognized only after both BSY and SEL are continuously false for a minimum of a bus settle delay. Cables longer than 25 meters should not be used even if the chosen driver, receiver, and cable provide adequate noise margins, because they increase the duration of the

glitch and could cause SCSI devices to inadvertently detect the BUS FREE phase.

5.1.4.2 RESELECTION Timeout Procedure. Two optional RESELECTION timeout procedures are specified for clearing the SCSI bus during a RESELECTION phase if the target waits a minimum of a selection timeout period and there has been no BSY response from the initiator:

(1) Optionally, the target shall assert the RST signal (see 5.2.2).
(2) Optionally, the target shall continue asserting SEL and I/O and shall release all DATA BUS signals. If the target has not detected BSY to be true after at least a selection abort time plus two deskew delays, the target shall release SEL and I/O allowing the SCSI bus to go to the BUS FREE phase. SCSI devices that respond to RESELECTION shall ensure that the RESELECTION was still valid within a selection abort time of their assertion of BSY. Failure to comply with this requirement could result in an improper reselection (two initiators connected to the same target or the wrong initiator connected to a target).

5.1.5 Information Transfer Phases.

NOTE: The COMMAND, DATA, STATUS, and MESSAGE phases are all grouped together as the information transfer phases because they are all used to transfer data or control information via the DATA BUS. The actual contents of the information is beyond the scope of this section.

The C/D, I/O, and MSG signals are used to distinguish between the different information transfer phases. (See Table 5-1.) The target drives these three signals and therefore controls all changes from one phase to another. The initiator can request a MESSAGE OUT phase by asserting ATN, while the target can cause the BUS FREE phase by releasing MSG, C/D, I/O, and BSY.

Table 5-1
Information Transfer Phases

Signal						
MSG	C/D	I/O	Phase Name	Direction Of Transfer		Comment
0	0	0	DATA OUT	Initiator to target	\	Data
0	0	1	DATA IN	Initiator from target	/	Phase
0	1	0	COMMAND	Initiator to target		
0	1	1	STATUS	Initiator from target		
1	0	0	*			
1	0	1	*			
1	1	0	MESSAGE OUT	Initiator to target	\	Message
1	1	1	MESSAGE IN	Initiator from target	/	Phase

Key: 0 = False, 1 = True, * = Reserved for future standardization.

The information transfer phases use one or more REQ/ACK handshakes to control the information transfer. Each REQ/ACK handshake allows the transfer of one byte of information. During the information transfer phases BSY shall remain true and SEL shall remain false. Additionally, during the information transfer phases, the target shall continuously envelope the REQ/ACK handshake(s) with C/D, I/O, and MSG in such a manner that these control signals are valid for a bus settle delay before the assertion of REQ of the first handshake and remain valid until the negation of ACK at the end of the last handshake.

5.1.5.1 Asynchronous Information Transfer. The target shall control the direction of information transfer by means of the I/O signal. When I/O is true, information shall be transferred from the target to the initiator. When I/O is false, information shall be transferred from the initiator to the target.

If I/O is true (transfer to the initiator), the target shall first drive DB(7-0,P) to their desired values, delay at least one deskew delay plus a cable skew delay, then assert REQ. DB(7-0,P) shall remain valid until ACK is true at the target. The initiator shall read DB(7-0,P) after REQ is true, then signal its acceptance of the data by asserting ACK. When ACK becomes true at the target, the target may change or release DB(7-0,P) and shall negate REQ. After REQ is false the initiator shall then negate ACK. After ACK is false the target may continue the transfer by driving DB(7-0,P) and asserting REQ, as described above.

If I/O is false (transfer to the target) the target shall request information by asserting REQ. The initiator shall drive DB(7-0,P) to their desired values, delay at least one deskew delay plus a cable skew delay and assert ACK. The initiator shall continue to drive DB(7-0,P) until REQ is false. When ACK becomes true at the target, the target shall read DB(7-0,P), then negate REQ. When REQ becomes false at the initiator, the initiator may change or release DB(7-0,P) and shall negate ACK. The target may continue the transfer by asserting REQ, as described above.

5.1.5.2 Synchronous Data Transfer (Optional). Synchronous data transfer is optional, and may be used only in the data phase if previously agreed to by the initiator and target through the message system (see SYNCHRONOUS DATA TRANSFER REQUEST message, 5.5.5). The messages determine the use of synchronous mode by both SCSI devices and establish a REQ/ACK offset and a transfer period.

The REQ/ACK offset specifies the maximum number of REQ pulses that can be sent by the target in advance of the number of ACK pulses received from the initiator, establishing a pacing mechanism. If the number of REQ pulses exceeds the number of ACK pulses by the REQ/ACK offset, the target shall not assert REQ until the next ACK pulse is received. A requirement for successful completion of the data phase is that the number of ACK and REQ pulses be equal.

The target shall assert the REQ signal for a minimum of an assertion period. The target shall wait at least the greater of a transfer period from the last transition of REQ to true or a minimum of a negation period from the last transition of REQ to false before asserting the REQ signal.

The initiator shall send one pulse on the ACK signal for each REQ pulse received. The initiator shall assert the ACK signal for a minimum of an assertion period. The initiator shall wait at least the greater of a transfer period from the last transition of ACK to true or for a minimum of a negation period from the last transition of ACK to false before asserting the ACK signal.

If I/O is true (transfer to the initiator), the target shall first drive DB(7-0,P) to their desired values, wait at least one deskew delay plus one cable skew delay, then assert REQ. DB(7-0,P) shall be held valid for a minimum of one deskew delay plus one cable skew delay plus one hold time after the assertion of REQ. The target shall assert REQ for a minimum of an assertion period. The target may then negate REQ and change or release DB(7-0,P). The initiator shall read the value on DB(7-0,P) within one hold time of the transition of REQ to true. The initiator shall then respond with an ACK pulse.

If I/O is false (transfer to the target), the initiator shall transfer one byte for each REQ pulse received. After receiving a REQ pulse, the initiator shall first drive DB(7-0,P) to their desired values, delay at least one deskew delay plus one cable skew delay, then assert ACK. The initiator shall hold DB(7-0,P) valid for at least one deskew delay plus one cable skew delay plus one hold time after the assertion of ACK. The initiator shall assert ACK for a minimum of an assertion period. The initiator may then negate ACK and may change or release DB(7-0,P). The target shall read the value of DB(7-0,P) within one hold time of the transition of ACK to true.

5.1.6 COMMAND Phase. The COMMAND phase allows the target to request command information from the initiator.

The target shall assert the C/D signal and negate the I/O and MSG signals during the REQ/ACK handshake(s) of this phase.

5.1.7 Data Phase. The data phase is a term that encompasses both the DATA IN phase and the DATA OUT phase.

5.1.7.1 DATA IN Phase. The DATA IN phase allows the target to request that data be sent to the initiator from the target.

The target shall assert the I/O signal and negate the C/D and MSG signals during the REQ/ACK handshake(s) of this phase.

5.1.7.2 DATA OUT Phase. The DATA OUT phase allows the target to request that data be sent from the initiator to the target.

The target shall negate the C/D, I/O, and MSG signals during the REQ/ACK handshake(s) of this phase.

5.1.8 STATUS Phase. The STATUS phase allows the target to request that status information be sent from the target to the initiator.

The target shall assert C/D and I/O and negate the MSG signal during the REQ/ACK handshake of this phase.

5.1.9 **Message Phase.** The message phase is a term that references either a MESSAGE IN, or a MESSAGE OUT phase. Multiple messages may be sent during either phase. The first byte transferred in either of these phases shall be either a single-byte message or the first byte of a multiple-byte message. Multiple-byte messages shall be wholly contained within a single message phase.

5.1.9.1 **MESSAGE IN Phase.** The MESSAGE IN phase allows the target to request that message(s) be sent to the initiator from the target.

The target shall assert C/D, I/O, and MSG during the REQ/ACK handshake(s) of this phase.

5.1.9.2 **MESSAGE OUT Phase.** The MESSAGE OUT phase allows the target to request that message(s) be sent from the initiator to the target. The target may invoke this phase at its convenience in response to the ATTENTION condition (see 5.2.1) created by the initiator.

The target shall assert C/D and MSG and negate I/O during the REQ/ACK handshake(s) of this phase. The target shall handshake byte(s) in this phase until ATN goes false, unless an error occurs (see MESSAGE REJECT, 5.5.2).

If the target detects one or more parity error(s) on the message byte(s) received, it may indicate its desire to retry the message(s) by asserting REQ after detecting ATN has gone false and prior to changing to any other phase. The initiator, upon detecting this condition, shall resend all of the previous message byte(s) sent during this phase. When resending more than one message byte, the initiator shall assert ATN prior to asserting ACK on the first byte and shall maintain ATN asserted until the last byte is sent as described in 5.2.1.

If the target receives all of the message byte(s) successfully (i.e., no parity errors), it shall indicate that it does not wish to retry by changing to any information transfer phase other than the MESSAGE OUT phase and transfer at least one byte. The target may also indicate that it has successfully received the message byte(s) by changing to the BUS FREE phase (e.g., ABORT or BUS DEVICE RESET messages).

5.1.10 **Signal Restrictions between Phases.** When the SCSI bus is between two information transfer phases, the following restrictions shall apply to the SCSI bus signals:

(1) The BSY, SEL, REQ, and ACK signals shall not change.
(2) The C/D, I/O, MSG, and DATA BUS signals may change. When switching the DATA BUS direction from out (initiator driving) to in (target driving), the target shall delay driving the DATA BUS by at least a data release delay plus a bus settle delay after asserting the I/O signal and the initiator shall release the DATA BUS no later than a data release delay after the transition of the I/O signal to true. When switching the DATA BUS direction from in (target driving) to out (initiator driving), the target shall release the DATA BUS no later than a deskew delay after negating the I/O signal.
(3) The ATN and RST signals may change as defined under the descriptions for the ATTENTION condition (5.2.1) and RESET condition (5.2.2).

5.2 SCSI Bus Conditions. The SCSI bus has two asynchronous conditions; the ATTENTION condition and the RESET condition. These conditions cause the SCSI device to perform certain actions and can alter the phase sequence.

5.2.1 ATTENTION Condition. The ATTENTION condition allows an initiator to inform a target that the initiator has a message ready. The target may get this message at its convenience by performing a MESSAGE OUT phase.

The initiator creates the ATTENTION condition by asserting ATN at any time except during the ARBITRATION or BUS FREE phases.

The target may respond with the MESSAGE OUT phase.

The initiator shall keep ATN asserted if more than one byte is to be transferred. The initiator may negate the ATN signal at any time except it shall not negate the ATN signal while the ACK signal is asserted during a MESSAGE OUT phase. Normally, the initiator negates ATN while REQ is true and ACK is false during the last REQ/ACK handshake of the MESSAGE OUT phase.

5.2.2 RESET Condition. The RESET condition is used to immediately clear all SCSI devices from the bus. This condition shall take precedence over all other phases and conditions. Any SCSI device may create the RESET condition by asserting RST for a minimum of a reset hold time. During the RESET condition, the state of all SCSI bus signals other than RST is not defined.

All SCSI devices shall release all SCSI bus signals (except RST) within a bus clear delay of the transition of RST to true. The BUS FREE phase always follows the RESET condition.

The effect of the RESET condition on uncompleted commands, SCSI device reservations, and SCSI device operating modes is determined by whether the SCSI device has implemented the "hard" RESET option or the "soft" RESET option (one of which shall be implemented) as defined in 5.2.2.1 and 5.2.2.2.

5.2.2.1 "Hard" RESET Option. SCSI devices that implement the "hard" RESET option, upon detection of the RESET condition, shall:

(1) Clear all uncompleted commands
(2) Release all SCSI device reservations
(3) Return any SCSI device operating modes (MODE SELECT, PREVENT/ALLOW MEDIUM REMOVAL commands, etc) to their default conditions.

5.2.2.2 "Soft" RESET Option. SCSI devices that implement the "soft" RESET option, upon detection of the RESET condition, shall:

(1) Attempt to complete any uncompleted commands that were fully identified
(2) Preserve all SCSI device reservations
(3) Preserve any SCSI device operating modes (MODE SELECT, PREVENT/ALLOW MEDIUM REMOVAL commands, etc)

The "soft" RESET option allows a single initiator to reset the SCSI bus without disturbing the operation of other initiators in a multiple initiator system. To ensure proper operation the following conditions shall be met:

(1) An initiator shall not consider a command to be fully identified until the IDENTIFY message is sent to the target and the target responds by changing to any other information transfer phase and requests that at least one byte be transferred.

(2) A target shall consider a command to be fully identified when it successfully receives the IDENTIFY message.

(3) If an initiator selects a logical unit for which there already is an active command for the same initiator, the target shall clear the original command and perform the new command.

(4) If a target reselects an initiator to continue a command for which the initiator has no record, the initiator shall abort that command by sending the ABORT message.

(5) An initiator shall consider a command to be completed when it negates ACK for a successfully received COMMAND COMPLETE message.

(6) A target shall consider a command to be completed when it detects the false transition of ACK for the COMMAND COMPLETE message with the ATN signal false.

(7) An initiator shall not negate ACK for the SAVE DATA POINTER message until it has actually saved the data pointer for the operation.

(8) A target shall consider the data pointer to be saved when it detects the false transition of ACK for the SAVE DATA POINTER message with the ATN signal false.

(9) If the RESET condition occurs between the time that the target asserts REQ for the SAVE DATA POINTER message and it detects the false transition of ACK, the target shall terminate the command with a CHECK CONDITION status. If extended sense is implemented, the target shall set the sense key to ABORTED COMMAND. This is necessary because the target cannot determine whether the data pointer has actually been saved.

NOTE: If the ATN signal is true in conditions (6) or (8), the target would normally switch to MESSAGE OUT phase and attempt to transfer a message byte. If the RESET condition occurs before it is able to successfully receive the message byte, the target shall assume that the initiator may not have successfully received the COMMAND COMPLETE message or the SAVE DATA POINTER message. In the case of COMMAND COMPLETE message, the target shall reselect the initiator and attempt to send the COMMAND COMPLETE message again. In the case of the SAVE DATA POINTER message, the target shall reselect the initiator and terminate the command as described in condition (9).

5.3 SCSI Bus Phase Sequences. The order in which phases are used on the SCSI bus follows a prescribed sequence.

In all systems, the RESET condition can abort any phase and is always followed by the BUS FREE phase. Also, any other phase can be followed by the BUS FREE phase.

5.3.1 **Nonarbitrating Systems.** In systems where the ARBITRATION phase is not implemented, the allowable sequences shall be as shown in Figure 5-1. The normal progression is from the BUS FREE phase to SELECTION, and from SELECTION to one or more of the information transfer phases (COMMAND, DATA, STATUS, or MESSAGE).

5.3.2 **Arbitrating Systems.** In systems where the ARBITRATION phase is implemented, the allowable sequences shall be as shown in Figure 5-2. The normal progression is from the BUS FREE phase to ARBITRATION, from ARBITRATION to SELECTION or RESELECTION, and from SELECTION or RESELECTION to one or more of the information transfer phases (COMMAND, DATA, STATUS, or MESSAGE).

5.3.3 **All Systems.** There are no restrictions on the sequences between information transfer phases. A phase type may even be followed by the same phase type (e.g., a data phase may be followed by another data phase).

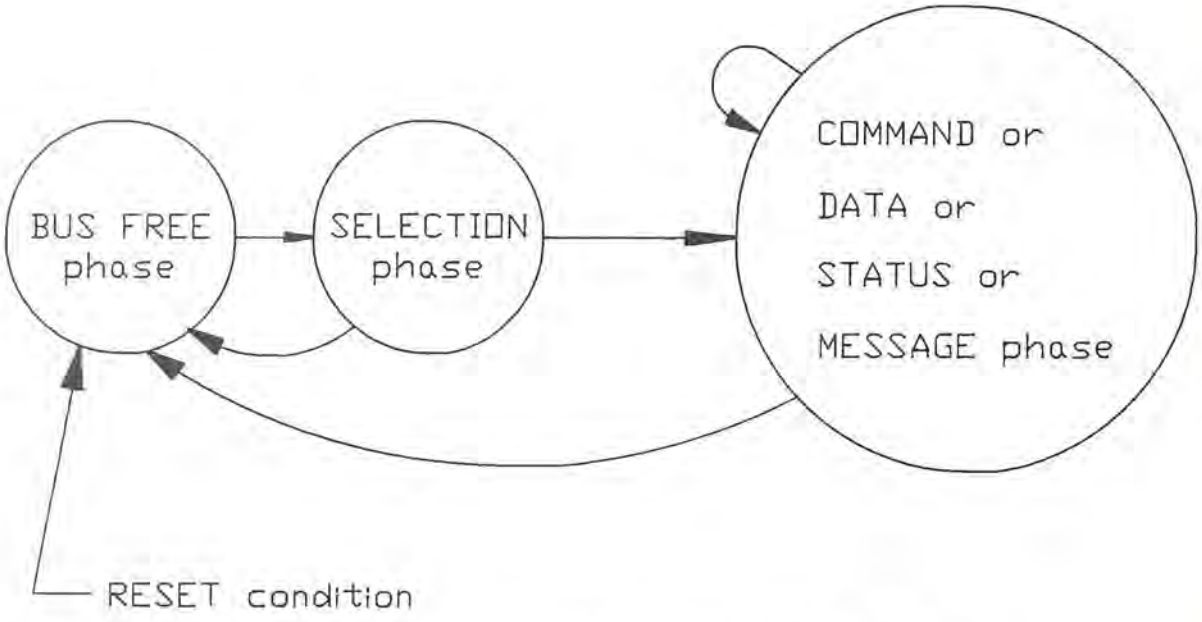


Figure 5-1. Phase Sequences without Arbitration

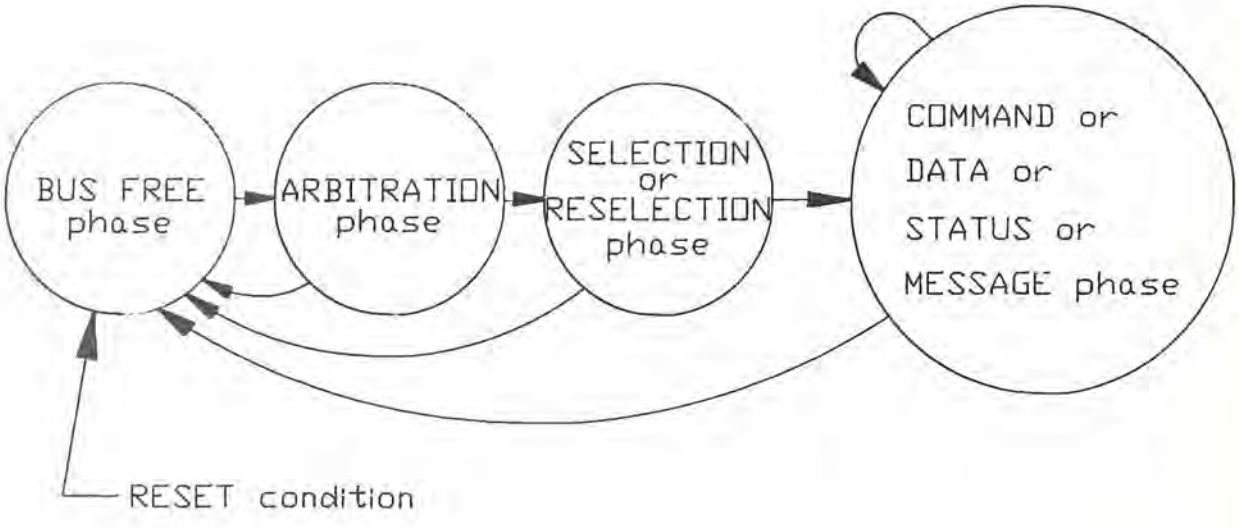


Figure 5-2. Phase Sequences with Arbitration

5.4 SCSI Pointers. Consider the system shown in Figure 5-3 in which an initiator and target communicate on the SCSI bus in order to execute a command.

```
-------------------------                  -------------------------
: Function : : Initiator:-----------------: Target   : : Function :
: Origin   : : Path     :    SCSI BUS     : Path     : : Execution:
:          : : Control  :-----------------: Control  : :          :
-------------------------                  -------------------------

        Initiator                                     Target
```

Figure 5-3. Simplified SCSI System

The SCSI architecture provides for two sets of three pointers within each initiator. The pointers reside in the initiator path control. The first set of pointers are known as the current (or active) pointers. These pointers are used to represent the state of the interface and point to the next command, data, or status byte to be transferred between the initiator's memory and the target. There is only one set of current pointers in each initiator. The current pointers are used by the target currently connected to the initiator.

The second set of pointers are known as the saved pointers. There is one set of saved pointers for each command that is currently active (whether or not it is currently connected). The saved command pointer always points to the start of the command descriptor block (see 6.2) for the current command. The saved status pointer always points to the start of the status area for the current command. At the beginning of each command, the saved data pointer points to the start of the data area. It remains at this value until the target sends a SAVE DATA POINTER message (see 5.5.2) to the initiator. In response to this message, the initiator stores the value of the current data pointer into the saved data pointer. The target may restore the current pointers to their saved values by sending a RESTORE POINTERS message (see 5.5.2) to the initiator. The initiator moves the saved value of each pointer into the corresponding current pointer. Whenever an SCSI device disconnects from the bus, only the saved pointer values are retained. The current pointer values are restored from the saved values upon the next reconnection.

5.5 Message System Specification. The message system allows communication between an initiator and target for the purpose of physical path management.

5.5.1 Message Protocol. All SCSI devices shall implement the COMMAND COMPLETE message. A functional SCSI device can be constructed without using any of the other messages if the logical unit number is specified in the command descriptor block. The remainder of this section deals with the additional requirements on SCSI devices that support messages other than COMMAND COMPLETE.

SCSI devices indicate their ability to accommodate more than the COMMAND COMPLETE message by asserting or responding to the ATN signal. The initiator indicates this in the SELECTION phase by asserting ATN prior to the SCSI bus

condition of SEL true, and BSY false. The target indicates its ability to accommodate more messages by responding to the ATTENTION condition with the MESSAGE OUT phase after going through the SELECTION phase.

For SCSI devices that support messages other than COMMAND COMPLETE, the first message sent by the initiator after the SELECTION phase shall be the IDENTIFY message. This allows the establishment of the physical path for a particular logical unit specified by the initiator. After the RESELECTION phase, the target's first message shall be IDENTIFY. This allows the physical path to be reestablished for the target's specified logical unit number. Under some exceptional conditions, an initiator may send the ABORT message or the BUS DEVICE RESET message instead of the IDENTIFY message, as the first message. Only one logical unit number shall be identified for any one selection sequence; a second IDENTIFY message with a new logical unit number shall not be issued before the SCSI bus has been released (BUS FREE phase).

Whenever a physical path is established in an initiator that can accommodate disconnection and reconnection, the initiator shall ensure that the active pointers of the physical path are equal to the saved pointers for that particular logical unit number. (An implied restore pointers operation occurs as a result of connect or reconnect.)

SCSI devices that implement any message other than the COMMAND COMPLETE message shall also implement the MESSAGE REJECT message.

Table 5-2
Message Codes

Code	Type	Description	Direction	
00h	M	COMMAND COMPLETE	In	
01h	O	EXTENDED MESSAGE	In	Out
02h	O	SAVE DATA POINTER	In	
03h	O	RESTORE POINTERS	In	
04h	O	DISCONNECT	In	
05h	O	INITIATOR DETECTED ERROR		Out
06h	O	ABORT		Out
07h	O	MESSAGE REJECT	In	Out
08h	O	NO OPERATION		Out
09h	O	MESSAGE PARITY ERROR		Out
0Ah	O	LINKED COMMAND COMPLETE	In	
0Bh	O	LINKED COMMAND COMPLETE (WITH FLAG)	In	
0Ch	O	BUS DEVICE RESET		Out
0Dh – 7Fh	R	Reserved Codes		
80h – FFh	O	IDENTIFY	In	Out

Key: In = Target to initiator, Out = Initiator to target.

5.5.2 Messages. The single byte messages (Table 5-2) are listed along with their code values and their definitions.

COMMAND COMPLETE 00h (Mandatory). 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. After successfully sending this message, the target shall go to the BUS FREE phase by releasing BSY.

NOTE: The command may have been executed successfully or unsuccessfully as indicated in the status.

EXTENDED MESSAGE 01h (Optional). This message is sent from either the initiator or the target as the first byte of a multiple-byte message. (See 5.5.3 for descriptions of extended messages.)

SAVE DATA POINTER 02h (Optional). This message is sent from a target to direct the initiator to save a copy of the present active data pointer for the currently attached logical unit. (See 5.4 for a definition of pointers.)

RESTORE POINTERS 03h (Optional). 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 the 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 in the absence of a SAVE DATA POINTER message or to the value at the point at which the last SAVE DATA POINTER message occurred for that logical unit.

DISCONNECT 04h (Optional). 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 by releasing BSY), but that a later reconnect will be required in order to complete the current operation. If the initiator detects the BUS FREE phase (other than as a result of a RESET condition) without first receiving a DISCONNECT or COMMAND COMPLETE message, the initiator shall consider this as a catastrophic error condition. If the target intentionally creates this condition, the target shall clear the current command. This message shall not cause the initiator to save the data pointer. Note: If DISCONNECT messages are used 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 05h (Optional). This message is sent from an initiator to inform a target that an error (e.g., parity error) has occurred that does not preclude the target from retrying the operation. Although present pointer integrity is not assured, a RESTORE POINTERS message or a disconnect followed by a reconnect, shall cause the pointers to be restored to their defined prior state.

ABORT 06h (Optional). This message is sent from the initiator to the target to clear the present operation. If a logical unit has been identified, all pending data and status for the issuing initiator from the effected logical unit shall be cleared, and the target shall go to the BUS FREE phase. Pending

data and status for other initiators shall not be cleared. If a logical unit has not been identified, the target shall go to the BUS FREE phase. No status or ending message shall be sent for the operation. It is not an error to issue this message to an logical unit that is not currently performing an operation for the initiator.

MESSAGE REJECT 07h (Optional). This message is sent from either the initiator or target to indicate that the last message it received was inappropriate or has not been implemented.

In order to indicate its intentions of sending this message, the initiator shall assert the ATN signal prior to its release of ACK for the REQ/ACK handshake of the message that is to be rejected. When a target sends this message, it shall change to MESSAGE IN phase and send this message prior to requesting additional message bytes from the initiator. This provides an interlock so that the initiator can determine which message is rejected.

This message shall be implemented if any other optional messages are implemented.

NO OPERATION 08h (Optional). 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 09h (Optional). 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.

In order to indicate its intentions of sending this message, the initiator shall assert the ATN signal prior to its release of ACK for the REQ/ACK handshake of the message that has the parity error. This provides an interlock so that the target can determine which message has the parity error.

LINKED COMMAND COMPLETE 0Ah (Optional). This message is 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 shall then set the pointers to the initial state for the next linked command.

LINKED COMMAND COMPLETE (WITH FLAG) 0Bh (Optional). This message is sent from a target to an initiator to indicate that the execution of a linked command (with the flag bit set to one) has completed and that status has been sent. The initiator shall then set the pointers to the initial state of the next linked command. Typically this message would be used to cause an interrupt in the initiator between two linked commands.

BUS DEVICE RESET 0Ch (Optional). This message is sent from an initiator to direct a target to clear all current commands on that SCSI device. This message forces the SCSI device to an initial state with no operations pending for any initiator. Upon recognizing this message, the target shall go to the BUS FREE phase.

Reserved 0Dh to 7Fh. These message codes are reserved for future standardization.

IDENTIFY 80h to FFh (Optional). These messages are sent by either the initiator or the target to establish the physical path connection between an initiator and target for a particular logical unit.

Bit 7. This bit is always set to one to distinguish these messages from the other messages.

Bit 6. This bit is only set to one by the initiator. When set to one, it indicates that the initiator has the ability to accommodate disconnection and reconnection.

Bits 5-3. Reserved.

Bits 2-0. These bits specify a logical unit number in a target.

Only one logical unit number shall be identified for any one selection sequence; a second IDENTIFY message with a new logical unit number shall not be issued before the bus has been released (BUS FREE phase).

When sent from a target to an initiator during reconnection, an implied RESTORE POINTERS message shall be performed by the initiator prior to completion of this message.

5.5.3 Extended Messages (Optional). A value of one in the first byte of a message indicates the beginning of a multiple-byte extended message. The minimum number of bytes sent for an extended message is three. The extended message format and the extended message codes are shown in Tables 5-3 and 5-4, respectively.

Table 5-3
Extended Message Format

Byte	Value	Description
0	01h	Extended message
1	n	Extended message length
2	y	Extended message code
3 – n+1	x	Extended message arguments

The extended message length specifies the length in bytes of the extended message code plus the extended message arguments to follow. Therefore, the total length of the message is equal to the extended message length plus two. A value of zero for the extended message length indicates 256 bytes follow.

The extended message codes are listed in Table 5-4. The extended message arguments are specified for the defined extended messages in 5.5.4 through 5.5.6.

Table 5-4
Extended Message Codes

Code (y)	Description
00h	MODIFY DATA POINTER (Optional)
01h	SYNCHRONOUS DATA TRANSFER REQUEST (Optional)
02h	EXTENDED IDENTIFY (Optional)
03h – 7Fh	Reserved
80h – FFh	Vendor Unique

5.5.4 MODIFY DATA POINTER Message (Optional)

Table 5-5
MODIFY DATA POINTER

Byte	Value	Description
0	01h	Extended message
1	05h	Extended message length
2	00h	MODIFY DATA POINTER code
3	x	Argument (MSB)
4	x	Argument
5	x	Argument
6	x	Argument (LSB)

The MODIFY DATA POINTER message (Table 5-5) is sent from the target to the initiator and requests that the signed argument be added (two's complement) to the value of the current data pointer.

5.5.5 SYNCHRONOUS DATA TRANSFER REQUEST Message (Optional)

Table 5-6
SYNCHRONOUS DATA TRANSFER REQUEST

Byte	Value	Description
0	01h	Extended message
1	03h	Extended message length
2	01h	SYNCHRONOUS DATA TRANSFER REQUEST code
3	m	Transfer period (m times 4 nanoseconds)
4	x	REQ/ACK offset

A pair of SYNCHRONOUS DATA TRANSFER REQUEST messages (Table 5-6) are exchanged between an initiator and a target whenever an SCSI device that can support synchronous data transfer recognizes that it has not communicated with the other SCSI device since receiving the last "hard" RESET condition or a BUS DEVICE RESET message. The SCSI devices may also exchange messages to establish synchronous data transfer when requested to do so. The message exchange establishes the transfer period and the REQ/ACK offset. The transfer period is the minimum time between leading edges of successive REQ pulses and of successive ACK pulses.

The REQ/ACK offset is the maximum number of REQ pulses that may be outstanding before its corresponding ACK pulse is received at the target. A REQ/ACK offset value of zero shall indicate asynchronous mode; a value of FFh shall indicate unlimited offset.

If the initiator recognizes that negotiation is required, it asserts ATN and, if the target implements message transfers, sends a SYNCHRONOUS DATA TRANSFER REQUEST message indicating an REQ/ACK offset and minimum transfer period. The REQ/ACK offset is chosen to prevent initiator buffer overflows, while the minimum transfer period is chosen to meet the data handling requirements of the initiator. The target responds in any of the following ways:

Target Response	Implied Agreement
(1) REQ/ACK offset less than or equal to the requested value.	REQ/ACK offset equal to target value.
Minimum transfer period equal to or greater than requested period.	Minimum transfer period equal to the target value.
(2) REQ/ACK offset equal to zero.	Asynchronous transfer.
(3) MESSAGE REJECT.	Asynchronous transfer.

If the target recognizes that negotiation is required, it sends a SYNCHRONOUS DATA TRANSFER REQUEST message to the initiator. The REQ/ACK offset is selected to prevent buffer and offset counter overflows, while the minimum transfer period is chosen to meet the data handling requirements of the target. The initiator responds in any of the following ways if the target chooses an REQ/ACK offset equal to FFh:

Initiator Response	Implied Agreement
(1) REQ/ACK offset equal to FFh. Minimum transfer period equal to or greater than requested period.	REQ/ACK offset unlimited. Minimum transfer period equal to the initiator value.
(2) REQ/ACK offset equal to 00h.	Asynchronous transfer. The target may renegotiate for an REQ/ACK offset less than FFh and greater than 00h.
(3) MESSAGE REJECT.	Asynchronous transfer.

The initiator responds in any of the following ways if the target selects an REQ/ACK offset less than FFh:

Initiator Response	Implied Agreement
(1) REQ/ACK offset less than or equal to the requested value.	REQ/ACK offset equals initiator value.
Minimum transfer period equal to or greater than requested value.	Minimum transfer period equal to the initiator value.
(2) REQ/ACK offset equal to zero.	Asynchronous transfer.
(3) MESSAGE REJECT.	Asynchronous transfer.

The implied agreement shall remain in effect until a BUS DEVICE RESET message is received, until a "hard" RESET condition occurs, or until one of the two SCSI devices elects to modify the agreement. Renegotiation at every selection is not recommended, since a significant performance impact is likely. The default mode of data transfer is asynchronous mode. The default mode is entered at power on, after a BUS DEVICE RESET message, or after a "hard" RESET condition. The SYNCHRONOUS DATA TRANSFER REQUEST message exchange can only take place following a SELECTION phase that includes the SCSI IDs for both the initiator and the target. Violation of this rule may make data transfer impossible owing to disagreements among SCSI devices about the data transfer mode.

5.5.6 EXTENDED IDENTIFY Message (Optional)

Table 5-7
EXTENDED IDENTIFY

Byte	Value	Description
0	01h	Extended message
1	02h	Extended message length
2	02h	EXTENDED IDENTIFY code
3	x	Sub-logical unit number

The EXTENDED IDENTIFY message (Table 5-7) is optional and may be sent by a target or an initiator. It may be used in conjunction with the normal IDENTIFY message in order to expand the logical unit number addressing in a target. The sub-logical unit number specifies the encoded eight-bit sub-logical unit number used to identify one of 256 sub-logical units within the logical unit. This allows up to 2048 units to be addressed on a single target.

6. SCSI Commands

This section defines the SCSI command structure and gives several examples.

The command definitions assume a data structure providing the appearance at the interface of a contiguous set of logical blocks of a fixed or explicitly defined data length. The SCSI device maps the physical characteristics of the attached peripheral devices to one of several logical structures defined by the device type code.

A single command may transfer one or more logical blocks of data. Multiple commands may be linked if they are sent to the same logical unit. A target may disconnect from the SCSI bus to allow activity by other SCSI devices while a logical unit is being prepared to transfer data.

Upon command completion (successful or unsuccessful), the target returns a status byte to the initiator. Since most error and exception conditions cannot be adequately described with a single status byte, one status code, CHECK CONDITION, indicates that additional information is available. The initiator may issue a REQUEST SENSE command to retrieve this additional information.

By keeping to a minimum the functions essential to communicate via this protocol, a wide range of peripheral devices of varying capability can operate in the same environment.

Because subsets of the full architecture may be implemented, optional functions are noted.

6.1 Command Implementation Requirements. The first byte of any SCSI command shall contain an operation code as defined in this document. Three bits (bits 7 - 5) of the second byte of each SCSI command specify the logical unit if it is not specified using the IDENTIFY message (see 5.5.2). The last byte of all SCSI commands shall contain a control byte as defined in 6.2.6.

6.1.1 Reserved. Reserved bits, fields, bytes, and code values are set aside for future standardization. Their use and interpretation will be specified by future extensions to this standard. A reserved bit, field, or byte shall be set to zero, or in accordance with a future extension to this standard. A target that receives a reserved bit, field, or byte that is not zero or receives a reserved code value shall terminate the command with a CHECK CONDITION status and, if extended sense is implemented, the sense key shall be set to ILLEGAL REQUEST. It shall also be acceptable for a target to interpret the bit, field, byte, or code value in accordance with a future extension to this standard.

6.1.2 Operation Code Types

Operation Code Type	Description
M	Mandatory - Commands so designated shall be implemented in order to meet the minimum requirement of this standard.
E	Extended - Commands so designated shall be implemented in addition to mandatory commands to meet the extended requirement of this standard.
O	Optional - Commands so designated, if implemented, shall be implemented as defined in this standard.
V	Vendor unique - Operation codes so designated are available for vendor defined commands. See the vendor specifications where compatibility is desired.
R	Reserved - Operation codes so designated shall not be used. They are reserved for future extensions to this standard.

6.1.3 Unit Attention Condition. A unit attention condition for a logical unit shall begin for each initiator whenever the removable medium may have been changed or the target has been reset (by a BUS DEVICE RESET message or a "hard" RESET condition). The unit attention condition shall persist for each initiator until that initiator issues a command to the logical unit other than REQUEST SENSE or INQUIRY for which the target shall return CHECK CONDITION status. If the next command from that initiator to the logical unit (following the CHECK CONDITION status) is REQUEST SENSE, and if the target supports extended sense, then the UNIT ATTENTION sense key shall be returned. (If any command other than REQUEST SENSE is received, the unit attention condition is lost.)

If an INQUIRY command is received from an initiator with a pending unit attention condition (before the target reports CHECK CONDITION status), the target shall perform the INQUIRY command and shall not clear the unit

attention condition.

If a REQUEST SENSE command is received from an initiator with a pending unit attention condition (before the target reports CHECK CONDITION status), then the target may either:

(1) Report any pending sense data and preserve the unit attention condition
(2) Discard any pending sense data, report UNIT ATTENTION sense key, and clear the unit attention condition for that initiator.

If an initiator issues a command other than INQUIRY or REQUEST SENSE while a unit attention condition exists for that initiator, the target shall not perform the command and shall report CHECK CONDITION status.

6.2 Command Descriptor Block. A request to a peripheral device is performed by sending a command descriptor block to the target. For several commands, the request is accompanied by a list of parameters sent during the DATA OUT phase. See the specific commands for detailed information.

The command descriptor block always has an operation code as the first byte of the command. This is followed by a logical unit number, command parameters (if any), and a control byte.

For all commands, if there is an invalid parameter in the command descriptor block, then the target shall terminate the command without altering the medium.

6.2.1 Operation Code. The operation code (Table 6-1) of the command descriptor block has a group code field and a command code field. The three-bit group code field provides for eight groups of command codes. The five-bit command code field provides for thirty-two command codes in each group. Thus, a total of 256 possible operation codes exist. Operation codes are defined in Sections 7 through 13.

The group code specifies one of the following groups:

Group 0 - six-byte commands (see Table 6-2)
Group 1 - ten-byte commands (see Table 6-3)
Group 2 - reserved
Group 3 - reserved
Group 4 - reserved
Group 5 - twelve-byte commands (see Table 6-4)
Group 6 - vendor unique
Group 7 - vendor unique

Table 6-1
Operation Code

Bit / Byte	7	6	5	4	3	2	1	0
0	Group Code			Command Code				

Table 6-2
Typical Command Descriptor Block for Six-Byte Commands

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Logical Block Address (if required) (MSB)				
2	Logical Block Address (if required)							
3	Logical Block Address (if required) (LSB)							
4	Transfer Length (if required)							
5	Control Byte							

Table 6-3
Typical Command Descriptor Block for Ten-Byte Commands

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

Table 6-4
Typical Command Descriptor Block for Twelve-Byte Commands

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				RelAdr
2	Logical Block Address (if required) (MSB)							
3	Logical Block Address (if required)							
4	Logical Block Address (if required)							
5	Logical Block Address (if required) (LSB)							
6	Reserved							
7	Reserved							
8	Reserved							
9	Transfer Length (if required) (MSB)							
10	Transfer Length (if required) (LSB)							
11	Control Byte							

6.2.2 **Logical Unit Number.** The logical unit number addresses one of up to eight physical or virtual devices attached to a target. This method of addressing is provided for systems that do not implement the IDENTIFY message. A target that accepts an IDENTIFY message shall use the logical unit number specified within the message. In this case, the target shall ignore the logical unit number specified within the command descriptor block. (Implementors note: It is a good practice for initiators that implement the IDENTIFY message to specify the same logical unit number in the command descriptor block.)

6.2.3 **Logical Block Address.** The logical block address on logical units shall begin with block zero and be contiguous up to the last logical block on that logical unit.

Group 0 command descriptor blocks contain 21-bit logical block addresses. Groups 1 and 5 command descriptor blocks contain 32-bit logical block addresses.

The logical block concept implies that the initiator and target shall have previously established the number of data bytes per logical block. This may be established through the use of the READ CAPACITY command or the MODE SENSE command or by prior arrangement.

6.2.4 Relative Address Bit. The relative address (RelAdr) bit of the group 1 and group 5 commands is set to one to indicate that the logical block address portion of the command descriptor block is a two's complement displacement. This negative or positive displacement is to be added to the logical block address last accessed on the logical unit to form the logical block address for this command. This feature is only available when linking commands. The feature requires that a previous command in the linked group have accessed a block of data on the logical unit. (For an example of the operation of this function, see 6.3.3.)

6.2.5 Transfer Length. The transfer length specifies the amount of data to be transferred, usually the number of blocks. For several commands the transfer length indicates the requested number of bytes to be sent as defined in the command description. For these commands the transfer length field may be identified by a different name. See the following descriptions and the individual command descriptions for further information.

Commands that use one byte for transfer length allow up to 256 blocks of data to be transferred by one command. A transfer length value of 1 to 255 indicates the number of blocks that shall transferred. A value of zero indicates 256 blocks.

Commands that use two bytes for transfer length allow up to 65,535 blocks of data to be transferred by one command. In this case, a transfer length of zero indicates that no data transfer shall take place. A value of 1 to 65,535 indicates the number of blocks that shall be transferred.

For several commands more than two bytes are allocated for transfer length. Refer to the specific command description for further information.

The transfer length of the commands that are used to send a list of parameters to a target is called the parameter list length. The parameter list length specifies the number of bytes sent during the DATA OUT phase.

The transfer length of the commands that are used to return sense data (e.g., REQUEST SENSE, INQUIRY, MODE SENSE, etc) to an initiator is called the allocation length. The allocation length specifies the number of bytes that the initiator has allocated for returned data. The target shall terminate the DATA IN phase when allocation length bytes have been transferred or when all available sense data have been transferred to the initiator, whichever is less.

6.2.6 Control Byte. The control byte is the last byte of every command descriptor block. A typical control byte is described in Table 6-5.

Table 6-5
Control Byte

Bit / Byte	7	6	5	4	3	2	1	0
Last	Vendor unique		Reserved				Flag	Link

Bit	Description
7 – 6	Vendor unique
5 – 2	Reserved
1	Flag bit - If the link bit is zero, then the flag bit shall be set to zero. If the link bit is one, and if the command terminates successfully, the target shall send LINKED COMMAND COMPLETE message if the flag bit is zero and shall send LINKED COMMAND COMPLETE (WITH FLAG) message if the flag bit is one. Typically, this bit is used to cause an interrupt in the initiator between linked commands.
0	Link bit - This bit is set to one to indicate that the initiator desires an automatic link to the next command upon successful completion of the current command. Implementation of linked commands is optional. If the link bit is one, targets that implement linked commands, upon successful termination of the command, shall return INTERMEDIATE status and shall then send one of the two messages defined by the flag bit (above). Targets that do not implement linked commands shall return a CHECK CONDITION status and, if extended sense is implemented, shall set the sense key to ILLEGAL REQUEST if either of the link and flag bits are set to one.

6.3 Command Examples

6.3.1 Single Command Example. A typical operation on the SCSI bus is likely to include a single READ command to a peripheral device. This operation is described in detail starting with a request from the initiator. This example assumes that no linked commands and no malfunctions or errors occur.

The initiator has active pointers and a set of stored pointers representing active disconnected SCSI devices (an initiator without disconnect capability does not require stored pointers). The initiator sets up the active pointers for the operation requested, arbitrates for the SCSI bus, and selects the target. Once this process is completed, the target assumes control of the operation.

The target obtains the command from the initiator (in this case, a READ command). The target interprets the command and executes it. In this case, the target gets the data from the peripheral device and sends it to the initiator. At the completion of the READ command, the target sends a status

byte to the initiator. To end the operation, the target sends a COMMAND COMPLETE message to the initiator.

6.3.2 Disconnect Example. In the above single command example, the length of time necessary to obtain the data may require a time-consuming physical seek. In order to improve system throughput, the target may disconnect from the initiator, freeing the SCSI bus to allow other requests to be sent to other logical units. To do this, the initiator needs to be reselectable and capable of restoring the pointers upon reconnection. The target needs to be capable of arbitrating for the SCSI bus and reselecting the initiator.

After the target has received the READ command (and has determined that there will be a delay), it disconnects by sending a DISCONNECT message and releasing BSY.

When the data are ready to be transferred, the target reconnects to the initiator. As a result of this reconnection, the initiator restores the pointers to their most recent saved values (which, in this case, are the initial values) and the target continues (as in the single command example) to finish the operation. The initiator recognizes that the operation is complete when COMMAND COMPLETE message is received.

If target wishes to disconnect after transferring part of the data (e.g., while crossing a cylinder boundary), it may do so by sending a SAVE DATA POINTER message and a DISCONNECT message to the initiator and then disconnecting. When reconnection is completed, the current data pointer value is restored to its value immediately prior to the SAVE DATA POINTER message.

On those occasions when an error or exception condition occurs and the target elects to repeat the information transfer, the target may repeat the transfer by either issuing a RESTORE POINTERS message or by disconnecting without issuing a SAVE DATA POINTER message. When reconnection is completed, the most recent saved pointer values are restored.

6.3.3 Linked Command Example. The link function defines a relationship between commands that when combined with the relative address bit, allows previous operations to modify subsequent operations. Link makes high-performance functions possible by providing a relative addressing capability and allowing multiple command execution without invoking the functional component of the initiator.

If the desired data address (in the previously described READ command example) is unknown, but a search key defined as some particular bytes of a field is known, then by linking the READ command to a SEARCH DATA EQUAL command, the data can be quickly and effectively transferred to the initiator.

A LINKED COMMAND COMPLETE message is sent from the target to the initiator to indicate linked command completion. The initiator then updates the stored pointers so that subsequent requests from the target will reference the next command of the chain. Command processing of linked and single commands is similar except that relative addressing is permitted in linked commands. Linked commands shall be addressed to a single logical unit.

For example, the successful completion of a SEARCH DATA EQUAL command causes

the target to fetch the linked READ command from the initiator. If the relative address bit in the READ command has been set to one, and the address field of the READ command is set to zero, the target transfers the successfully searched block to the initiator.

7. Command Descriptions for All Device Types

7.1 Group 0 Commands for All Device Types. These commands shall be as listed in Table 7-1.

Table 7-1
Group 0 Common Commands for All Device Types

Operation Code	Type	Command Name	Section	Page
00h	O	TEST UNIT READY	7.1.1	62
01h	*			
02h	V			
03h	M	REQUEST SENSE	7.1.2	63
04h	*			
05h	*			
06h	V			
07h	*			
08h	*			
09h	V			
0Ah	*			
0Bh	*			
0Ch	V			
0Dh	V			
0Eh	V			
0Fh	*			
10h	*			
11h	*			
12h	E	INQUIRY	7.1.3	69
13h	*			
14h	*			
15h	*			
16h	*			
17h	*			
18h	O	COPY	7.1.4	71
19h	*			
1Ah	*			
1Bh	*			
1Ch	O	RECEIVE DIAGNOSTIC RESULTS	7.1.5	80
1Dh	O	SEND DIAGNOSTIC	7.1.6	81
1Eh	*			
1Fh	R			

Key: M = Command implementation is mandatory.
E = Command implementation is required for SCSI devices that support device-independent self-configuring software.
O = Command implementation is optional.
R = Operation code is reserved for future standardization.
V = Operation code is available for vendor unique commands.
* = These operation codes may have different meanings for specific types of peripheral devices. (See the appropriate section for further information.)

7.1.1 TEST UNIT READY Command

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

Table 7-2
TEST UNIT READY Command

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 (Table 7-2) 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.

7.1.2 REQUEST SENSE Command

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

Table 7-3
REQUEST SENSE Command

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 (Table 7-3) requests that the target transfer sense data to the initiator.

The sense data shall be valid for a CHECK CONDITION status returned on the prior command. This sense data shall be preserved by the target for the initiator until retrieved by 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. Sense data shall be cleared 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 (see 5.1.3.4), the target shall assume that the REQUEST SENSE command is from the same initiator.

The allocation length specifies the number of bytes that the initiator has allocated for returned sense data. An allocation length of zero indicates that four bytes of sense data shall be transferred. 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 sense data have been transferred to the initiator, whichever is less.

The REQUEST SENSE command shall return the CHECK CONDITION status only to report fatal errors for the REQUEST SENSE command. For example:

(1) The target receives a nonzero reserved bit in the command descriptor block.
(2) An unrecovered parity error occurs on the DATA BUS.
(3) A target malfunction prevents return of the sense data.

If any nonfatal error occurs during the execution of the REQUEST SENSE command, the target shall return the sense data with GOOD status.

Following a fatal error on a REQUEST SENSE command, sense data may be invalid.

A target may implement the nonextended, the extended, or both sense data formats. (Implementors note: Targets that implement both sense data formats may select the nonextended sense data format in response to an allocation length of zero. Other methods of selection are also feasible.)

7.1.2.1 Nonextended Sense. The format of the sense data is determined by the error class. Error classes 0 through 6 use the nonextended sense data format (Table 7-4). Error class 7 is described in 7.1.2.2.

Table 7-4
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.

7.1.2.2 Extended Sense. Error class 7 specifies extended sense. Error code zero specifies the extended sense data format. Error code Fh specifies a vendor unique data format for extended sense. Error codes 1h through Eh are reserved.

The extended sense data format is shown in Table 7-5.

Table 7-5
Extended Sense Data Format

Bit / Byte	7	6	5	4	3	2	1	0
0	Valid	Error Class (7)			Error Code (0)			
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 minus 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 minus 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 condition (end-of-tape, beginning-of-tape, out-of-paper, etc) exists 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 (see Table 7-6).

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 sense keys are described in Tables 7-6 and 7-7.

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.

Table 7-6
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.
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. If the target detects an invalid parameter in the additional parameters supplied as data, then the target may have already altered the medium.
6h	UNIT ATTENTION. Indicates that the removable medium may have been changed or the target has been reset. See 6.1.3 for more detailed information about the unit attention condition.
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.

Table 7-7
Sense Key (8h-Fh) Descriptions

Sense Key	Description
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. This sense key is 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.

7.1.3 INQUIRY Command

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

Table 7-8
INQUIRY Command

Byte \ Bit	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 (Table 7-8) 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 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 INQUIRY data have been transferred to the initiator, whichever is less.

The INQUIRY command shall return a CHECK CONDITION status only when the target cannot return the requested INQUIRY data. (Implementors note: It is recommended that the INQUIRY data be returned even though the peripheral device may not be ready for other commands.)

If an INQUIRY command is received from an initiator with a pending unit attention condition (before the target reports CHECK CONDITION status), the target shall perform the INQUIRY command and shall not clear the unit attention condition. (See 6.1.3.)

The INQUIRY data (Table 7-9) contains a five-byte header, followed by the vendor unique parameters, if any.

Table 7-9
INQUIRY Data

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

The peripheral device type is shown in Table 7-10.

Table 7-10
Peripheral Device Type

Code	Description
00h	Direct-access device (e.g., magnetic disk)
01h	Sequential-access device (e.g., magnetic tape)
02h	Printer device
03h	Processor device
04h	Write-once read-multiple device (e.g., some optical disks)
05h	Read-only direct-access device (e.g., some optical disks)
06h – 7Eh	Reserved
7Fh	Logical unit not present
80h – FFh	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 shall 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 usage of nonzero code values in the ISO version and ECMA version fields are defined by the International Organization for Standardization and the European Computer Manufacturers Association, respectively. A zero code value in these fields shall indicate that the target does not claim compliance to the ISO version of SCSI (ISO DP 9316) or the ECMA version of SCSI (ECMA-111). Note that it is possible to claim compliance to more than one of these SCSI standards.

The ANSI-approved version indicates the implemented version of this standard and is defined as shown in Table 7-11.

Table 7-11
ANSI-Approved Version

Code	Description
0h	Version is unspecified.
1h	This version. This code shall be used by SCSI devices that claim to comply with this standard.
2h – 7h	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.

7.1.4 COPY Command

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

Table 7-12
COPY Command

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 (Table 7-12) 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 that shall be sent during the DATA OUT phase of the command. A parameter list length of zero indicates that no data shall be transferred. This condition shall not be considered as an error.

The COPY parameter list (Table 7-13) begins with a four-byte header that contains the COPY function code and priority. Following the header is one or more segment descriptors.

Table 7-13
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 COPY function codes are defined in Table 7-14.

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. (See Table 7-14.) The segment descriptor formats are described in Tables 7-15 through 7-17. A maximum of 256 segment descriptors are permitted. The segment descriptors are identified by ascending numbers beginning with zero.

Table 7-14
COPY Functions

Peripheral Device Type		COPY Function Code	Segment Descriptor Table	Comment
Source	Destination			
00h	01h	00h	Table 7-15	
00h	02h	00h	Table 7-15	
00h	03h	00h	Table 7-15	
04h	01h	00h	Table 7-15	Direct Access
04h	02h	00h	Table 7-15	to
04h	03h	00h	Table 7-15	Sequential Access
05h	01h	00h	Table 7-15	
05h	02h	00h	Table 7-15	
05h	03h	00h	Table 7-15	
01h	00h	01h	Table 7-15	Sequential Access
01h	04h	01h	Table 7-15	to
03h	00h	01h	Table 7-15	Direct Access
03h	04h	01h	Table 7-15	
00h	00h	02h	Table 7-16	
00h	04h	02h	Table 7-16	Direct Access
04h	00h	02h	Table 7-16	to
04h	04h	02h	Table 7-16	Direct Access
05h	00h	02h	Table 7-16	
05h	04h	02h	Table 7-16	
01h	01h	03h	Table 7-17	
01h	02h	03h	Table 7-17	Sequential Access
01h	03h	03h	Table 7-17	to
03h	01h	03h	Table 7-17	Sequential Access
03h	02h	03h	Table 7-17	
03h	03h	03h	Table 7-17	

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

7.1.4.1 **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.

7.1.4.2 **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 shall specify 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 shall contain the status byte from the source logical unit. The subsequent bytes shall 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.

7.1.4.3 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 Table 7-15. 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.

Table 7-15
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 shall be terminated with a CHECK CONDITION status and the sense key shall be 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.

7.1.4.4 COPY Function Code 02h. The format for the segment descriptors for COPY transfers among direct-access devices is specified by Table 7-16. This format is required for COPY function code 02h. The segment descriptor may be repeated up to 256 times within the parameter list length specified in the command descriptor block.

Table 7-16
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.

7.1.4.5 COPY Function Code 03h. The format for the segment descriptors for COPY transfers among sequential-access devices is specified by Table 7-17. This format is required for COPY function code 03h. The segment descriptor may be repeated up to 256 times within the parameter list length specified in the command descriptor block.

Table 7-17
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 shall be terminated with a CHECK CONDITION status and the sense key shall be 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.

7.1.5 RECEIVE DIAGNOSTIC RESULTS Command

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

Table 7-18
RECEIVE DIAGNOSTIC RESULTS Command

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 (Table 7-18) requests analysis data be sent to the initiator after completion of a SEND DIAGNOSTIC command (see 7.1.6).

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.

7.1.6 SEND DIAGNOSTIC Command

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

Table 7-19
SEND DIAGNOSTIC Command

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 (Table 7-19) 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 (7.1.5).

7.2 Group 1 Commands for All Device Types. These commands shall be as listed in Table 7-20.

Table 7-20
Group 1 Commands for All Device Types

Operation Code	Type	Command Name	Section	Page
20h	V			
21h	V			
22h	V			
23h	V			
24h	V			
25h	*			
26h	V			
27h	V			
28h	*			
29h	V			
2Ah	*			
2Bh	*			
2Ch	V			
2Dh	V			
2Eh	*			
2Fh	*			
30h	*			
31h	*			
32h	*			
33h	*			
34h	R			
35h	R			
36h	R			
37h	R			
38h	R			
39h	O	COMPARE	7.2.1	83
3Ah	O	COPY AND VERIFY	7.2.2	84
3Bh	R			
3Ch	R			
3Dh	R			
3Eh	R			
3Fh	R			

Key: O = Command implementation is optional.
R = Operation code is reserved for future standardization.
V = Operation code is available for vendor unique commands.
* = These operation codes may have different meanings for specific types of peripheral devices. (See the appropriate section for further information.)

7.2.1 COMPARE Command

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

Table 7-21
COMPARE Command

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 (Table 7-21) 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. (See 7.1.4 for additional information about the COPY command.)

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.

7.2.2 COPY AND VERIFY Command

Peripheral Device Type: All
Operation Code Type: Optional
Operation Code: 3Ah

Table 7-22
COPY AND VERIFY Command

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 (Table 7-22) 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 7.1.4 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.

7.3 Group 2 Commands for All Device Types. The Group 2 commands (operation codes 40h through 5Fh) are all reserved for future standardization.

7.4 Group 3 Commands for All Device Types. The Group 3 commands (operation codes 60h through 7Fh) are all reserved for future standardization.

7.5 Group 4 Commands for All Device Types. The Group 4 commands (operation codes 80h through 9Fh) are all reserved for future standardization.

7.6 Group 5 Commands for All Device Types. The Group 5 commands with operation codes A0h through AFh are vendor unique. The Group 5 commands with operation codes B0h through BFh are reserved for future standardization.

7.7 Group 6 Commands for All Device Types. The Group 6 commands (operation codes C0h through DFh) are all vendor unique.

7.8 Group 7 Commands for All Device Types. The Group 7 commands (operation codes E0h through FFh) are all vendor unique.

8. Command Descriptions for Direct-Access Devices

8.1 Group 0 Commands for Direct-Access Devices. The Group 0 commands for direct-access devices shall be as shown in Table 8-1.

Table 8-1
Group 0 Commands for Direct-Access Devices

Operation Code	Type	Command Name	Section	Page
00h	O	TEST UNIT READY	7.1.1	62
01h	O	REZERO UNIT	8.1.1	87
02h	V			
03h	M	REQUEST SENSE	7.1.2	63
04h	M	FORMAT UNIT	8.1.2	87
05h	V			
06h	V			
07h	O	REASSIGN BLOCKS	8.1.3	93
08h	M	READ	8.1.4	95
09h	V			
0Ah	M	WRITE	8.1.5	96
0Bh	O	SEEK	8.1.6	97
0Ch	V			
0Dh	V			
0Eh	V			
0Fh	V			
10h	V			
11h	V			
12h	E	INQUIRY	7.1.3	69
13h	V			
14h	V			
15h	O	MODE SELECT	8.1.7	98
16h	O	RESERVE	8.1.8	102
17h	O	RELEASE	8.1.9	107
18h	O	COPY	7.1.4	71
19h	V			
1Ah	O	MODE SENSE	8.1.10	108
1Bh	O	START/STOP UNIT	8.1.11	111
1Ch	O	RECEIVE DIAGNOSTIC RESULTS	7.1.5	80
1Dh	O	SEND DIAGNOSTIC	7.1.6	81
1Eh	O	PREVENT/ALLOW MEDIUM REMOVAL	8.1.12	112
1Fh	R			

Key: M = Command implementation is mandatory.
E = Command implementation is required for SCSI devices that support device-independent self-configuring software.
O = Command implementation is optional.
R = Operation code is reserved for future standardization.
V = Operation code is available for vendor unique commands.

8.1.1 REZERO UNIT Command

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

Table 8-2
REZERO UNIT Command

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 (Table 8-2) requests that the target set the logical unit to a specific state. See vendor specifications for details.

8.1.2 FORMAT UNIT Command

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

Table 8-3
FORMAT UNIT Command

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 (Table 8-3) 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.

The FORMAT UNIT command is both a mandatory command and an extended command with multiple options (see Table 8-4).

The FORMAT UNIT command shall be rejected with RESERVATION CONFLICT status if any extent (see 8.1.8.2) in the specified logical unit is reserved.

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. (See Table 8-4 for further information.)

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.

Table 8-4
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. (See Table 8-5.)
1	1	0	X	X	Optional	Format using defects in the defect list as the full set of known defects. (See Table 8-5.)
1	0	1	0	0	Optional	Format adding the defects in the defect list to the known defects. (See Table 8-6.)
1	1	1	0	0	Optional	Format using the defects in the defect list as the full set of known defects. (See Table 8-6.)
1	0	1	0	1	Optional	Format adding the defects in the defect list to the known defects. (See Table 8-7.)
1	1	1	0	1	Optional	Format using the defects in the defect list as the full set of known defects. (See Table 8-7.)
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 Tables 8-5, 8-6, and 8-7 contain a four-byte header followed by one or more defect descriptors. The length of the defect descriptors vary 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 Table 8-5, the defect list length is equal to four times the number of defect descriptors. In Tables 8-6 and 8-7, the defect list length is equal to eight times the number of defect descriptors.

Table 8-5
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.

Table 8-6
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 comprised 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 FFFFFFFFh indicates that the entire track shall be reassigned.

Table 8-7
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 comprised 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.

8.1.3 REASSIGN BLOCKS Command

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

Table 8-8
REASSIGN BLOCKS Command

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 (Table 8-8) 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. The target shall reassign 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 assign to multiple physical addresses (until no more spare locations remain on the medium).

The REASSIGN BLOCKS defect list (Table 8-9) 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.

Table 8-9
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 shall be in ascending order.

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

8.1.4 READ Command

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

Table 8-10
READ Command

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 (Table 8-10) 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 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.

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 (see 8.1.8) 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 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)
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.

8.1.5 WRITE Command

Peripheral Device Type: Direct Access
Operation Code Type: Mandatory
Operation Code: 0Ah

Table 8-11
WRITE Command

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 (Table 8-11) 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 to 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 RESERVATION CONFLICT status if any reservation access conflict (see 8.1.8) 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)
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.

8.1.6 SEEK Command

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

Table 8-12
SEEK Command

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 (Table 8-12) requests that the logical unit seek to the specified logical block address.

8.1.7 MODE SELECT Command

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

Table 8-13
MODE SELECT Command

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 (Table 8-13) 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 (Table 8-14) contains a four-byte header, followed by zero or more block descriptors, followed by the vendor unique parameters, if any.

Table 8-14
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 are defined in Table 8-15.

Table 8-15
Direct-Access Medium-Type Codes

Code Value	Medium Type
00h	Default medium type (currently mounted medium type)
01h	Flexible disk, single-sided; unspecified medium
02h	Flexible disk, double-sided; unspecified medium

Flexible Disks

	Diameter mm (inch)	Bit Density Bits/Radian	Track Density /mm (/inch)	Number Of Sides	Reference Standard
05h	200 (8.0)	6 631	1.9 (48)	1	ANSI X3.73-1980
06h	200 (8.0)	6 631	1.9 (48)	2	Note 1
09h	200 (8.0)	13 262	1.9 (48)	1	None
0Ah	200 (8.0)	13 262	1.9 (48)	2	ANSI X3.121-1984
0Dh	130 (5.25)	3 979	1.9 (48)	1	ANSI X3.82-1980
12h	130 (5.25)	7 958	1.9 (48)	2	ANSI X3.125-1985
16h	130 (5.25)	7 958	3.8 (96)	2	ANSI X3.126-1986
1Ah	130 (5.25)	13 262	3.8 (96)	2	ISO DIS8630-1985
1Eh	90 (3.5)	7 958	5.3 (135)	2	ANSI X3.137 (Note 2)

Direct-Access Magnetic Tapes

	Width mm (inch)	Tracks	Density ftpmm (ftpi)	Reference Standard
40h	6.3 (0.25)	12	394 (10 000)	Note 1
44h	6.3 (0.25)	24	394 (10 000)	Note 1

80h – FFh	Vendor unique
All others	Reserved

NOTES:
(1) See Appendix F for additional standards that may be applicable.
(2) This standard is under development. Please contact the Secretariat for current information on status and availability.

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 (default density of medium)
01h	Flexible disk, single-density
02h	Flexible disk, double-density
03h — 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.

8.1.8 RESERVE Command

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

Table 8-16
RESERVE Command

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 (Table 8-16) 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 third-party reservation option is implemented, the logical units or extents may be reserved for another specified SCSI device. The RESERVE and RELEASE commands provide the basic mechanism for contention resolution in multiple-initiator systems.

8.1.8.1 Logical Unit Reservation. If the extent bit is zero, this command shall request that the entire logical unit be reserved for the exclusive use of the initiator until the reservation is superseded by another valid RESERVE command from the initiator that made the reservation or until released by a RELEASE command from the same initiator, by a BUS DEVICE RESET message from any initiator, or by a "hard" RESET condition. A logical unit reservation shall not be granted if the logical unit or any extent is reserved by another initiator. It shall be permissible for an initiator to reserve a logical unit that is currently reserved by that initiator. If the extent bit is zero, the reservation identification and the extent list length shall be ignored.

If the logical unit, or any extent within the logical unit is reserved for another initiator, the target shall respond by either:

(1) Returning a RESERVATION CONFLICT status

(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 command, which may be queued, or a RELEASE command, which shall be ignored, then the command shall be rejected with RESERVATION CONFLICT status.

8.1.8.2 Extent Reservation (Optional). The reservation identification provides a means for an initiator to identify each extent reservation. This allows an initiator in a multiple tasking environment, to have multiple reservations outstanding. The reservation identification is used in the RELEASE command to specify which reservation is to be released. It is also used in superseding RESERVE commands to specify which reservation is to be superseded.

If the extent reservation option is implemented, then the extent release option (see 8.1.9.2) shall also be implemented. These options permit multiple extents within the logical unit to be reserved, each with a separate reservation type.

If the extent bit is one, and the extent reservation option is implemented, then the target shall process the reservation request as follows:

(1) The extent list shall be checked for the number of extents in the reservation request. If the extent list length is zero, no reservations shall be changed and this condition shall not be treated as an error. If the extent list contains more extents than are supported on the logical unit, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST. If the extent list contains more extents than are currently available on the logical unit, then the target shall either:

(a) Return RESERVATION CONFLICT status

(b) Queue the request and disconnect until sufficient extents are made available. When sufficient extents are available, the target shall reconnect to continue processing the request.

(2) The extent list shall be checked for valid extent logical block addresses. If any logical block address is invalid for this logical unit, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST. The extent list shall be checked for invalid extent overlaps (as defined by reservation type) with other extent descriptors in the extent list and if invalid overlaps are found, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

(3) If the requested reservation does not conflict with any active or previously requested reservation, the extents specified shall be reserved until superseded by another valid RESERVE command from the initiator that made the reservation or until released by a RELEASE command from the same initiator, by a BUS DEVICE RESET message from any initiator, or by a "hard" RESET condition. If either of the last two conditions occur, the next command from each initiator shall be terminated with a CHECK CONDITION status and the sense key shall be set to UNIT ATTENTION.

(4) If the reservation request conflicts with a reservation already active or a reservation request that is already queued, then the target shall either:

(a) Return RESERVATION CONFLICT status

(b) Queue the reservation and disconnect until it is allowed to be active. Reservations are only made active when all extents are free from conflict with active or previously queued reservations.

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.

Table 8-17
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 zero or more descriptors as shown in Table 8-17. 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. Four types of reservations are possible as follows:

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.

8.1.8.3 Third Party Reservation (Optional). The third-party reservation option for the RESERVE command 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. Any target that implements the third-party reservation option shall also implement the third-party release option (see 8.1.9.3).

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 command shall reserve the specified logical unit or extents for the SCSI device specified in the third-

party device ID field. The target shall preserve the reservation until it is superseded by another valid RESERVE command from the initiator that made the reservation or until it is released by the same initiator, 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 command with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

8.1.8.4 **Superseding Reservations.** An initiator that holds a current reservation may modify that reservation by issuing another RESERVE command to the same logical unit and, if the extent bit is one, using the same reservation identification. The superseding RESERVE command shall release the previous reservation state when the new reservation request is granted. The previous reservation shall not be modified if the new reservation request cannot be granted. If the superseding reservation cannot be granted because of conflicts with a previous active reservation (other than the reservation being superseded), then the target shall either:

(1) Return RESERVATION CONFLICT status

(2) Queue the reservation request and disconnect until it is allowed to be active. The reservation request shall be made active when it is free from conflict with all active reservations. A superseding reservation takes priority over any previously queued reservation request.

IMPLEMENTORS NOTE: Superseding reservations are principally intended to allow the SCSI device ID to be changed on a reservation using the third-party reservation option. This capability is necessary for certain situations when using the COPY command.

8.1.9 RELEASE Command

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

Table 8-18
RELEASE Command

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 (Table 8-18) 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. In this case, the target returns GOOD status without altering any other reservation.

8.1.9.1 Logical Unit Release. If the extent bit is zero, this command shall cause the target to terminate all logical unit and extent reservations that are active from the initiator to the specified logical unit.

8.1.9.2 Extent Release (Optional). If the extent bit is one and the extent release option is not implemented, then the RELEASE command shall be terminated with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST. This option shall be implemented if the extent reservation option (see 8.1.8.2) is implemented.

If the extent bit is one and the extent release option is implemented, this command shall cause any reservation from the requesting initiator with a matching reservation identification 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 to terminate the 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.

8.1.9.3 Third Party Release (Optional). 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 (see 8.1.8.3). 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 or extents, but only if the reservation was made using the third-party reservation option by the initiator that is requesting the release 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.

8.1.10 MODE SENSE Command

Peripheral Device Type: Direct Access
Operation Code Type: Optional
Operation Code: 1Ah

Table 8-19
MODE SENSE Command

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 (Table 8-19) 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 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 (Table 8-20) contains a four-byte header, followed by zero or more eight-byte block descriptors, followed by the vendor unique parameters, if any.

Table 8-20
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 are defined in Table 8-15.

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.

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

00h	Default (only one density supported)
01h	Flexible disk, single density
02h	Flexible disk, double density
03h — 7Fh	Reserved
80h — FFh	Vendor unique

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.

8.1.11 START/STOP UNIT Command

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

Table 8-21
START/STOP UNIT Command

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 (Table 8-21) 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.

8.1.12 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: 1Eh

Table 8-22
PREVENT/ALLOW MEDIUM REMOVAL Command

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 (Table 8-22) 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.

8.2 Group 1 Commands for Direct-Access Devices. The Group 1 commands for direct-access devices shall be as shown in Table 8-23.

Table 8-23
Group 1 Commands for Direct-Access Devices

Operation Code	Type	Command Name	Section	Page
20h	V			
21h	V			
22h	V			
23h	V			
24h	V			
25h	E	READ CAPACITY	8.2.1	114
26h	V			
27h	V			
28h	E	READ	8.2.2	116
29h	V			
2Ah	E	WRITE	8.2.3	118
2Bh	O	SEEK	8.2.4	119
2Ch	V			
2Dh	V			
2Eh	O	WRITE AND VERIFY	8.2.5	120
2Fh	O	VERIFY	8.2.6	121
30h	O	SEARCH DATA HIGH	8.2.7.1	125
31h	O	SEARCH DATA EQUAL	8.2.7.2	125
32h	O	SEARCH DATA LOW	8.2.7.3	125
33h	O	SET LIMITS	8.2.8	126
34h	R			
35h	R			
36h	R			
37h	R			
38h	R			
39h	O	COMPARE	7.2.1	83
3Ah	O	COPY AND VERIFY	7.2.2	84
3Bh	R			
3Ch	R			
3Dh	R			
3Eh	R			
3Fh	R			

Key: M = Command implementation is mandatory.
E = Command implementation is required for SCSI devices that support device-independent self-configuring software.
O = Command implementation is optional.
R = Operation code is reserved for future standardization.
V = Operation code is available for vendor unique commands.

8.2.1 READ CAPACITY Command

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

Table 8-24
READ CAPACITY Command

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 (Table 8-24) 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 after which a substantial delay in data transfer will be encountered. This logical block address shall be greater than or equal to the logical block address specified in the command descriptor block. (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 8-25 shall be sent during the DATA IN phase of the command.

Table 8-25
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)

8.2.2 READ Command

Peripheral Device Type: Direct Access
Operation Code Type: Extended
Operation Code: 28h

Table 8-26
READ Command

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 (Table 8-26) 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 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. 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 (see 8.1.8) exists and no data shall be read.

If any of the following conditions occur, this command shall return 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.

8.2.3 WRITE Command

Peripheral Device Type: Direct Access
Operation Code Type: Extended
Operation Code: 2Ah

Table 8-27
WRITE Command

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 (Table 8-27) 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 (see 8.1.8) 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.

8.2.4 SEEK Command

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

Table 8-28
SEEK Command

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

The SEEK command (Table 8-28) requests that the logical unit seek to the specified logical block address.

8.2.5 WRITE AND VERIFY Command

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

Table 8-29
WRITE AND VERIFY Command

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 (Table 8-29) 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.

8.2.6 VERIFY Command

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

Table 8-30
VERIFY Command

Byte \ Bit	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 (Table 8-30) 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.

8.2.7 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

Table 8-31
SEARCH DATA Commands

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 (Table 8-31) 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.

The invert bit determines whether the search condition is to be inverted. See 8.2.7.1 through 8.2.7.3 for a description of the search conditions for the individual SEARCH DATA commands.

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. If the search is not satisfied and no error occurs, the command shall be terminated with GOOD status.

A link bit of one indicates a command is linked to the SEARCH DATA command and if the search is satisfied, CONDITION MET status is returned and 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 (Table 8-32) contains a fourteen-byte header, followed by one or more search argument descriptors.

Table 8-32
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 logical record length field specifies the record length in bytes.

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) have 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.

8.2.7.1 **SEARCH DATA HIGH Command.** The SEARCH DATA HIGH command (Table 8-31, 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 8.2.7.)

8.2.7.2 **SEARCH DATA EQUAL Command.** The SEARCH DATA EQUAL command (Table 8-31, 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 8.2.7.)

8.2.7.3 **SEARCH DATA LOW Command.** The SEARCH DATA LOW command (Table 8-31, 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 8.2.7.)

8.2.8 SET LIMITS Command

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

Table 8-33
SET LIMITS Command

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 (Table 8-33) defines the range within which subsequent linked commands may 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 within the range are inhibited. A write inhibit (WrInh) bit of one indicates that write operations within the range are inhibited.

The logical block address specifies the starting address for the range. The number of blocks specifies the number of logical blocks within the range. 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, if extended sense is implemented, 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, if extended sense is implemented, the sense key shall be set to DATA PROTECT.

9. Group 0 Command Descriptions for Sequential-Access Devices

The Group 0 commands for sequential-access devices shall be as shown in Table 9-1.

Table 9-1
Group 0 Commands for Sequential-Access Devices

Operation Code	Type	Command Name	Section	Page
00h	O	TEST UNIT READY	7.1.1	62
01h	M	REWIND	9.1	128
02h	V			
03h	M	REQUEST SENSE	7.1.2	63
04h	R			
05h	E	READ BLOCK LIMITS	9.2	129
06h	V			
07h	V			
08h	M	READ	9.3	130
09h	V			
0Ah	M	WRITE	9.4	132
0Bh	O	TRACK SELECT	9.5	133
0Ch	V			
0Dh	V			
0Eh	V			
0Fh	O	READ REVERSE	9.6	134
10h	M	WRITE FILEMARKS	9.7	135
11h	O	SPACE	9.8	136
12h	E	INQUIRY	7.1.3	69
13h	O	VERIFY	9.9	138
14h	O	RECOVER BUFFERED DATA	9.10	139
15h	O	MODE SELECT	9.11	140
16h	O	RESERVE UNIT	9.12.1	143
17h	O	RELEASE UNIT	9.12.2	144
18h	O	COPY	7.1.4	71
19h	O	ERASE	9.13	145
1Ah	O	MODE SENSE	9.14	146
1Bh	O	LOAD/UNLOAD	9.15	149
1Ch	O	RECEIVE DIAGNOSTIC RESULTS	7.1.5	80
1Dh	O	SEND DIAGNOSTIC	7.1.6	81
1Eh	O	PREVENT/ALLOW MEDIUM REMOVAL	9.16	150
1Fh	R			

Key: M = Command implementation is mandatory.
E = Command implementation is required for SCSI devices that support device-independent self-configuring software.
O = Command implementation is optional.
R = Operation code is reserved for future standardization.
V = Operation code is available for vendor unique commands.

9.1 REWIND Command

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

Table 9-2
REWIND Command

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 (Table 9-2) 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.

9.2 READ BLOCK LIMITS Command

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

Table 9-3
READ BLOCK LIMITS Command

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 (Table 9-3) requests that the target's capability for block length limits be returned for the logical unit. The READ BLOCK LIMITS data shown in Table 9-4 shall be sent during the DATA IN phase of the command.

Table 9-4
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, only fixed-length blocks of the length indicated are supported. Otherwise, variable-length blocks are supported. For variable-length blocks, if the maximum block length equals zero, no upper limit is specified.

9.3 READ Command

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

Table 9-5
READ Command

Byte \ Bit	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 (Table 9-5) 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 are 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 shall transfer the current block length times the transfer length bytes of data to the initiator. Upon termination of the READ command, the medium shall be 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 shall send a CHECK CONDITION status to the initiator and shall set 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 shall set the valid bit to one and 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 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 shall set the end-of-medium (EOM) bit to one in extended sense. The sense key shall be set to MEDIUM ERROR. If the fixed bit is one, the target shall set 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 shall be transferred and the current position on the logical unit shall not be changed. This condition shall not be considered as an error.

9.4 WRITE Command

Peripheral Device Type: Sequential Access
Operation Code Type: Mandatory
Operation Code: 0Ah

Table 9-6
WRITE Command

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 (Table 9-6) 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 shall be transferred from the initiator and shall be 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 (returned by the READ BLOCK LIMITS command, 9.2). 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 (see 9.3). Upon termination, 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 shall be transferred and the current position on the logical unit shall not be changed. This condition shall not be considered as an error.

9.5 TRACK SELECT Command

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

Table 9-7
TRACK SELECT Command

Byte \ Bit	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 (Table 9-7) requests that the track specified in the track value field be selected.

9.6 READ REVERSE Command

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

Table 9-8
READ REVERSE Command

Byte \ Bit	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 (Table 9-8) 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 shall terminate with a CHECK CONDITION status and the EOM bit in extended sense shall be set to one if beginning-of-medium or load-point is encountered. The sense key shall be set to NO SENSE. If the fixed bit is one, then the valid bit shall be 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 shall not be 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.

9.7 WRITE FILEMARKS Command

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

Table 9-9
WRITE FILEMARKS Command

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 (Table 9-9) 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 (see buffered mode in the MODE SENSE command, 9.14) 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.

9.8 SPACE Command

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

Table 9-10
SPACE Command

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 (Table 9-10) 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.

9.9 VERIFY Command

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

Table 9-11
VERIFY Command

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 (Table 9-11) 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 shall be verified and the current position on the logical unit shall not be changed. This condition shall not be considered as an error.

9.10 RECOVER BUFFERED DATA Command

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

Table 9-12
RECOVER BUFFERED DATA Command

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 (Table 9-12) 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 similarly to 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 would have been transferred to the medium. One or more RECOVER BUFFERED DATA commands may be used to read the unwritten buffered data.

Refer to the READ command (9.3) for a definition of the fixed bit and the transfer length field.

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.

9.11 MODE SELECT Command

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

Table 9-13
MODE SELECT Command

Byte \ Bit	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 (Table 9-13) 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 shall not be considered as an error.

The MODE SELECT parameter list shown in Table 9-14 contains a four-byte header, followed by zero or more eight-byte block descriptors, followed by the vendor unique parameters, if any.

Table 9-14
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 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 are defined in Table 9-15.

Table 9-15
Sequential-Access Density Codes

Code Value	Density							
00h	Default (peripheral device's default or only density)							
	Magnetic Tapes							
	Width mm (inch)	Tracks	Density bpmm	Density (bpi)	Code	Type	ANSI Reference Standard	Note
01h	12.7 (0.5)	9	32	(800)	NRZI	R	X3.22-1983	2
02h	12.7 (0.5)	9	63	(1 600)	PE	R	X3.39-1986	2
03h	12.7 (0.5)	9	246	(6 250)	GCR	R	X3.54-1986	2
04h	6.3 (0.25)	4/9	315	(8 000)	GCR	C		1,3
05h	6.3 (0.25)	4/9	315	(8 000)	GCR	C	X3.136-1986	1
06h	12.7 (0.5)	9	126	(3 200)	PE	R		2,4
07h	6.3 (0.25)	4	252	(6 400)	IMFM	C	X3.116-1986	1
08h	3.81 (0.15)	4	315	(8 000)	GCR	CS		1,4
09h	12.7 (0.5)	18	1491	(37 871)	GCR	C		2,4
0Ah	12.7 (0.5)	22	262	(6 667)	MFM	C		1,4
0Bh	6.3 (0.25)	4	63	(1 600)	PE	C	X3.56-1986	1
0Ch	12.7 (0.5)	24	500	(12 690)	GCR	C		1
0Dh	12.7 (0.5)	24	999	(25 380)	GCR	C		1
80h – FFh	Vendor unique							
All others	Reserved							

Key:

Code		Type	
NRZI	Non Return to Zero, change on ones	R	Reel-to-Reel
GCR	Group Code Recording	C	Cartridge
PE	Phase Encoded	CS	Cassette
IMFM	Inverted Modified Frequency Modulation		
MFM	Modified Frequency Modulation		

NOTES:
(1) Serially Recorded.
(2) Parallel Recorded.
(3) Old format known as QIC-11.
(4) See Appendix F for additional standards that may be applicable.

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.

9.12 RESERVE UNIT and RELEASE UNIT Commands

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

Table 9-16
RESERVE UNIT and RELEASE UNIT Commands

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 in Table 9-16.

9.12.1 RESERVE UNIT Command. The RESERVE UNIT command (Table 9-16, operation code 16h) shall reserve the specified logical unit for the exclusive use by the requesting initiator or, if third-party reservation option is implemented, to another specified SCSI device.

The reservation shall remain in effect until superseded by another RESERVE UNIT command from the initiator that made the reservation or until released by a RELEASE UNIT command from the same initiator, or a BUS DEVICE RESET message from any initiator, or a "hard" RESET condition. 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 by another initiator, then the

target shall either:

(1) Return a RESERVATION CONFLICT status

(2) Queue the reservation request and disconnect until all previously queued reservations have been released. When the logical unit is available, the target shall reconnect 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, or a RELEASE UNIT command, which shall be ignored, 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 (see 9.12.2).

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 superseded by another RESERVE UNIT command from the initiator that made the reservation or until released by the same initiator, by a BUS DEVICE RESET message from any initiator, or by a "hard" RESET condition. The target shall ignore (i.e., return GOOD status) any attempt made by any other initiator to release the reservation.

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.

An initiator that holds a current reservation may modify that reservation (e.g., switch third-parties) by issuing another RESERVE UNIT command to the same logical unit. The superseding RESERVE UNIT command shall release the previous reservation state only when the new reservation is granted. A superseding reservation takes priority over any previously queued reservation request.

9.12.2 RELEASE UNIT Command. The RELEASE UNIT command (Table 9-16, operation code 17h) shall release the logical unit if it is currently reserved by 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 9.12.1). 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 initiator that is requesting the release and 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.

9.13 ERASE Command

Peripheral Device Type: Sequential Access
Operation Code Type: Optional
Operation Code: 19h

Table 9-17
ERASE Command

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 (Table 9-17) 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 this 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.

9.14 MODE SENSE Command

Peripheral Device Type: Sequential Access
Operation Code Type: Optional
Operation Code: 1Ah

Table 9-18
MODE SENSE Command

Byte \ Bit	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 (Table 9-18) 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 9.11) 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 (Table 9-19) contains a four-byte header, followed by zero or more eight-byte block descriptors, followed by the vendor unique parameters, if any.

Table 9-19
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 write protected 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 are defined in Table 9-15.

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.

9.15 LOAD/UNLOAD Command

Peripheral Device Type: Sequential Access
Operation Code Type: Optional
Operation Code: 1Bh

Table 9-20
LOAD/UNLOAD Command

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 (Table 9-20) requests that the target enable or disable the logical unit for further operations. This command may also be used to request the re-tension 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.

9.16 PREVENT/ALLOW MEDIUM REMOVAL Command

Peripheral Device Type: Sequential Access
Operation Code Type: Optional
Operation Code: 1Eh

Table 9-21
PREVENT/ALLOW MEDIUM REMOVAL Command

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 (Table 9-21) 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.

10. Group 0 Command Descriptions for Printer Devices

The Group 0 commands for printer devices shall be as shown in Table 10-1.

Table 10-1
Group 0 Commands for Printer Devices

Operation Code	Type	Command Name	Section	Page
00h	O	TEST UNIT READY	7.1.1	62
01h	V			
02h	V			
03h	M	REQUEST SENSE	7.1.2	63
04h	O	FORMAT	10.1	152
05h	V			
06h	V			
07h	V			
08h	V			
09h	V			
0Ah	M	PRINT	10.2	153
0Bh	O	SLEW AND PRINT	10.3	154
0Ch	V			
0Dh	V			
0Eh	V			
0Fh	V			
10h	O	FLUSH BUFFER	10.4	155
11h	V			
12h	E	INQUIRY	7.1.3	69
13h	V			
14h	O	RECOVER BUFFERED DATA	10.5	156
15h	O	MODE SELECT	10.6	157
16h	O	RESERVE UNIT	10.7.1	159
17h	O	RELEASE UNIT	10.7.2	160
18h	O	COPY	7.1.4	71
19h	V			
1Ah	O	MODE SENSE	10.8	161
1Bh	O	STOP PRINT	10.9	163
1Ch	O	RECEIVE DIAGNOSTIC RESULTS	7.1.5	80
1Dh	O	SEND DIAGNOSTIC	7.1.6	81
1Eh	R			
1Fh	R			

Key: M = Command implementation is mandatory.
E = Command implementation is required for SCSI devices that support device-independent self-configuring software.
O = Command implementation is optional.
R = Operation code is reserved for future standardization.
V = Operation code is available for vendor unique commands.

10.1 FORMAT Command

Peripheral Device Type: Printer
Operation Code Type: Optional
Operation Code: 04h

Table 10-2
FORMAT Command

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 (Table 10-2) 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.

10.2 PRINT Command

Peripheral Device Type: Printer
Operation Code Type: Mandatory
Operation Code: 0Ah

Table 10-3
PRINT Command

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 (Table 10-3) 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 shall not be considered as an error.

10.3 SLEW AND PRINT Command

Peripheral Device Type: Printer
Operation Code Type: Optional
Operation Code: 0Bh

Table 10-4
SLEW AND PRINT Command

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 (Table 10-4) 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.

10.4 FLUSH BUFFER Command

Peripheral Device Type: Printer
Operation Code Type: Optional
Operation Code: 10h

Table 10-5
FLUSH BUFFER Command

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 (Table 10-5) 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.

10.5 RECOVER BUFFERED DATA Command

Peripheral Device Type: Printer
Operation Code Type: Optional
Operation Code: 14h

Table 10-6
RECOVER BUFFERED DATA Command

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 (Table 10-6) 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 shall not be considered as an error.

10.6 MODE SELECT Command

Peripheral Device Type: Printer
Operation Code Type: Optional
Operation Code: 15h

Table 10-7
MODE SELECT Command

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 (Table 10-7) 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 (Table 10-8) contains a four-byte header, followed by the vendor unique parameters, if any.

Table 10-8
MODE SELECT Parameter List

Byte \ Bit	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 2h through 7h are reserved.

10.7 RESERVE UNIT and RELEASE UNIT Commands

Peripheral Device Type: Printer
Operation Code Type: Optional
Operation Code: 16h and 17h, respectively

Table 10-9
RESERVE UNIT and RELEASE UNIT Commands

Byte \ Bit	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 in Table 10-9.

10.7.1 RESERVE UNIT Command. The RESERVE UNIT command (Table 10-9, operation code 16h) shall reserve the specified logical unit for the exclusive use by the requesting initiator or, if third-party reservation option is implemented, to another specified SCSI device.

The reservation shall remain in effect until superseded by another RESERVE UNIT command from the initiator that made the reservation or until released by a RELEASE UNIT command from the same initiator, or a BUS DEVICE RESET message from any initiator, or a "hard" RESET condition. 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 by another initiator, then the target shall either:

(1) Return a RESERVATION CONFLICT status

(2) Queue the reservation request and disconnect until all previously queued reservations have been released. When the logical unit is available, the target shall reconnect 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, or a RELEASE UNIT command, which shall be ignored, 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 (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 superseded by another RESERVE UNIT command from the initiator that made the reservation or until released by the same initiator, by a BUS DEVICE RESET message from any initiator, or by a "hard" RESET condition. The target shall ignore (i.e., return GOOD status) any attempt made by any other initiator to release the reservation.

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.

An initiator that holds a current reservation may modify that reservation

(e.g., switch third-parties) by issuing another RESERVE UNIT command to the same logical unit. The superseding RESERVE UNIT command shall release the previous reservation state only when the new reservation is granted. A superseding reservation takes priority over any previously queued reservation request.

10.7.2 RELEASE UNIT Command. The RELEASE UNIT command (Table 10-9, operation code 17h) shall release the logical unit if it is currently reserved by 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 initiator that is requesting the release and 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.

10.8 MODE SENSE Command

Peripheral Device Type: Printer
Operation Code Type: Optional
Operation Code: 1Ah

Table 10-10
MODE SENSE Command

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 (Table 10-10) 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 (Table 10-11) contains a four-byte header, followed the vendor unique parameters, if any.

Table 10-11
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.

10.9 STOP PRINT Command

Peripheral Device Type: Printer
Operation Code Type: Optional
Operation Code: 1Bh

Table 10-12
STOP PRINT Command

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 (Table 10-12) 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.

11. Group 0 Command Descriptions for Processor Devices

The Group 0 commands for processor devices shall be as shown in Table 11-1.

Table 11-1
Group 0 Commands for Processor Devices

Operation Code	Type	Command Name	Section	Page
00h	O	TEST UNIT READY	7.1.1	62
01h	V			
02h	V			
03h	M	REQUEST SENSE	7.1.2	63
04h	V			
05h	V			
06h	V			
07h	V			
08h	O	RECEIVE	11.1	165
09h	V			
0Ah	M	SEND	11.2	166
0Bh	V			
0Ch	V			
0Dh	V			
0Eh	V			
0Fh	V			
10h	V			
11h	V			
12h	E	INQUIRY	7.1.3	69
13h	V			
14h	V			
15h	V			
16h	V			
17h	V			
18h	O	COPY	7.1.4	71
19h	R			
1Ah	R			
1Bh	R			
1Ch	O	RECEIVE DIAGNOSTIC RESULTS	7.1.5	80
1Dh	O	SEND DIAGNOSTIC	7.1.6	81
1Eh	R			
1Fh	R			

Key: M = Command implementation is mandatory.
E = Command implementation is required for SCSI devices that support device-independent self-configuring software.
O = Command implementation is optional.
R = Operation code is reserved for future standardization.
V = Operation code is available for vendor unique commands.

11.1 RECEIVE Command

Peripheral Device Type: Processor Devices
Operation Code Type: Optional
Operation Code: 08h

Table 11-2
RECEIVE Command

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 (Table 11-2) 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.

11.2 SEND Command

Peripheral Device Type: Processor Devices
Operation Code Type: Mandatory
Operation Code: 0Ah

Table 11-3
SEND Command

Byte \ Bit	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 (Table 11-3) 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.

12. Command Descriptions for Write-Once Read-Multiple Devices

12.1 Group 0 Commands for Write-Once Read-Multiple Devices. The Group 0 commands for write-once read-multiple devices shall be as shown in Table 12-1.

Table 12-1
Group 0 Commands for Write-Once Read-Multiple Devices

Operation Code	Type	Command Name	Section	Page
00h	O	TEST UNIT READY	7.1.1	62
01h	O	REZERO UNIT	8.1.1	87
02h	V			
03h	M	REQUEST SENSE	7.1.2	63
04h	R			
05h	V			
06h	V			
07h	O	REASSIGN BLOCKS	8.1.3	93
08h	O	READ	12.1.1	168
09h	V			
0Ah	O	WRITE	12.1.2	169
0Bh	O	SEEK	8.1.6	97
0Ch	V			
0Dh	V			
0Eh	V			
0Fh	V			
10h	V			
11h	V			
12h	E	INQUIRY	7.1.3	69
13h	V			
14h	V			
15h	O	MODE SELECT	12.1.3	171
16h	O	RESERVE	8.1.8	102
17h	O	RELEASE	8.1.9	107
18h	O	COPY	7.1.4	71
19h	V			
1Ah	O	MODE SENSE	12.1.4	173
1Bh	O	START/STOP UNIT	8.1.11	111
1Ch	O	RECEIVE DIAGNOSTIC RESULTS	7.1.5	80
1Dh	O	SEND DIAGNOSTIC	7.1.6	81
1Eh	O	PREVENT/ALLOW MEDIUM REMOVAL	8.1.12	112
1Fh	R			

Key: M = Command implementation is mandatory.
E = Command implementation is required for SCSI devices that support device-independent self-configuring software.
O = Command implementation is optional.
R = Operation code is reserved for future standardization.
V = Operation code is available for vendor unique commands.

12.1.1 READ Command

Peripheral Device Type: Write-Once Read-Multiple and Read-Only Direct Access
Operation Code Type: Optional
Operation Code: 08h

Table 12-2
READ Command

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 (Table 12-2) 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 (see 8.1.8) 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.

12.1.2 WRITE Command

Peripheral Device Type: Write-Once Read-Multiple
Operation Code Type: Optional
Operation Code: 0Ah

Table 12-3
WRITE Command

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 (Table 12-3) 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 (see 8.1.8) 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.

12.1.3 MODE SELECT Command

Peripheral Device Type: Write-Once Read-Multiple and Read-Only Direct Access
Operation Code Type: Optional
Operation Code: 15h

Table 12-4
MODE SELECT Command

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 (Table 12-4) 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 (Table 12-5) contains a four-byte header, followed by zero or more eight-byte block descriptors, followed by the vendor unique parameters, if any.

Table 12-5
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.

12.1.4 MODE SENSE Command

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

Table 12-6
MODE SENSE Command

Byte \ Bit	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 (Table 12-6) 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 (Table 12-7) contains a four-byte header, followed by zero or more eight-byte block descriptors, followed by the vendor unique parameters, if any.

Table 12-7
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.

12.2 Group 1 Commands for Write-Once Read-Multiple Devices. The Group 1 commands for write-once read-multiple devices shall be as shown in Table 12-8.

Table 12-8
Group 1 Commands for Write-Once Read-Multiple Devices

Operation Code	Type	Command Name	Section	Page
20h	V			
21h	V			
22h	V			
23h	V			
24h	V			
25h	E	READ CAPACITY	8.2.1	114
26h	V			
27h	V			
28h	M	READ	12.2.1	177
29h	V			
2Ah	M	WRITE	12.2.2	179
2Bh	O	SEEK	8.2.4	119
2Ch	V			
2Dh	V			
2Eh	O	WRITE AND VERIFY	12.2.3	181
2Fh	O	VERIFY	12.2.4	182
30h	O	SEARCH DATA HIGH	8.2.7.1	125
31h	O	SEARCH DATA EQUAL	8.2.7.2	125
32h	O	SEARCH DATA LOW	8.2.7.3	125
33h	O	SET LIMITS	8.2.8	126
34h	R			
35h	R			
36h	R			
37h	R			
38h	R			
39h	O	COMPARE	7.2.1	83
3Ah	O	COPY AND VERIFY	7.2.2	84
3Bh	R			
3Ch	R			
3Dh	R			
3Eh	R			
3Fh	R			

Key: M = Command implementation is mandatory.
E = Command implementation is required for SCSI devices that support device-independent self-configuring software.
O = Command implementation is optional.
R = Operation code is reserved for future standardization.
V = Operation code is available for vendor unique commands.

12.2.1 READ Command

Peripheral Device Type: Write-Once Read-Multiple and Read-Only Direct Access
Operation Code Type: Mandatory
Operation Code: 28h

Table 12-9
READ Command

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 (Table 12-9) 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 (see 8.1.8) 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.

12.2.2 WRITE Command

Peripheral Device Type: Write-Once Read-Multiple
Operation Code Type: Mandatory
Operation Code: 2Ah

Table 12-10
WRITE Command

Byte \ Bit	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 (Table 12-10) 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 (see 8.1.8) 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.

12.2.3 WRITE AND VERIFY Command

Peripheral Device Type: Write-Once Read-Multiple
Operation Code Type: Optional
Operation Code: 2Eh

Table 12-11
WRITE AND VERIFY Command

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 (Table 12-11) 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.

12.2.4 VERIFY Command

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

Table 12-12
VERIFY Command

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 (Table 12-12) 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.

13. Command Descriptions for Read-Only Direct-Access Devices

13.1 Group 0 Commands for Read-Only Direct-Access Devices. The Group 0 commands for read-only direct-access devices shall be as shown in Table 13-1.

Table 13-1
Group 0 Commands for Read-Only Direct-Access Devices

Operation Code	Type	Command Name	Section	Page
00h	O	TEST UNIT READY	7.1.1	62
01h	O	REZERO UNIT	8.1.1	87
02h	V			
03h	M	REQUEST SENSE	7.1.2	63
04h	R			
05h	V			
06h	V			
07h	R			
08h	O	READ	12.1.1	168
09h	V			
0Ah	R			
0Bh	O	SEEK	8.1.6	97
0Ch	V			
0Dh	V			
0Eh	V			
0Fh	V			
10h	V			
11h	V			
12h	E	INQUIRY	7.1.3	69
13h	V			
14h	V			
15h	O	MODE SELECT	12.1.3	171
16h	O	RESERVE	8.1.8	102
17h	O	RELEASE	8.1.9	107
18h	O	COPY	7.1.4	71
19h	V			
1Ah	O	MODE SENSE	12.1.4	173
1Bh	O	START/STOP UNIT	8.1.11	111
1Ch	O	RECEIVE DIAGNOSTIC RESULTS	7.1.5	80
1Dh	O	SEND DIAGNOSTIC	7.1.6	81
1Eh	O	PREVENT/ALLOW MEDIUM REMOVAL	8.1.12	112
1Fh	R			

Key: M = Command implementation is mandatory.
E = Command implementation is required for SCSI devices that support device-independent self-configuring software.
O = Command implementation is optional.
R = Operation code is reserved for future standardization.
V = Operation code is available for vendor unique commands.

13.2 Group 1 Commands for Read-Only Direct-Access Devices. The Group 1 commands for read-only direct-access devices shall be as shown in Table 13-2.

Table 13-2
Group 1 Commands for Read-Only Direct-Access Devices

Operation Code	Type	Command Name	Section	Page
20h	V			
21h	V			
22h	V			
23h	V			
24h	V			
25h	E	READ CAPACITY	8.2.1	114
26h	V			
27h	V			
28h	M	READ	12.2.1	177
29h	V			
2Ah	R			
2Bh	O	SEEK	8.2.4	119
2Ch	V			
2Dh	V			
2Eh	R			
2Fh	O	VERIFY	12.2.4	182
30h	O	SEARCH DATA HIGH	8.2.7.1	125
31h	O	SEARCH DATA EQUAL	8.2.7.2	125
32h	O	SEARCH DATA LOW	8.2.7.3	125
33h	O	SET LIMITS	8.2.8	126
34h	R			
35h	R			
36h	R			
37h	R			
38h	R			
39h	O	COMPARE	7.2.1	83
3Ah	O	COPY AND COMPARE	7.2.2	84
3Bh	R			
3Ch	R			
3Dh	R			
3Eh	R			
3Fh	R			

Key: M = Command implementation is mandatory.
E = Command implementation is required for SCSI devices that support device-independent self-configuring software.
O = Command implementation is optional.
R = Operation code is reserved for future standardization.
V = Operation code is available for vendor unique commands.

14. Status

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 Tables 14-1 and 14-2 unless the command is cleared by an ABORT message, by a BUS DEVICE RESET message, or by a "hard" RESET condition.

Table 14-1
Status Byte

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved	Vendor Unique		Status Byte Code				V

Table 14-2
Status Byte Code Bit Values

Bits of Status Byte								Status(es) Represented
7	6	5	4	3	2	1	0	
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

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 return 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 returned whenever a target is unable to accept a command from an initiator. The normal initiator recovery action is to issue the command again at a later time.

INTERMEDIATE. This status shall be returned 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 returned, the chain of linked commands is broken; no further commands in the series are executed.

RESERVATION CONFLICT. This status shall be returned 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.

Appendixes (These Appendixes are not part of American National Standard X3.131-1986, but are included for information only.)

Appendix A

SCSI Signal Sequence Example

This Appendix is included to provide an example of the signal sequencing of an SCSI command that includes most of the SCSI bus phases (Figure A1). In this example, the target does not disconnect from the SCSI bus prior to completing the command.

The following notes apply to Figure A1:

GENERAL NOTE:

In a typical system, the computer's host adapter acts as the "initiator" and the peripheral device's controller acts as the "target." In general, this standard does not attempt to distinguish between a computer and its host adapter. These functions may be separate or merged. The term "initiator" encompasses both. Similarly, the term "target" does not distinguish between the peripheral device and its controller, which may be separate or merged. The term "SCSI device" refers to a device that may be connected to the SCSI bus. An SCSI device may act as in the initiator role, the target role, or both roles.

TIMING NOTES:

Bus Settle Delay. The minimum time to wait for the SCSI bus to settle after changing certain control signals.

Bus Free Delay. The minimum time that an SCSI device must wait from its detection of BUS FREE phase until it may assert BSY and its ID bit.

Bus Set Delay. The maximum time that an SCSI device may wait to assert BSY and its ID bit after BUS FREE phase was last detected if it intends to participate in the ARBITRATION phase.

Bus Clear Delay. The maximum time for an SCSI device to stop driving all signals after BUS FREE phase is detected, after SEL is received during ARBITRATION phase, or after RST becomes true.

Arbitration Delay. The minimum time that an SCSI device must wait after asserting BSY for arbitration until it may examine the DATA BUS to determine whether it won. There is no maximum time.

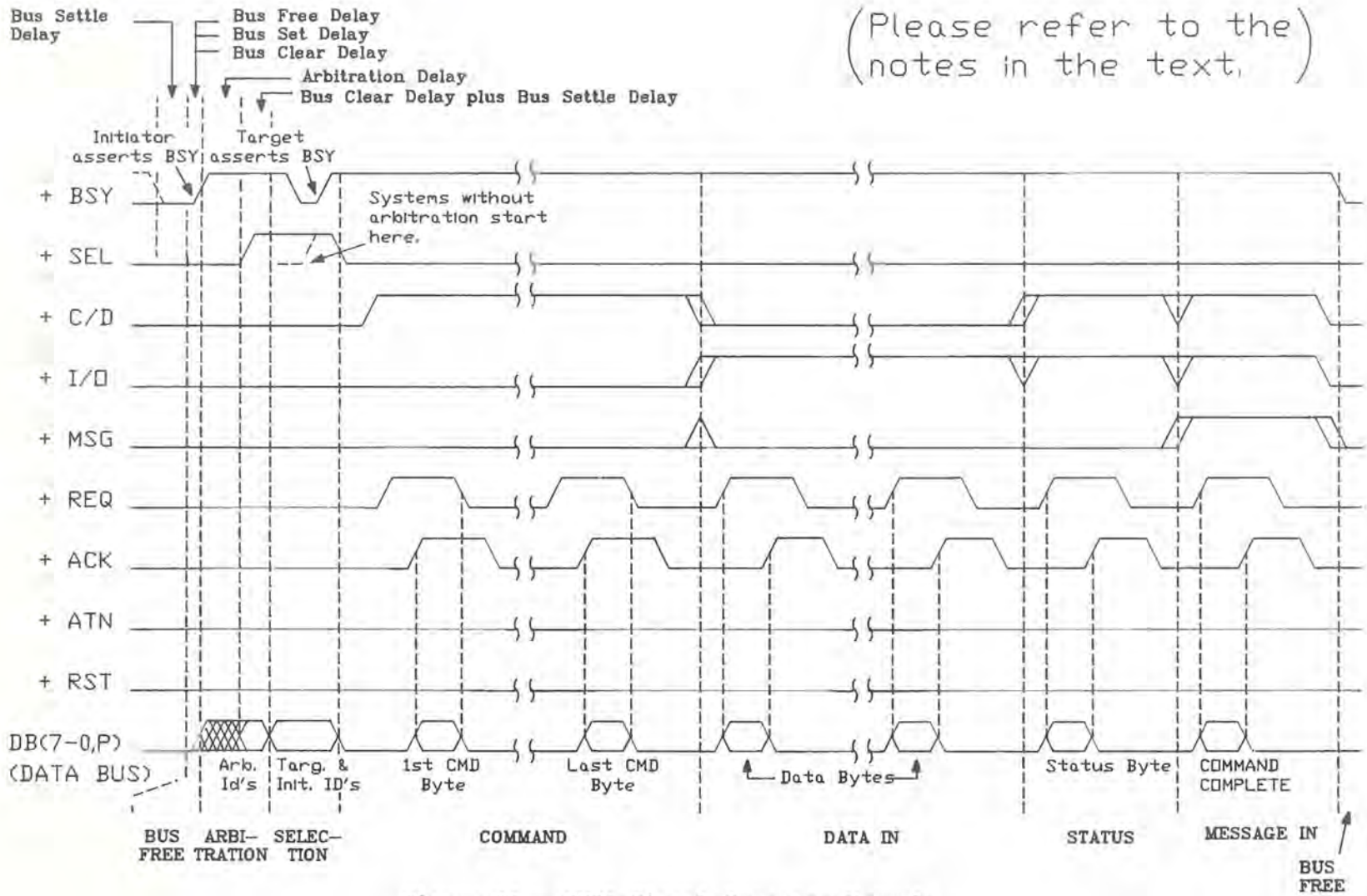


Figure A1. SCSI Signal Sequence Example

DATA BUS NOTES:

(1) DB(7) is the most significant bit.

(2) DB(7) is the highest priority arbitration bit.

(3) DB(P) is the data parity bit (odd). Parity is not valid during the ARBITRATION phase. Use of parity is a system option (i.e., a system is configured so that all SCSI devices on a bus generate parity and have parity detection enabled, or all SCSI devices have parity detection disabled or not implemented).

BUS PHASE NOTES:

BUS FREE phase. BUS FREE phase begins when SEL and BSY are both continuously false for a bus settle delay. It ends when SEL or BSY becomes true.

ARBITRATION phase. Implementation of this phase is optional. If there is more than one initiator, then all initiators must implement this phase. Targets that do not implement this phase cannot disconnect from the bus until the command has completed. SCSI devices that implement the COPY command must implement ARBITRATION phase.

At least one bus free delay after first detecting BUS FREE phase, but no more than a bus set delay after the bus was last free, the initiator asserts BSY and its assigned SCSI device ID bit on the DATA BUS. The initiator waits an arbitration delay, then examines the DATA BUS. If a higher priority SCSI device ID bit is true, the initiator loses arbitration and releases BSY and its ID bit. Otherwise, the initiator wins arbitration and asserts SEL. SCSI devices must release BSY and their ID bit within a bus clear delay after SEL becomes true (even if they have not yet examined the DATA BUS). The winning SCSI device waits at least a bus clear delay plus a bus settle delay after asserting SEL before changing any signals on the bus.

SELECTION phase. The I/O signal is false during this phase to distinguish it from the RESELECTION phase.

NONARBITRATING SYSTEMS: In such systems, the initiator waits at least a bus clear delay after detecting BUS FREE phase, then it asserts the target's ID bit and, optionally, the initiator's ID bit on the DATA BUS. After at least two deskew delays, the initiator asserts SEL.

ARBITRATING SYSTEMS: In such systems, the SCSI device that won arbitration has both BSY and SEL asserted. After at least a bus clear delay plus a bus settle delay, it places both the target's and the initiator's ID bits on the DATA BUS. At least two two deskew delays later, it releases BSY.

ALL SYSTEMS: The target determines that it is selected when SEL and its SCSI ID bit are true and BSY and I/O are false for at least a bus settle delay. The target then asserts BSY within a selection abort time after it last determined that it was still being selected. (The target is not required to respond to a selection within a selection abort time; but it must ensure that it will not assert BSY more than a selection abort time after the initiator aborts a selection attempt.)

At least two deskew delays after the initiator detects BSY true, it releases SEL and may change or release the DATA BUS.

COMMAND phase. The target asserts C/D and negates I/O and MSG for all of the bytes transferred during this phase. The direction of transfer is from the initiator to the target.

HANDSHAKE PROCEDURE: The target asserts REQ. Upon detecting REQ true, the initiator drives the DATA BUS to the desired value, waits at least one deskew delay plus a cable skew delay and then asserts ACK. The initiator continues to drive the DATA BUS until REQ is false.

When ACK is true at the target, the target reads the DATA BUS and then negates REQ.

When REQ becomes false at the initiator, the initiator may change or release the DATA BUS and negate ACK.

The target may continue to request command bytes by asserting REQ again. The number of command bytes is determined by the group code (most significant 3 bits) that is contained in the first command byte.

DATA IN phase. The target asserts I/O and negates C/D and MSG for all of the bytes transferred during this phase. The direction of transfer is from the target to the initiator.

HANDSHAKE PROCEDURE: The target first drives the DATA BUS to their desired values, waits at least one deskew delay plus a cable skew delay, and then asserts REQ. The target continues to drive the DATA BUS until ACK is true.

When REQ is true at the initiator, the initiator reads the DATA BUS and then asserts ACK.

When ACK is true at the target, the target may change or release the DATA BUS and negate REQ.

When REQ is false at the initiator, the initiator negates ACK. After ACK is false, the target may continue the transfer by driving the DATA BUS and asserting REQ as described above.

DATA OUT phase (not shown in the figure). The target negates C/D, I/O, and MSG for all of the bytes transferred during this phase. The direction of transfer is from the initiator to the target. (Refer to the handshake procedure for the COMMAND phase.)

STATUS phase. The target asserts C/D and I/O and negates MSG for the byte transferred during this phase. The direction of transfer is from the target to the initiator. (Refer to the handshake procedure for the DATA IN phase.)

MESSAGE phase. The target C/D, I/O, and MSG during the byte transferred during this phase. Typically, a COMMAND COMPLETE message would be sent at this point. The direction of transfer is from the target to the initiator (Refer to the handshake procedure for the DATA IN phase.)

BUS FREE phase. The target returns to BUS FREE phase by releasing BSY. Both the target and the initiator release all bus signals within a bus clear delay after BSY is continuously false for a bus settle delay.

Appendix B

Typical Bus Phase Sequence

This Appendix is included to provide an example of the SCSI bus phase sequence for a typical READ command (Tables B1 and B2). In this example, the target does not disconnect from the SCSI bus prior to completing the command.

Table B1
Typical READ Command Phase Sequence

	Signals											
Bus Phase	BSY	SEL	ATN	MSG	C/D	I/O	REQ	ACK	RST	DB (7-0)	DB (P)	Comment
BUS FREE	-	-	-	-	-	-	-	-	-	-	-	SCSI bus is available.
ARBITRATION	1	-	-	-	-	-	-	-	-	ID	X	Initiator tries to get
"		1										the SCSI bus.
SELECTION	1	1	1	-	-	-	-	0	-	ID I,T	V	Initiator has SCSI bus
"	-	1								ID I,T	V	and selects a target.
"	1	1								ID I,T	V	ATN is on.
"	1	-								X	X	
MESSAGE OUT	1	-	1	1	1	0	0	0	-	X	X	Target has control of
"			1				1	0		X	X	the bus and gets the
"			1				1	0		X	X	IDENTIFY message from
"			1				1	1		Message	V	the initiator.
"			1				0	1		X	X	
"			0				0	0		X	X	
COMMAND	1	-	0	0	1	0	0	0	-	X	X	Target gets a command
"							1	0		X	X	from the initiator.
"							1	1		Command	V	(This phase is repeated
"							0	1		X	X	for each byte.)
"							0	0		X	X	

Table B2
Typical READ Command Phase Sequence (Continued)

	Signals											
Bus Phase	BSY	SEL	ATN	MSG	C/D	I/O	REQ	ACK	RST	DB (7-0)	DB (P)	Comment
DATA IN	1	-	0	0	0	1	0	0	-	X	X	Target sends data to
"							1	0	Read	Data	V	the initiator. (This
"							1	1		X	X	phase is repeated for
"							0	1		X	X	each byte.)
"							0	0		X	X	
STATUS	1	-	0	0	1	1	0	0	-	X	X	Target sends status to
"							1	0		Status	V	the initiator.
"							1	1		X	X	
"							0	1		X	X	
"							0	0		X	X	
MESSAGE IN	1	-	0	1	1	1	0	0	-	X	X	Target sends a COMMAND
"							1	0		Message	V	COMPLETE message to the
"							1	1		X	X	initiator.
"							0	1		X	X	
"							0	0		X	X	
BUS FREE	-	-	-	-	-	-	-	-	-	-	-	SCSI bus is available.

Key:
- \- = Signal driver is passive.
- 0 = Signal is false.
- 1 = Signal is true.
- "Blank" = Signal state is the same as the previous line.
- ID = SCSI ID for arbitration.
- ID I,T = SCSI ID of initiator and target.
- V = Parity is valid.
- X = The signal is not guaranteed to be in a known state.

Appendix C

SCSI System Operation

This Appendix is included to provide an explanation of the relationship of the various pieces of an SCSI system. This Appendix also provides additional information about the use of SCSI in a multi-tasking system. Such systems typically use a host adapter printed-circuit board to interface from the host memory to the SCSI bus. Although other architectures are possible (including native or imbedded SCSI), the host adapter logic still exists as part of the system. The term "initiator" is used throughout this standard to encompass all such architectures. The term "host adapter" is used within this Appendix to refer to the logic that interfaces from the host memory to the SCSI bus.

C1. Host Memory / Host Adapter / SCSI Controller Relationship

The SCSI architecture utilizes the concept of host memory blocks for command, data, and status interchange between the host system and the SCSI controller. In the middle of this exchange is the SCSI host adapter, which acts as the SCSI peripheral's gateway into host memory. The host adapter 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 host adapter is intimately involved in assuring data integrity and proper performance of the I/O subsystem.

In order to fully understand SCSI operation, the concepts of I/O memory blocks and logical threads must be detailed. Figure C1 presents a block diagram of a single host/single peripheral SCSI I/O subsystem. The host memory contains three I/O blocks: command, data, and status. The SCSI controller 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 controller needs to both read and write the data block. As was previously mentioned, the SCSI controller "reaches into host memory" via the SCSI host adapter. The host adapter must know the addresses of the command, data, and status blocks in order for it to "reach" into the right spot in memory. In other words, the host adapter must be given a pointer to the start of each block by the host. As the SCSI controller 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.

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, data, or status byte to be transferred between the host memory and the SCSI controller. There is only one set of current pointers in the host adapter. The current pointers are shared amoung all devices and are used by the current device connected to the host adapter. The second set is known as the saved pointer values. There is one set of saved pointers for each supported logical thread. 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 command. It remains at this value until the controller sends a SAVE DATA POINTER message to the host adapter which in turn saves the value of the current data pointer. The controller may retrieve the saved value by sending a RESTORE POINTERS message. This moves

the saved value of each pointer into the current pointer registers. Whenever an SCSI device disconnects from the bus, only the saved pointer values are retained. The current pointer values are set from the saved values upon the next reconnection. The current and saved pointers provide an efficient means of error retry and recovery during large data exchanges on the SCSI bus.

C2. SCSI READ Command Example

One method to understanding the host/host adapter/SCSI peripheral relationship is via an example. Let us consider the case of a multiple sector READ command that will cross a cylinder boundary on a direct-access device such as a disk.

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

Upon receiving the 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 signal asserted. The ATTENTION condition indicates to the SCSI target that the initiator (the host adapter) has a message to send to the target.

After the SELECTION phase is completed, the disk controller responds to the initiator's ATTENTION condition by receiving a message from the initiator. This message, generated by the host adapter, indicates the desired logical unit number in the target and whether the initiator can support bus disconnect. In this example, the initiator supports disconnect.

Input/output 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 command descriptor block (CDB).

After decoding the instruction, the controller determines that a disk seek is required to get the starting data block. Since the SCSI bus will not be utilized until data has been read from the disk, the target controller disconnects from the bus. The disconnect process includes the transmission of a SAVE DATA POINTER message and DISCONNECT message from the target to the host adapter. The host adapter responds to the SAVE DATA POINTER message by saving the current data pointer, which is still set to the start of the data block. After transmission of the DISCONNECT message the target will release BSY, freeing the bus.

Although the initiator host adapter and target disk controller are disconnected, they are logically connected or, "threaded", together. Both devices know they have a command to finish and will return to that job at a later point in 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 controller, but runs all the way from the host memory I/O block to the

peripheral device performing the operation. (See Figure C2 for a pictorial presentation of this concept.)

Once the target has started filling its data buffers, it can transmit data to the initiator, but first it must reestablish the physical path. The reselection process involves the target arbitrating for the bus and reselecting the host adapter. After the physical reconnection is made, the target sends an IDENTIFY message to the host adapter to indicate which target logical unit is reconnecting. This information provides reconnection, to the correct thread into host memory. 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 controller determines that a second physical seek is required to complete the READ command. The target again performs a SAVE DATA POINTER message and a DISCONNECT message. However, this time the current data pointer is not at the beginning of the memory data block. The saved value at disconnect reflects the change.

After seek completion and transfer of data into its buffer, the controller reconnects to the host adapter and completes the data transfer as requested by the READ command. At this point, the controller sends ending status into host memory via the host adapter. The final action of the target is to send to the host adapter a COMMAND COMPLETE message and disconnect from the SCSI bus. The target has completed its operation and considers the logical thread broken.

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.

C3. I/O Channel Concept

The I/O channel concept fully utilizes the high performance capability of SCSI. The I/O channel is basically an intelligent SCSI host adapter that can maintain multiple simultaneous threads between host memory I/O blocks and different SCSI devices.

The I/O channel utilizes a single direct memory access (DMA) path into host memory supporting the DMA operations of numerous SCSI peripherals. 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 many implementations of a multiple DMA channel architecture, when a channel is accessing memory, all other channels are idle. In such implementations, a single channel supporting multiple threads can supply the same performance as separate DMA peripherals. An obvious advantage to the host is lower system cost as well as the saving in backplane card slots.

In the READ command example discussed in Section C2, 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 that is assigned the task of managing the physical link and logical thread between the host adapter and the target controller. All physical connections and reconnections to the host adapter are managed by this subchannel. 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 subchannels (14,336 subchannels, if the EXTENDED IDENTIFY message is implemented).

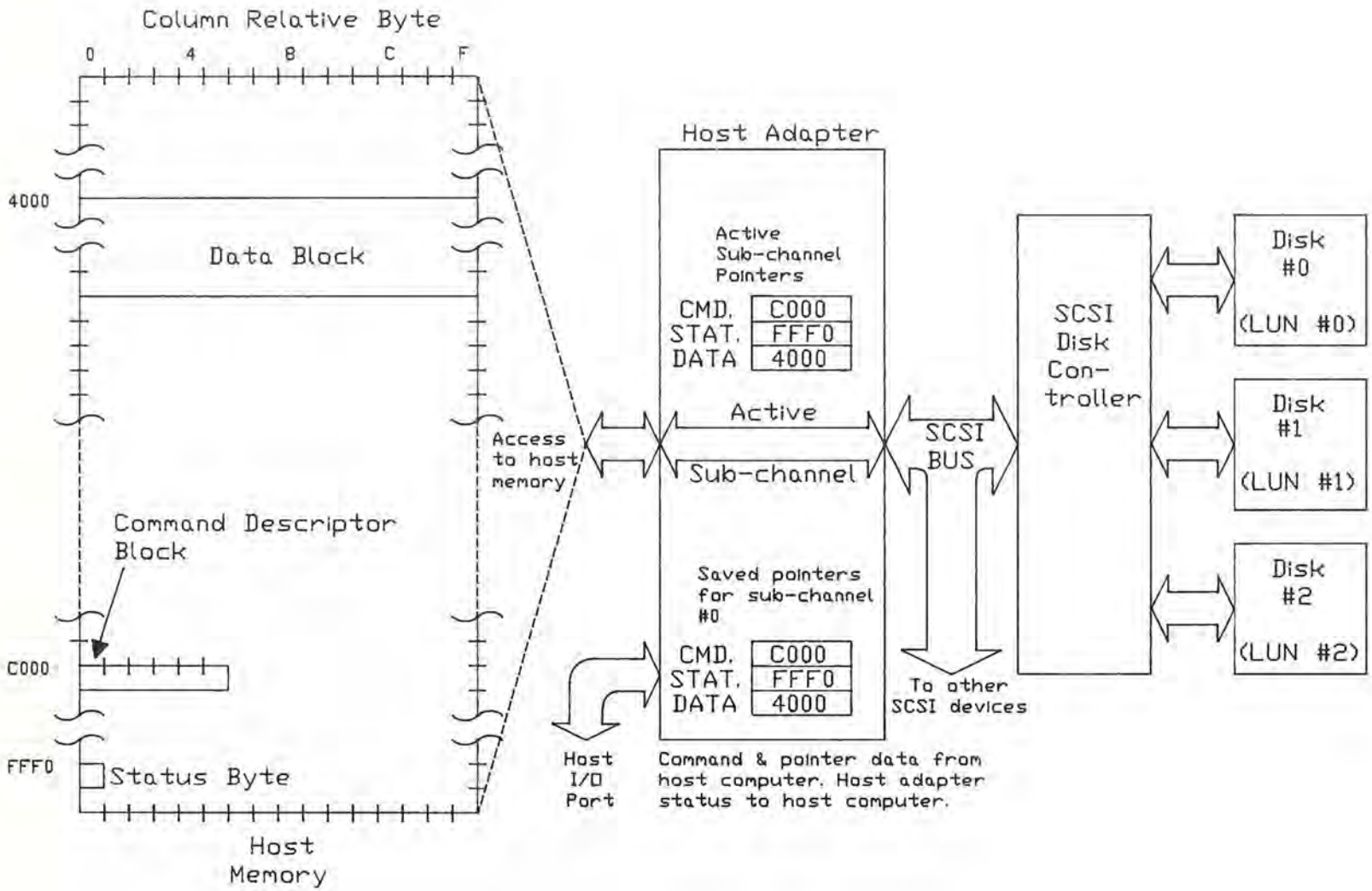


Figure C1. Snapshot Prior to Initial Selection

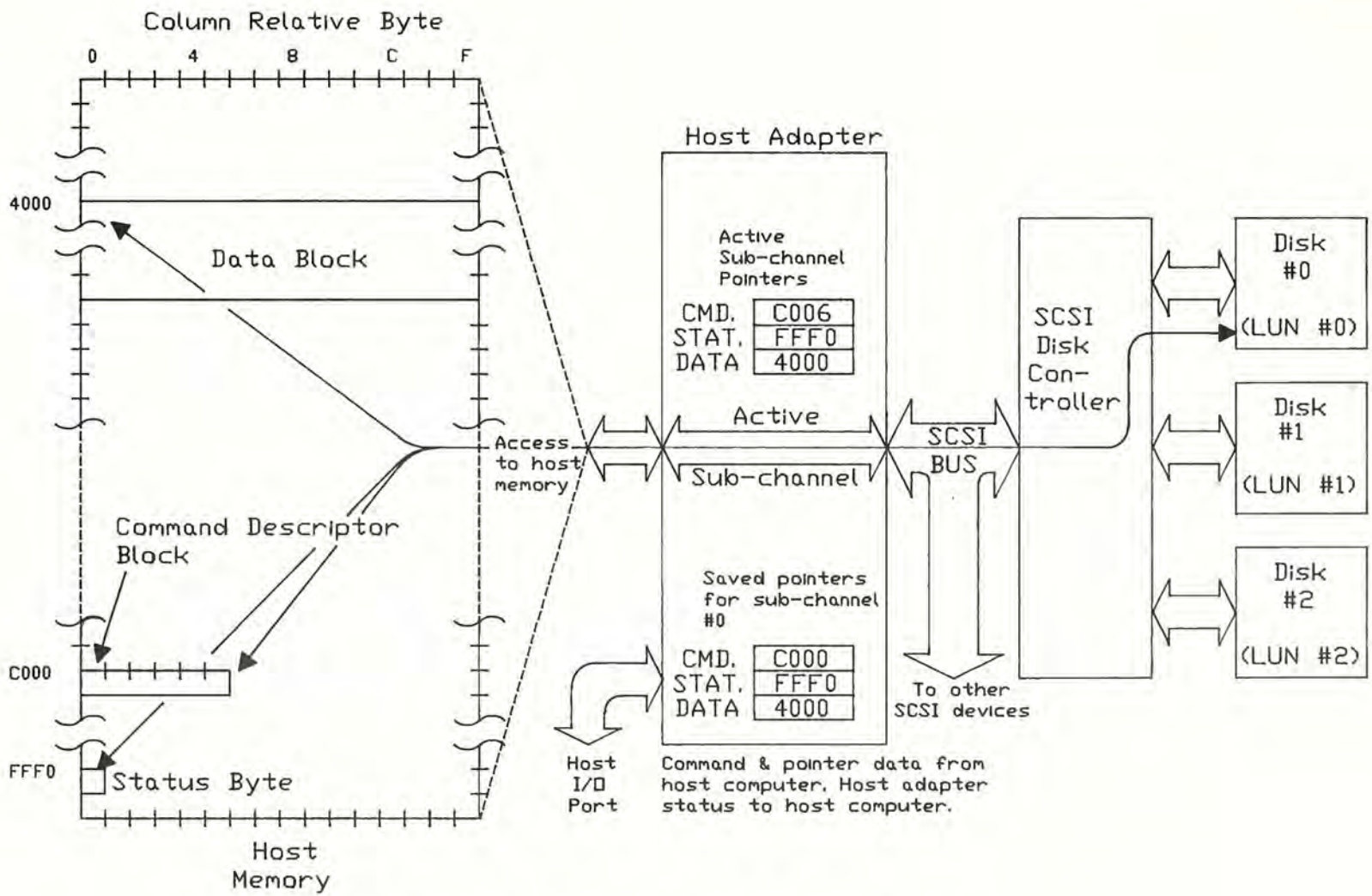


Figure C2. Snapshot Prior to Data Transfer

Appendix D

Recommended Shielded Connectors

Three alternative shielded connector systems are defined by this Appendix. For each alternative, the connector shielding system should provide a dc resistance of less than 10 milliohms from the cable shield at its termination point to the SCSI device enclosure.

In order to support daisy-chain connections, SCSI devices that use shielded connectors should provide two shielded device connectors on the device enclosure. These two connectors may be wired "one-to-one" with a stub to the SCSI device's drivers and receivers provided the maximum stub length is not violated. Alternatively, two cables may be run from the two shielded connectors to the drivers and receivers so that the maximum stub length is not violated. The length of the cable within the device enclosure is included when calculating the total cable length of the SCSI bus.

D1. Shielded Connector, Alternative 1

The shielded cable connector (Figure D1) is a 50-conductor connector consisting of two rows of 25 female contacts with adjacent contacts 2.54 mm (0.1 in) apart. The nonmating portion of the connector is shown for reference only.

The shielded SCSI device connector (Figure D2) is a 50-conductor connector consisting of two rows of 25 male pins with adjacent pins 2.54 mm (0.1 in) apart. The nonmating portion of the connector is shown for reference only.

The connector pin assignments are shown in Table 4-1 for single-ended drivers and are shown in Table 4-2 for differential drivers.

D2. Shielded Connector, Alternative 2

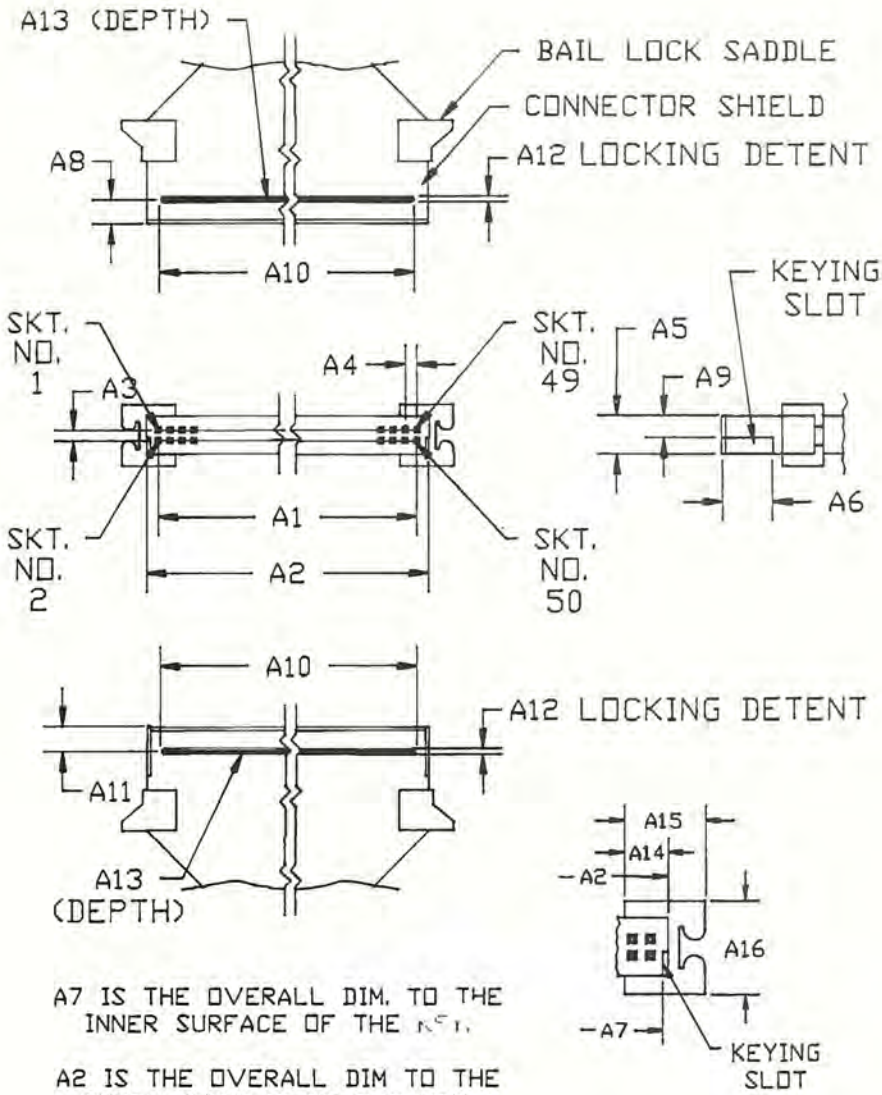
The shielded device connector (Figure D3) is a 50-conductor connector consisting of two rows of ribbon contacts spaced 2.16 mm (0.085 in) apart. The nonmating portion of the connector is shown for reference only. FCC document Part 68 Subpart F 68.500 should be used for reference.

The shielded cable connector (Figure D4) is a 50-conductor connector consisting of two rows of ribbon contacts spaced 2.16 mm (0.085 in) apart. The nonmating portion of the connector is shown for reference only.

The connector pin assignments are shown in Table D1 for single-ended drivers and are shown in Table D2 for differential drivers.

D3. EUROCARD Boxes

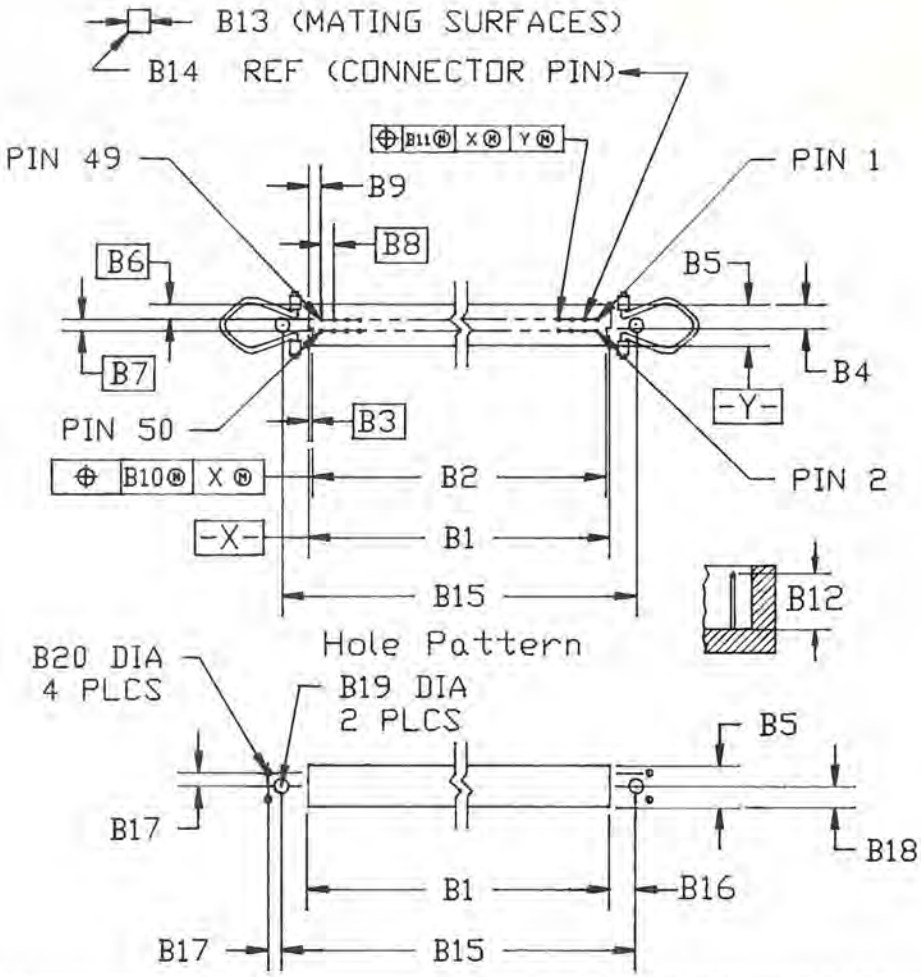
For boards in EUROCARD boxes (IEC 297 or DIN 41494) the EMI-screen may be in the front of the boards and in this case a screened cable and connector may be attached through the front panel to the board connector specified in 4.3. It is thereby possible to use flat ribbon cable or shielded cable, as required by users.



Dimensions	Millimeters	Inches
A1	60.96 ± 0.15	2.400 ± 0.006
A2	66.29 ± 0.18	2.610 ± 0.007
A3	2.54 ± 0.15	0.100 ± 0.006
A4	2.54 ± 0.15	0.100 ± 0.006
A5	8.56 ± 0.41	0.337 ± 0.016
A6	11.93 Minimum	0.470 Minimum
A7	65.02 ± 0.18	2.560 ± 0.007
A8	5.46 ± 0.127	0.215 ± 0.005
A9	4.14 ± 0.25	0.163 ± 0.010
A10	60.20 Maximum	2.370 Maximum
A11	6.60 ± 0.127	0.260 ± 0.005
A12	1.27 Maximum	0.050 Maximum
A13	0.64 Maximum	0.025 Maximum
A14	6.604 + 0.25, − 0.13	0.260 + 0.010, − 0.005
A15	13.46 ± 0.127	0.530 ± 0.005
A16	12.45 ± 0.38	0.490 ± 0.015

NOTE: Fifty contacts on 2.54-mm (0.100-inch) spacing = 60.96 mm (2.40 inch).

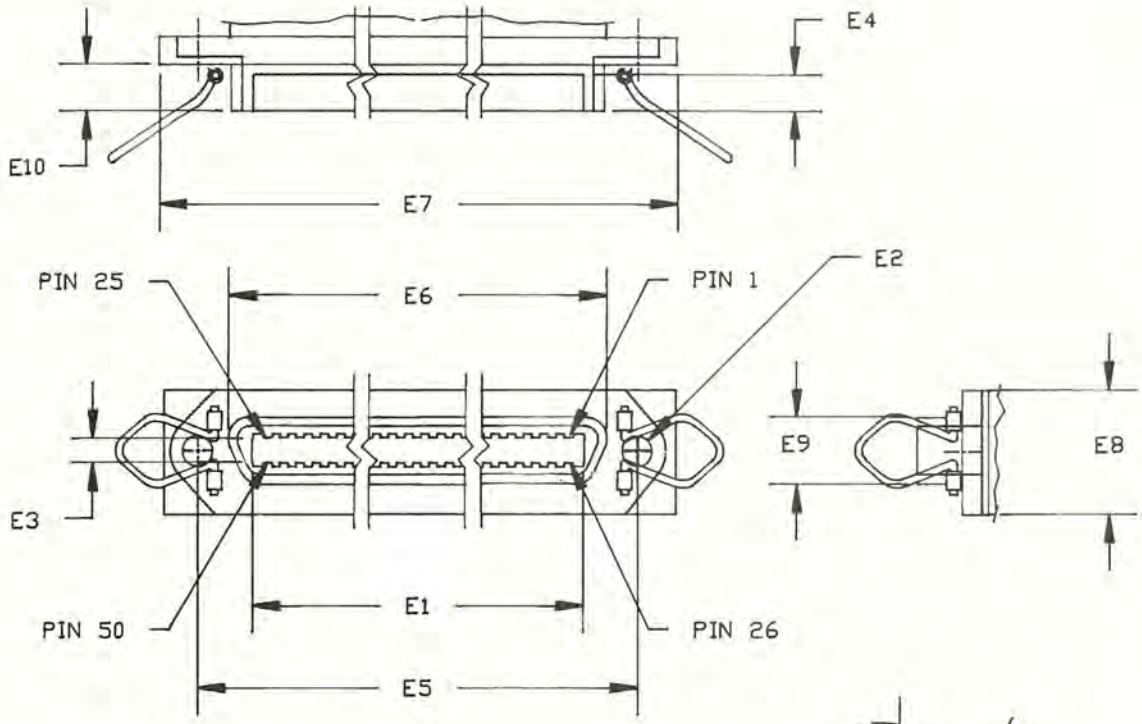
Figure D1. Female Shielded SCSI Cable Connector, Alternative 1

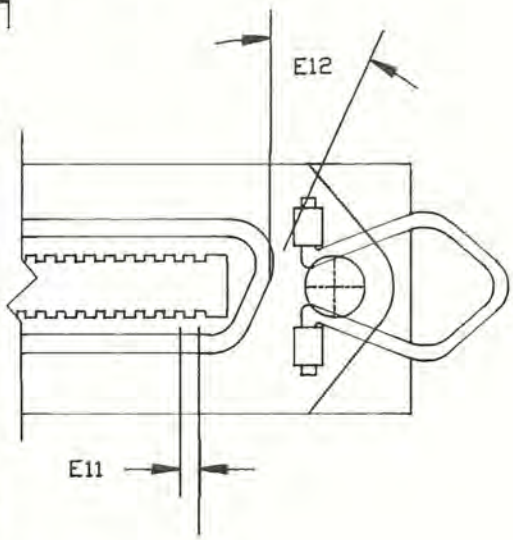


Dimensions	Millimeters	Inches	Comments
B1	66.65 ± 0.15	2.624 ± 0.006	
B2	65.28	2.570	
B3	0.69	0.027	
B4	5.77 ± 0.81	0.227 ± 0.032	
B5	9.14	0.360	
B6	3.30	0.130	
B7	2.54	0.100	
B8	2.54	0.100	
B9	2.84	0.112	Reference Only
B10	0.15	0.006	
B11	0.38	0.015	
B12	8.15 + 0.76, − 0.48	0.321 + 0.030, − 0.019	
B13	0.64 ± 0.02	0.025 ± 0.001	
B14	0.64 Square	0.025 Square	Reference Only
B15	78.84	3.104	
B16	5.77	0.227	Reference Only
B17	2.92	0.115	Reference Only
B18	4.83	0.190	
B19	3.18	0.125	Reference Only
B20	1.52	0.060	Reference Only
B21	67.31	2.650	
B22	9.65	0.380	

NOTES:
(1) Fifty contacts on 2.54-mm (0.100-inch) spacing = 60.96 mm (2.40 inch).
(2) Tolerances ± 0.127 mm (0.005 inch) noncumulative, unless specified otherwise.

Figure D2. Male Shielded SCSI Device Connector, Alternative 1





Dimensions	Millimeters		Inches	
E1	56.39	Minimum	2.220	Minimum
E2 (3)	2.62	Minimum	0.103	Minimum
E3	3.99	Minimum	0.157	Minimum
E4	5.84	Minimum	0.230	Minimum
E5	74.85		2.947	
E6	64.29	Maximum	2.531	Maximum
E7*	83.06		3.270	
E8*	15.24		0.600	
E9	12.04	Maximum	0.474	Maximum
E10	9.78	Maximum	0.385	Maximum
E11	2.16		0.085	
E12	15° ± 2°		15° ± 2°	

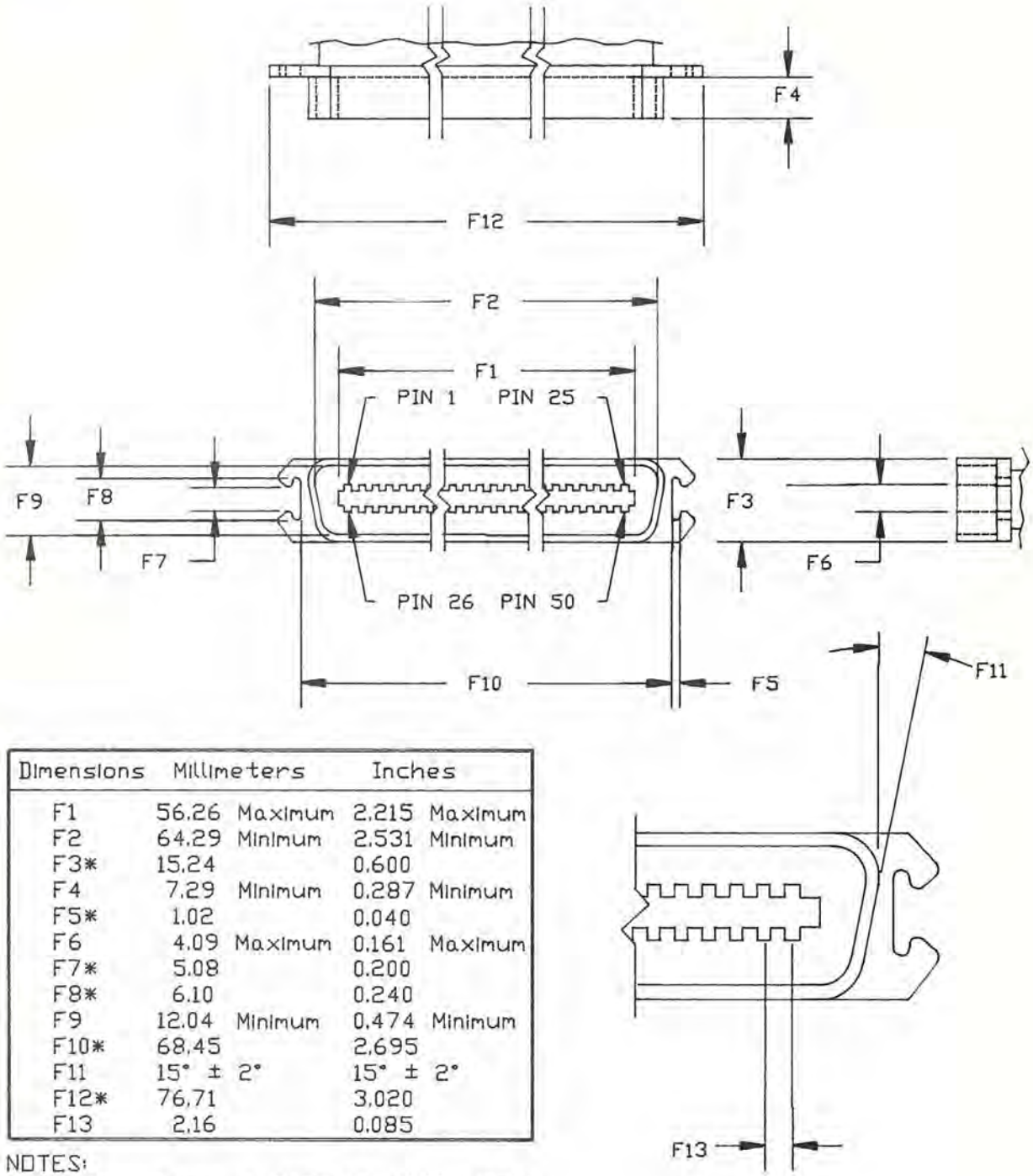
NOTES:

(1) Tolerances ± 0.127 mm (0.005 inch) noncumulative, unless specified otherwise.

(2) Dimensions listed with asterisks are shown for reference only.

(3) Dimension E2 to accomodate 4-40 or 6-32 threaded screws.

Figure D3. Shielded SCSI Device Connector, Alternative 2



Dimensions	Millimeters		Inches	
F1	56.26	Maximum	2.215	Maximum
F2	64.29	Minimum	2.531	Minimum
F3*	15.24		0.600	
F4	7.29	Minimum	0.287	Minimum
F5*	1.02		0.040	
F6	4.09	Maximum	0.161	Maximum
F7*	5.08		0.200	
F8*	6.10		0.240	
F9	12.04	Minimum	0.474	Minimum
F10*	68.45		2.695	
F11	15° ± 2°		15° ± 2°	
F12*	76.71		3.020	
F13	2.16		0.085	

NOTES:

(1) Tolerances ± 0.127 mm (0.005 inch) noncumulative, unless specified otherwise.

(2) Dimensions listed with asterisks are shown for reference only.

Figure D4. Shielded SCSI Cable Connector, Alternative 2

Table D1
Single-Ended Pin Assignments
(Shielded Connector, Alternative 2)

Signal	Pin Number
-DB(0)	26
-DB(1)	27
-DB(2)	28
-DB(3)	29
-DB(4)	30
-DB(5)	31
-DB(6)	32
-DB(7)	33
-DB(P)	34
GROUND	35
GROUND	36
GROUND	37
TERMPWR	38
GROUND	39
GROUND	40
-ATN	41
GROUND	42
-BSY	43
-ACK	44
-RST	45
-MSG	46
-SEL	47
-C/D	48
-REQ	49
-I/O	50

NOTES:

(1) Pins 1—12 and 14—25 are connected to ground. Pin 13 should be left open. Some products designed prior to the generation of this standard connected this pin to ground.

(2) The minus sign next to the signals indicates active low.

Table D2
Differential Pin Assignments
(Shielded Connector, Alternative 2)

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

NOTE: SHIELD GROUND is optional on some cables. (Implementors note: Some shielded flat ribbon cables use pin 1 as a connection to the shield.)

Appendix E

Conformance

This Appendix contains recommendations on conformance statements to this standard.

E1. Alternatives

This standard contains various alternatives that are mutually exclusive within a system.

(1) Single-ended or differential drivers
(2) Termination power supplied by the cable or not
(3) Parity implemented or not
(4) "Hard" RESET or "soft" RESET
(5) Reservation queuing implemented or not

E2. Levels of Conformance

By specifying which alternatives (listed above) are implemented, an environment is created enabling different features to be implemented. These features are divided into three nested minimum levels as shown in Table E1.

Table E1
Levels of Conformance

Level	Initiator	Target
0	Accept COMMAND COMPLETE message. Accept GOOD and CHECK CONDITION status codes.	Implement the mandatory commands of Section 7. Implement the mandatory commands of the supported device type(s). Implement the COMMAND COMPLETE message. If messages other than COMMAND COMPLETE are implemented, then implement MESSAGE REJECT message also. Implement GOOD and CHECK CONDITION statuses.
1	Same as above plus ARBITRATION phase.	Same as above.
2	Same as above plus accept DISCONNECT, MESSAGE REJECT, IDENTIFY, and SAVE DATA POINTER messages	Same as above plus implement the MESSAGE REJECT and IDENTIFY messages and implement all extended commands in Section 7 and for the supported device type(s).

E3. Options

E3.1 Optional Commands. This standard specifies a number of additional commands that can be implemented in each level. Some of these commands, if implemented, require the implementation of other optional commands, messages, or both. In this case, these additional commands, messages, or both should also be implemented.

E3.2 Optional Messages. This standard specifies a number of additional optional messages that may be implemented at any level.

E3.3 Other options. Within some commands, additional features, called "options", are described. These options need not necessarily be implemented.

E4. Statement of Conformance

Any statement of conformance to this standard should declare which of the alternates listed in Section E1 and which of the three conformance levels are implemented. In addition, it should indicate which of the options listed in Section E3 are implemented, if any. In the case of optional messages, the conformance statement should state whether the message is generated, accepted,

or both.

If the synchronous data transfer option is implemented, any statement of conformance should so indicate and state both the maximum REQ/ACK offset and the minimum transfer period implemented.

Appendix F

Additional Medium Type and Density Code Standards

In Sections 8 and 9 of this standard, the MODE SELECT and MODE SENSE commands define medium type codes and density codes for certain flexible disks and magnetic tapes. American National Standards are referenced for code values if a standard exists for that code value. In many cases, other standards or X3 draft documents also exist for a code value. Tables F1 and F2 in this Appendix provide additional references to those standards or draft documents.

DISCLAIMER: It is not the purpose of this Appendix to indicate that these standards are exactly equivalent to each other. However, these standards may be applicable. Please refer to Sections 8 and 9 for additional information.

Table F1
Direct-Access Medium-Type Codes

Code Value	Medium Type			
00h-02h	See Section 8.			
	Flexible-Disk Reference Standard(s)			
	ANSI	ECMA	ISO	
05h	X3.73-1980,	ECMA-54,	ISO 5654/1-1984,	ISO 5654/2-1985
06h		ECMA-59		
09h	(None — X3B8 has abandoned this project.)			
0Ah	X3.121-1984,	ECMA-69,	ISO 7065/1-1985,	ISO 7065/2-1985
0Dh	X3.82-1980,	ECMA-66,	ISO 6596/1-1985,	ISO 6596/2-1985
12h	X3.125-1985,	ECMA-70,	ISO 7487/1-1985,	ISO 7487/2-1985,
			ISO 7487/3-1984	
16h	X3.126-1986,	ECMA-78,	DIS 8378/1-1984,	DIS 8378/2,
			DIS 8378/3	
1Ah	X3B8/86-32 (Note 1),	ECMA-99,	DIS 8630/1-1985,	DIS 8630/2-1985
1Eh	X3.137 (Note 1),	ECMA-100,	DIS 8860/1-1985,	DIS 8860/2-1985
	Direct-Access Magnetic Tape Standard(s)			
40h	X3B5/85-151 (Note 2),	ECMA TC19/83/39		
44h	X3B5/85-151 (Note 2),	ECMA TC19/83/39		
80h — FFh	Vendor unique			
All others	Reserved			

NOTES:
(1) These listings are currently under development. Please check with the Secretariat for information concerning status and availability.
(2) This draft document is for unrecorded miniature cartridge media. The usage referred to here is for serial GCR recording using a format known as QIC-100. Since Subcommittee X3B5 issues a new document number for each revision of their working draft document, please contact the chairman of X3B5 for the latest document number.

Table F2
Sequential-Access Density Codes

Code Value	Density		
00h	See Section 9.		
	Magnetic Tape Reference Standard(s)		
	ANSI	ECMA	ISO
01h	X3.22-1983,	ECMA-62,	ISO 1863-1976
02h	X3.39-1986,	ECMA-62,	ISO 3788-1976
03h	X3.54-1986,	ECMA-62,	ISO 5652-1984
04h	Old format known as QIC-11		
05h	X3.136-1986,	ECMA-98	
06h	X3B5/85-194A (See Note)		
07h	X3.116-1986,	ECMA-79,	ISO 8063/1-1984
08h	X3B5/86-099 (See Note)		
09h	X3B5/86-055 (See Note)		
0Ah	X3B5/85-88 (See Note)		
0Bh	X3.55-1982, X3.56-1986,	ECMA-46,	ISO 4057-1979
80h — FFh	Vendor unique		
All others	Reserved		

NOTE: Draft document. Subcommittee X3B5 assigns a new document number to each revision of their documents. Please contact the chairman of X3B5 for the latest document number.

Appendix G

Future Standardization

This Appendix is included to provide insight into some possible future extensions to SCSI that are being considered by Task Group X3T9.2. Although X3T9.2 has addressed some of these extensions, many details remain to be resolved. Further work by X3T9.2 may result in changes or additions to these extensions. Please contact the chairman of X3T9.2 for the current status of this activity.

Six new commands are being considered, three for targets that support caching, one to provide a means to read the medium defect data, and two to provide a standardized method for writing and reading the target's data buffer (principally intended to support start-of-day tests that check the integrity of the data path). The tentative operation codes are as follows:

Operation Code	Type	Command Name	Device Types
34h	O	PRE-FETCH	\ Direct Access, Write-Once
35h	O	FLUSH CACHE	> Read-Multiple, and
36h	O	LOCK/UNLOCK CACHE	/ Read-Only Direct Access
37h	O	READ DEFECT DATA	Direct Access
3Bh	O	WRITE BUFFER	All
3Ch	O	READ BUFFER	All

New parameters are being considered in several of the existing commands:

(1) Cache control bits are being considered for bits 3 and 4 of byte 1 in several group 1 commands, including READ, WRITE, WRITE AND VERIFY, and VERIFY.

(2) Byte 1 of the Defect List in the FORMAT UNIT command may be used to provide additional defect management control.

(3) Bits 0 – 3 of byte 3 of the Inquiry Data in the INQUIRY command may be used to identify that the device supports certain additions to the standard. One such addition that is being considered is optional fields within the Inquiry Data to identify the manufacturer and the product model information.

(4) Extensions to the MODE SELECT and MODE SENSE commands are being considered to provide a standard means to control certain target parameters. Byte 2 of the MODE SENSE command is being considered for use in controlling which target parameters are returned by this command.

X3.115-1984 Unformatted 80 Megabyte Trident Pack for Use at 370 tpi and 6000 bpi (General, Physical, and Magnetic Characteristics)
X3.116-1986 Recorded Magnetic Tape Cartridge, 4-Track, Serial 0.250 Inch (6.30 mm) 6400 bpi (252 bpmm), Inverted Modified Frequency Modulation Encoded
X3.117-1984 Printable/Image Areas for Text and Facsimile Communication Equipment
X3.118-1984 Financial Services — Personal Identification Number — PIN Pad
X3.119-1984 Contact Start/Stop Storage Disk, 158361 Flux Transitions per Track, 8.268 Inch (210 mm) Outer Diameter and 3.937 inch (100 mm) Inner Diameter
X3.120-1984 Contact Start/Stop Storage Disk
X3.121-1984 Two-Sided, Unformatted, 8-Inch (200-mm), 48-tpi, Double-Density, Flexible Disk Cartridge for 13 262 ftpr Two-Headed Application
X3.122-1986 Computer Graphics Metafile for the Storage and Transfer of Picture Description Information
X3.124-1985 Graphical Kernel System (GKS) Functional Description
X3.124.1-1985 Graphical Kernel System (GKS) FORTRAN Binding
X3.125-1985 Two-Sided, Double-Density, Unformatted 5.25-inch (130-mm), 48-tpi (1,9-tpmm), Flexible Disk Cartridge for 7958 bpr Use
X3.126-1986 One- or Two-Sided Double-Density Unformatted 5.25-inch (130-mm), 96 Tracks per Inch, Flexible Disk Cartridge
X3.127-1987 Unrecorded Magnetic Tape Cartridge for Information Interchange
X3.128-1986 Contact Start-Stop Storage Disk — 83 000 Flux Transitions per Track, 130-mm (5.118-in) Outer Diameter and 40-mm (1.575-in) Inner Diameter
X3.129-1986 Intelligent Peripheral Interface, Physical Level
X3.130-1986 Intelligent Peripheral Interface, Logical Device Specific Command Sets for Magnetic Disk Drive
X3.131-1986 Small Computer Systems Interface
X3.132-1987 Intelligent Peripheral Interface — Logical Device Generic Command Set for Optical and Magnetic Disks
X3.133-1986 Database Language —NDL
X3.135-1986 Database Language — SQL
X3.136-1986 Serial Recorded Magnetic Tape Cartridge for Information Interchange, Four and Nine Track
X3.139-1987 Fiber Distributed Data Interface (FDDI) Token Ring Media Access Control (MAC)
X3.140-1986 Open Systems Interconnection — Connection Oriented Transport Layer Protocol Specification
X3.141-1987 Data Communication Systems and Services — Measurement Methods for User-Oriented Performance Evaluation
X3.146-1987 Device Level Interface for Streaming Cartridge and Cassette Tape Drives
X3.147-1987 Intelligent Peripheral Interface — Logical Device Generic Command Set for Magnetic Tapes
X3.153-1987 Open Systems Interconnection — Basic Connection Oriented Session Protocol Specification
X3.156-1987 Nominal 8-Inch Rigid Disk Removable Cartridge
X3.157-1987 Recorded Magnetic Tape for Information Interchange, 3200 CPI
X3.158-1987 Serial Recorded Magnetic Tape Cassette for Information Interchange, 0.150 Inch (3.81 mm), 8000 bpi (315 bpmm), Group Code Recording.
X11.1-1977 Programming Language MUMPS
IEEE 416-1978 Abbreviated Test Language for All Systems (ATLAS)
IEEE 716-1982 Standard C/ATLAS Language
IEEE 717-1982 Standard C/ATLAS Syntax
IEEE 770X3.97-1983 Programming Language PASCAL
IEEE 771-1980 Guide to the Use of ATLAS
ISO 8211-1986 Specifications for a Data Descriptive File for Information Interchange
MIL-STD-1815A-1983 Reference Manual for the Ada Programming Language
NBS-ICST 1-1986 Fingerprint Identification — Data Format for Information Interchange

---

X3/TRI-82 Dictionary for Information Processing Systems (Technical Report)

American National Standards for Information Processing

X3.1-1987 Synchronous Signaling Rates for Data Transmission
X3.2-1970 Print Specifications for Magnetic Ink Character Recognition
X3.4-1986 Coded Character Sets — 7-Bit ASCII
X3.5-1970 Flowchart Symbols and Their Usage
X3.6-1965 Perforated Tape Code
X3.9-1978 Programming Language FORTRAN
X3.11-1969 General Purpose Paper Cards
X3.14-1983 Recorded Magnetic Tape (200 CPI, NRZI)
X3.15-1976 Bit Sequencing of the American National Standard Code for Information Interchange in Serial-by-Bit Data Transmission
X3.16-1976 Character Structure and Character Parity Sense for Serial-by-Bit Data Communication in the American National Standard Code for Information Interchange
X3.17-1981 Character Set for Optical Character Recognition (OCR-A)
X3.18-1974 One-Inch Perforated Paper Tape
X3.19-1974 Eleven-Sixteenths-Inch Perforated Paper Tape
X3.20-1967 Take-Up Reels for One-Inch Perforated Tape
X3.21-1967 Rectangular Holes in Twelve-Row Punched Cards
X3.22-1983 Recorded Magnetic Tape (800 CPI, NRZI)
X3.23-1985 Programming Language COBOL
X3.25-1976 Character Structure and Character Parity Sense for Parallel-by-Bit Data Communication in the American National Standard Code for Information Interchange
X3.26-1980 Hollerith Punched Card Code
X3.27-1987 Magnetic Tape Labels and File Structure
X3.28-1976 Procedures for the Use of the Communication Control Characters of American National Standard Code for Information Interchange in Specified Data Communication Links
X3.29-1971 Specifications for Properties of Unpunched Oiled Paper Perforator Tape
X3.30-1986 Representation for Calendar Date and Ordinal Date
X3.31-1973 Structure for the Identification of the Counties of the United States
X3.32-1973 Graphic Representation of the Control Characters of American National Standard Code for Information Interchange
X3.34-1972 Interchange Rolls of Perforated Tape
X3.37-1987 Programming Language APT
X3.38-1972 Identification of States of the United States (Including the District of Columbia)
X3.39-1986 Recorded Magnetic Tape (1600 CPI, PE)
X3.40-1983 Unrecorded Magnetic Tape (9-Track 800 CPI, NRZI; 1600 CPI, PE; and 6250 CPI, GCR)
X3.41-1974 Code Extension Techniques for Use with the 7-Bit Coded Character Set of American National Standard Code for Information Interchange
X3.42-1975 Representation of Numeric Values in Character Strings
X3.43-1986 Representations of Local Time of Day
X3.44-1974 Determination of the Performance of Data Communication Systems
X3.45-1982 Character Set for Handprinting
X3.46-1974 Unrecorded Magnetic Six-Disk Pack (General, Physical, and Magnetic Characteristics)
X3.47-1977 Structure for the Identification of Named Populated Places and Related Entities of the States of the United States for Information Interchange
X3.48-1986 Magnetic Tape Cassettes (3.81-mm [0.150-Inch] Tape at 32 bpmm [800 bpi], PE)
X3.49-1975 Character Set for Optical Character Recognition (OCR-B)
X3.50-1986 Representations for U.S. Customary, SI, and Other Units to Be Used in Systems with Limited Character Sets
X3.51-1986 Representations of Universal Time, Local Time Differentials, and United States Time Zone References
X3.52-1976 Unrecorded Single-Disk Cartridge (Front Loading, 2200 BPI) (General, Physical, and Magnetic Requirements)
X3.53-1976 Programming Language PL/I
X3.54-1986 Recorded Magnetic Tape (6250 CPI, Group Coded Recording)
X3.55-1982 Unrecorded Magnetic Tape Cartridge, 0.250 Inch (6.30 mm), 1600 bpi (63 bpmm), Phase encoded
X3.56-1986 Recorded Magnetic Tape Cartridge, 4 Track, 0.250 Inch (6.30 mm), 1600 bpi (63 bpmm), Phase Encoded
X3.57-1977 Structure for Formatting Message Headings Using the American National Standard Code for Information Interchange for Data Communication Systems Control
X3.58-1977 Unrecorded Eleven-Disk Pack (General, Physical, and Magnetic Requirements)
X3.60-1978 Programming Language Minimal BASIC
X3.61-1986 Representation of Geographic Point Locations
X3.62-1987 Paper Used in Optical Character Recognition (OCR) Systems
X3.63-1981 Unrecorded Twelve-Disk Pack (100 Megabytes) (General, Physical, and Magnetic Requirements)
X3.64-1979 Additional Controls for Use with American National Standard Code for Information Interchange
X3.66-1979 Advanced Data Communication Control Procedures (ADCCP)
X3.72-1981 Parallel Recorded Magnetic Tape Cartridge, 4 Track, 0.250 Inch (6.30 mm), 1600 bpi (63 bpmm), Phase Encoded
X3.73-1980 Single-Sided Unformatted Flexible Disk Cartridge (for 6631-BPR Use)
X3.74-1987 Programming Language PL/I, General-Purpose Subset
X3.76-1981 Unformatted Single-Disk Cartridge (Top Loading 200 tpi 4400 bpi) (General, Physical, and Magnetic Requirements)
X3.77-1980 Representation of Pocket Select Characters
X3.78-1981 Representation of Vertical Carriage Positioning Characters in Information Interchange
X3.79-1981 Determination of Performance of Data Communications Systems That Use Bit-Oriented Communication Procedures
X3.80-1981 Interfaces between Flexible Disk Cartridge Drives and Their Host Controllers
X3.82-1980 One-Sided Single-Density Unformatted 5.25-Inch Flexible Disk Cartridge (for 3979-BPR Use)
X3.83-1980 ANSI Sponsorship Procedures for ISO Registration According to ISO 2375
X3.84-1981 Unformatted Twelve-Disk Pack (200 Megabytes)(General, Physical, and Magnetic Requirements
X3.85-1981 1/2-Inch Magnetic Tape Interchange Using a Self Loading Cartridge
X3.86-1980 Optical Character Recognition (OCR) Inks
X3.88-1981 Computer Program Abstracts
X3.89-1981 Unrecorded Single-Disk, Double-Density Cartridge (Front Loading, 2200 bpi, 200 tpi) (General, Physical, and Magnetic Requirements)
X3.91M-1987 Storage Module Interfaces
X3.92-1981 Data Encryption Algorithm
X3.93M-1981 OCR Character Positioning
X3.94-1985 Programming Language PANCM
X3.95-1982 Microprocessors — Hexadecimal Input/Output, Using 5-Bit and 7-Bit Teleprinters
X3.96-1983 Continuous Business Forms (Single-Part)
X3.98-1983 Text Information Interchange in Page Image Format (PIF)
X3.99-1983 Print Quality Guideline for Optical Character Recognition (OCR)
X3.100-1983 Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment for Packet Mode Operation with Packet Switched Data Communications Network
X3.101-1984 Interfaces Between Rigid Disk Drive(s) and Host(s)
X3.102-1983 Data Communication Systems and Services — User-Oriented Performance Parameters
X3.103-1983 Unrecorded Magnetic Tape Minicassette for Information Interchange, Coplanar 3.81 mm (0.150 in)
X3.104-1983 Recorded Magnetic Tape Minicassette for Information Interchange, Coplanar 3.81 mm (0.150 in), Phase Encoded
X3.105-1983 Data Link Encryption
X3.106-1983 Modes of Operation for the Data Encryption Algorithm
X3.110-1983 Videotex/Teletext Presentation Level Protocol Syntax
X3.111-1986 Optical Character Recognition (OCR) Matrix Character Sets for OCR-M
X3.112-1984 14-in (356-mm) Diameter Low-Surface-Friction Magnetic Storage Disk
X3.113-1987 Programming Language FULL BASIC
X3.114-1984 Alphanumeric Machines; Coded Character Sets for Keyboard Arrangements in ANSI X4.23-1982 and X4.22-1983

(Continued on reverse)

NIST-772
(REV. 10-88)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

FIPS PUBLICATION CHANGE NOTICE

CHANGE NUMBER 2 - FIPS 60-2 & 62
1-FIPS 61-1,63-1,97,111,130&131

DATE OF CHANGE
1990 December 26

FIPS PUBLICATION NUMBER
See above.

PUBLICATION TITLE FIPS 60-2, I/O Channel Interface; 62, Operational Specifications for Magnetic Tape Subsystems; 61-1, Channel Level Power Control Interface; 63-1, Operational Specifications for Variable Block Rotating Mass Storage Subsystems; 97, Operational Specifications for Fixed Block Rotating Mass Storage Subsystems; 111, Storage Module Interfaces (w/extens. for enhanced storage module interface); 130, Intelligent Peripheral Interface (IPI); 131, Small Computer System Interface (SCSI).

THIS OFFICE HAS A RECORD OF YOUR INTEREST IN RECEIVING CHANGES TO THE ABOVE FIPS PUBLICATION. THE CHANGE(S) INDICATED BELOW HAVE BEEN PROVIDED BY THE MAINTENANCE AGENCY FOR THIS PUBLICATION AND WILL BE INCLUDED IN THE NEXT PUBLISHED REVISION TO THIS FIPS PUBLICATION. QUESTIONS OR REQUESTS FOR ADDITIONAL INFORMATION SHOULD BE ADDRESSED TO THE MAINTENANCE AGENCY:

Department of Commerce
National Institute of Standards and Technology
National Computer Systems Laboratory
Gaithersburg, MD 20899

CHANGE ITEM(S)

Attached is a reprint from the December 18, 1990, FEDERAL REGISTER (55 FR 51941) which provides approved revisions by the Secretary of Commerce to the FIPS family of input/output interface standards, and the approved discontinuation of the Exclusion and Verification Lists for these standards.

These approved revisions became effective on December 18, 1990, and become an integral part of FIPS 60-2, 61-1, 62, 63-1, 97, 111, 130 and 131, and, as such, are considered to be included whenever reference is made to them.

These approved revisions should be filed with each FIPS listed above.

Attachment

Copies of FIPS are available from:

National Technical Information Service (NTIS)
ATTN: Sales Office, Sills Building
5285 Port Royal Road
Springfield, Virginia 22161

Phone - 703/487-4650 Office Hours - 7:45 a.m. to 4:15 p.m.

ELECTRONIC FORM

12-18-90
Vol. 55 No. 243
Pages 51895-52036

federal register

Tuesday
December 18, 1990

National Institute of Standards and Technology
NOTICES
Information processing standards, Federal:
Family of input/output interface standards, 51941

National Institute of Standards and Technology

[Docket No. 900101-0219]

RIN 0693-AA59

Approval of Revisions to Federal Information Processing Standards (FIPS) Family of Input/Output Interface Standards

AGENCY: National Institute of Standards and Technology (NIST), Commerce.

ACTION: The purpose of this notice is to announce that the Secretary of Commerce has approved revisions to the Federal Information Processing Standards (FIPS) family of input/output interface standards, and has approved discontinuation of the exclusion and verification lists for these standards.

SUMMARY: On March 20, 1990, notice was published in the Federal Register (55 FR 10272) proposing revision of Federal Information Processing Standards (FIPS) 60-2, 61-1, 62, 63-1, 97, 111, 130, and 131 to make them non-mandatory, and discontinue the exclusion and verification lists for these standards. This proposal superseded the proposal for revision of these standards announced in the Federal Register (52 FR 44462) of November 19, 1987. Procedures for the Exclusion List for FIPS 60, 61, 62, 63, and 97 were published in the Federal Register on

September 3, 1982 (47 FR 38959–38960). Procedures for the Verification List for FIPS 60, 61, 62, 63, and 97 were published in the **Federal Register** on December 11, 1979 (44 FR 71444–71445) and on April 7, 1981 (46 FR 20719–20720).

The written comments submitted by interested parties and other material available to the Department relevant to these proposed revisions were reviewed by NIST. On the basis of this review, NIST recommended that the Secretary approve revisions to the input/output family of standards and approve discontinuation of the exclusion and verification lists for these standards. NIST prepared a detailed justification document for the Secretary's review in support of those recommendations.

This notice provides only the changes to the revised standards.

EFFECTIVE DATE: These revisions are effective December 18, 1990.

ADDRESSES: Interested parties may obtain copies of FIPS PUBS 60–2, 61–1, 62, 63–1, 97, 111, 130, and 131 from the National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161.

FOR FURTHER INFORMATION CONTACT: Ms. Shirley Radack, National Institute of Standards and Technology, Gaithersburg, MD 20899, telephone (301) 975–2833.

SUPPLEMENTARY INFORMATION: Under the provisions of 40 U.S.C. 759(d), the Secretary of Commerce is authorized to promulgate standards and guidelines for Federal computer systems, and to make such standards compulsory and binding to the extent to which the Secretary determines necessary to improve the efficiency of operation, or security and privacy of Federal computer systems.

The family of I/O interface standards currently includes:

a. FIPS 60–2, I/O Channel Interface, revised July 29, 1983.

b. FIPS 61–1, Channel Level Power Control Interface, revised July 13, 1982.

c. FIPS 62, Operational Specifications for Magnetic Tape Subsystems, revised December 30, 1980.

d. FIPS 63–1, Operational Specifications for Variable Block Rotating Mass Storage Subsystems, revised April 14, 1983; Supplement to FIPS PUB. 63–1, Additional Operational Specifications for Variable Block Rotating Mass Storage Subsystems, April 14, 1983.

e. FIPS 97, Operational Specifications for Fixed Block Rotating Mass Storage Subsystems, February 4, 1983.

f. FIPS 111, Storage Module Interfaces (with extensions for enhanced storage module interfaces), April 18, 1985.

g. FIPS 130, Intelligent Peripheral Interface (IPI), July 16, 1987.

h. FIPS 131, Small Computer System Interface (SCSI) July 16, 1987.

The following revisions are being made effective immediately upon publication. A delayed effective date is not required because these standards are exempt from the Administrative Procedure Act by U.S.C. 553(a)(2).

Revisions to Federal Information Processing Standards 60–2, 61–1, 62, 63–1, 97, 111, 130, and 131.

FIPS 60–2, I/O Channel Interface, is revised as follows:

Applicability. This standard addresses the interconnection of computer peripheral equipment as a part of ADP systems for the following types of peripherals: (1) Magnetic tape equipment employing open reel-to-reel magnetic tape storage devices, specifically excluding magnetic tape cassette and tape cartridge storage devices, (2) magnetic disk storage equipment employing disk drives each having a capacity greater than 7 megabytes per storage module, excluding flexible disk and disk cartridge devices having a smaller storage capacity per device, and (3) other peripheral equipment employing peripheral device types for which operational specifications standards have been issued as Federal Information Processing Standards. This standard is recommended for use in the acquisition of peripheral equipment for ADP systems with input/output channel interfaces as specified in the technical specifications, when it is determined that interchange of equipment between different systems is likely.

Implementation. The original version of this standard became effective December 13, 1979. The first revision became effective June 23, 1980, and the second revision became effective July 29, 1983. This revision becomes effective December 18, 1990.

Waivers. This standard is non-mandatory. No waivers are required.

FIPS 61–1, Channel Level Power Control Interface, is revised as follows:

Applicability. This standard addresses the power control interface in connecting computer peripheral equipment to ADP systems. It is recommended for use when FIPS 60–2 is used, when it is determined that interchange of equipment between different systems is likely.

Implementation. The original version of this standard became effective June 23, 1980, and the first revision became effective July 13, 1982. This revision becomes effective December 18, 1990.

Waivers. This standard is non-mandatory. No waivers are required.

FIPS 62, Operational Specifications for Magnetic Tape Subsystems, is revised as follows:

Applicability. This standard addresses magnetic tape equipment connected to ADP systems through FIPS 60 interfaces. It is recommended for use in the acquisition of such equipment, when it is determined that interchange of equipment between different systems is likely.

Implementation. The original version of this standard became effective June 23, 1980. This revision becomes effective December 18, 1990.

Waivers. This standard is non-mandatory. No waivers are required.

FIPS 63–1, Operational Specifications for Variable Block Rotating Mass Storage Subsystems, is revised as follows:

Applicability. This standard addresses peripheral device dependent operational interfaces for connecting variable block rotating mass storage equipment to ADP systems through FIPS 60 interfaces. It is recommended for use in the acquisition of such variable block rotating mass storage equipment for connection to ADP systems, when it is determined that interchange of equipment between different systems is likely.

Implementation. This standard became effective June 23, 1980, and the first revision became effective April 14, 1983. This revision becomes effective December 18, 1990.

Waivers. This standard is non-mandatory. No waivers are required.

FIPS 97, Operational Specifications for Fixed Block Rotating Mass Storage Subsystems, is revised as follows:

Applicability. This standard addresses the peripheral device dependent operational interface specifications for connecting fixed block rotating mass storage equipment to ADP systems through FIPS 60 interfaces. It is recommended for use in the acquisition of such fixed block rotating mass storage equipment for connection to ADP systems, when it is determined that interchange of equipment between different systems is likely.

Implementation. The original version of this standard became effective February 4, 1983. This revision becomes effective December 18, 1990.

Waivers. This standard is non-mandatory. No waivers are required.

FIPS 111, Storage Module Interfaces, is revised as follows:

Applicability. This standard addresses connection of a disk drive to a controller as part of an ADP system. This standard is recommended for use in the acquisition of disk systems that are

connected to small and medium sized computer systems, when it is determined that interchange of equipment between different systems is likely.

Implementation. This standard became effective May 18, 1985. This revision becomes effective December 18, 1990.

Waivers. This standard is non-mandatory. No waivers are required.

FIPS 130, Intelligent Peripheral Interface (IPI), is revised as follows:

Section 8, Applicability. This standard applies to the connection of computers to storage peripheral device controllers. This standard is recommended for use in the acquisition of magnetic disk drives, optical disk drives, and tape drives to be connected to minicomputer systems, when it is determined that interchange of equipment between different systems is likely.

Section 10, Implementation. This standard became effective December 16, 1987. This revision becomes effective December 18, 1990.

Section 11, Waivers. This standard is non-mandatory. No waivers are required.

FIPS 131, Small Computer System Interface (SCSI) is revised as follows:

Section 8, Applicability. This standard addresses the connection of small computers to peripheral devices with integral controllers. This standard is recommended for use in the acquisition of storage peripherals and small computer systems for office or laboratory use, when it is determined that interchange of equipment between different systems is likely.

Section 10, Implementation. This standard became effective December 16, 1987. This revision becomes effective December 18, 1990.

Section 11, Waivers. This standard is non-mandatory. No waivers are required.

Dated: December 12, 1990.

John W. Lyons,

Director.

[FR Doc. 90-29563 Filed 12-17-90; 8:45 am]

BILLING CODE 3510-CN-M