American National Standard

for information systems –

intelligent peripheral interface – device-specific command set for magnetic disk drives

Adopted for Use by the Federal Government

FIPS PUB 130
See Notice on Inside Front Cover

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

ANSI X3.130-1986

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 130, Intelligent Peripheral Interface (IPI). 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®
X3.130-1986

American National Standard
for Information Systems –

Intelligent Peripheral Interface – Device-Specific Command Set for Magnetic Disk Drives

Secretariat

Computer and Business Equipment Manufacturers Association

Approved May 20, 1986

American National Standards Institute, Inc

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 periodically to reaffirm, revise, or withdraw this standard. 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

Printed in the United States of America

A1M588/12

Foreword

(This Foreword is not part of American National Standard X3.130-1986.)

This standard provides a definition of the device specific portion of a family of standards called the Intelligent Peripheral Interface (IPI). It is a new high-performance, general-purpose parallel peripheral interface. This standard responds to an industry market need (expressed both by users and manufacturers) to limit the increasing costs in hosts associated with changes in peripherals. The intent of the IPI is to isolate the host (CPU), both hardware and software, from changes in peripherals by providing a "function-generic" command set to allow the connection of multiple types of peripherals (disks, printers, tapes, communications). To smooth the transition from the current methods to the generic approach, the IPI supports device-specific command sets, such as this one, to aid in bridging the gap between the two approaches.

To accomplish this set of goals, the design of the IPI includes device-specific and device-generic command sets, both utilizing a common physical bus. The device-specific command set provides:

(1) Device-oriented control
(2) Physical Data Addressing
(3) Timing Critical Operations
(4) Lower Device Cost

The device-generic command set provides a higher level of functionality and portability. It includes:

(1) Host/Device Independence
(2) Logical Data Addressing
(3) Timing Independence
(4) Command Queuing Capability

A system is not restricted to the use of one level of command set or the other. It is possible that both levels of command sets will be utilized with a given system's architecture to balance such parameters as system performance, cost, and peripheral availability. It is also possible for the host to provide for migration from device-specific to device-generic levels while still retaining the same physical interface.

The development of an Intelligent Peripheral Interface (IPI) was begun after a preliminary investigation had been completed. The earliest proposals were made by participants of Task Group X3T9.3 in late 1978. At that time, the Task Group decided generic-oriented peripheral interfaces were not yet ready for standardization and that the group should concentrate on device-oriented interfaces and the system-oriented, high-speed serial interfaces. The group acknowledged the desirability of higher level intelligent commands by reserving code fields in American National Standard for Information Systems — Interface between Rigid Disk Drive(s) and Host(s), ANSI X3.101-1984, during its April 1980 meeting.

The basic architecture of the resultant IPI was first proposed at the X3T9.3 August 1980 meeting. In addition to the 1978 proposal, complete company implementations were proposed by several manufacturers from August 1980 to August 1981. These proposals resulted from the initiative of the contributors and from wide-spread solicitation by the task group.

Task Group X3T9.3 agreed upon preliminary functional requirements during the October 1980 meeting, which included the following:

(1) Parallel transfer
(2) Command and Data Handshaking
(3) Allowance for high-speed transfers without Handshaking
(4) Transfer rate up to 10 Megaoctets per second

Task Group X3T9.3 began work on the IPI in 1981 in response to an emerging need for a higher performance peripheral interface. Coincidental with the need for higher performance was the availability of low-cost VLSI circuit technologies, allowing increased intelligence in the peripheral device. These needs were confirmed by large and active participation from all areas of the computer industry.

The fundamental characteristics that the group achieved included the following:
(1) Single or dual octet transfers
(2) Data rates of at least 10 megabytes per second
(3) Cable lengths extending from 5 to 125 meters depending upon type of transmitter and cable type
(4) Low-cost, commonly available components
(5) High level of maintainability and availability
(6) A multilevel command structure allowing different levels of intelligence in the peripherals
(7) A definition that facilitates evolutionary changes in the levels with minimal impact on software and hardware components
(8) Definitions supporting an extensive group of peripheral devices including disks, tape, communications equipment, printers, and the like, with a common choice of interface hardware and commands

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, D.C. 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 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 Computing Machinery	Kenneth Magel Jon A. Meads (Alt)
Association of the Institute for Certification of Computer Professionals	Thomas M. Kurihara

Organization Represented	*Name of Representative*
AT&T Communications	Henry L. Marchese
	Richard Gibson (Alt)
AT&T Information Systems	Herbert V. Bertine
	Paul D. Bartoli (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
Digital Equipment Computer Users Society	William Hancock
	Charles Mustain (Alt)
Digital Equipment Corporation	Gary S. Robinson
	Delbert L. Shoemaker (Alt)
Eastman Kodak	Charleton C. Bard
	Gary Haines (Alt)
General Electric Company	William R. Kruesi
	Richard W. Signor (Alt)
General Services Administration	William C. Rinehuls
	Larry L. Jackson (Alt)
GUIDE International	Frank Kirshenbaum
	Thomas F. O'Leary, Jr (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
	John S. Colonias (Alt)
Moore Business Forms	Delmer H. Oddy
National Bureau of Standards	Robert E. Rountree
	James Moulton (Alt)
National Communications System	Marshall L. Cain
	George W. White (Alt)
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	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	J. Wade Van Valkenburg
	R. C. Smith (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 International	Jean T. McKenna
	Susan Crawford (Alt)
Wang Laboratories, Inc	Marsha Hayek
	Joseph St. Armand (Alt)
Xerox Corporation	John L. Wheeler

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

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

Dennis Appelyard
James R. Barnette
Steve Cooper
Duane Barney
Louis C. Domshy
Robert Dugan
Patrick Lannan
John B. Lohmeyer
John McCool
Gene Milligan
Ted Petrowich
Reinhard Knerr
Gary S. Robinson
Arnold J. Roccati
Floyd E. Ross
Mark Hammang (Alt)
Kris Kowal (Alt)
George Clark (Alt)
Roger Cormier (Alt)
Charles Brill (Alt)
Sunil Joshi (Alt)
Dennis Krob (Alt)
John Hancock (Alt)
Robert B. Anthony (Alt)
Kirk Moulton (Alt)

Task Group X3T9.3 on Device Level Interfaces, which was responsible for the development of this standard, had the following participants:

Gary S. Robinson, Chair
I. Dal Allan, Vice-Chair

J. Amstutz
D. Appleyard
R. Barnes
D. Barney
R. Bender
R. Bergey
F. Berkowitz
B. Bonner
M. Bradac
C. Brill
B. Brown
R. Brown
W. Burr
E. Calkins
C. Chen
E. Cieniawa
S. Cooper
R. Davidelt
R. Davis
R. Derr
S. Dick
R. Dillon
R. Driscal
T. Eiland
D. Filpus
S. Finch
R. Fish
M. Fitzpatrick
M. Gamerl
R. Geller
S. Gersten
M. Glier
W. Grace
B. Graham
E. Grivna
D. Guss
K. Hallam
M. Hammang
D. Hartig
P. Hayden
C. Hess
C. Jarboe
S. Juhasz
D. Klang
K. Kong
A. Kononov
T. Leland
J. Lohmeyer
R. Lopez
J. Luttrull
R. Matheson
T. McClendon
D. McIntyre
P. Mclean
F. Meadows
J. Meyer
G. Milligan
P. Mizera
D. Moczarny
K. Moe
J. Monaco
R. Morris
J. Mulligan
R. Notari
T. O'Connor
M. O'Donnell
J. Patton
R. Peacock
J. Peterson
T. Petrowich
P. Phillips
M. Poehler
D. Ray
B. Reago
C. Ridgeway
D. Roberts
W. Roberts
A. Roccati
F. Ross
L. Russell
A. Salthouse
W. Sanderson
E. Sandoval
K. Scharf
D. Schneider
J. Schuessler
R. Schultz
D. Shoemaker
E. Slater
J. Smith
R. Snively
C. Stead
H. Stehle
M. Stewart
H. Truestedt
D. Tsai
N. Vashi
D. Voigt
C. Walker
O. Weeden
D. Williams
L. Zorza

Contents

SECTION PAGE

1. Scope and Editorial Conventions 9
   1.1 Scope 9
   1.2 Editorial Conventions 9

2. Referenced and Related American National Standards 9
   2.1 Referenced American National Standard 9
   2.2 Related American National Standards 9

3. General Description 9
   3.1 Application Environment 9
   3.2 Logical Interface Characteristics 9
   3.3 Responsibility 10

4. Glossary 10

5. Physical Interface Considerations 11
   5.1 Information Transfer 11
   5.2 Bus Octets 12
   5.3 ATTENTION IN Signal 14

6. Slave Functions 14
   6.1 Disk Format 14
   6.2 Format Specification 14
   6.3 Slave Conditions 18
   6.4 Dual Port 19
   6.5 Reset 20
   6.6 Head Control 20
   6.7 Rotational Position Sensing (RPS) 20
   6.8 Slave ECC 20
   6.9 Power Sequencing 21

7. Bus Controls 21
   7.1 Command/Response Controls 21
   7.2 Fixed Block Data Controls 28
   7.3 Variable Block Data Controls 30

8. Status 33
   8.1 Slave Ending Status 33
   8.2 Status Response 34
   8.3 Extended Status Response 36

Table 1 Slave Conditions 18

Appendixes

Appendix A Defect Map 38
   A1. Location 38
   A2. Contents 38
   A3. Formats 38
   A4. Protection 39

Appendix B Storage of Format Specifications and Attributes 40

American National Standard for Information Systems –

Intelligent Peripheral Interface – Device-Specific Command Set for Magnetic Disk Drives

1. Scope and Editorial Conventions

1.1 Scope. This document describes the Logical Level 2 (device level) Interface for disk drives. The physical, electrical, and configuration characteristics and the transmission protocol of this interface are in accordance with American National Standard for Information Systems – Intelligent Peripheral Interface – Physical Level, ANSI X3.129-1986. The interface is capable of handling data rates from 0 to at least 10 megaoctets per second, depending on driver and receiver classes.

The purpose of this standard is to facilitate the development and utilization of a device level interface which permits the interconnection of disk slave peripherals to a controller.

1.2 Editorial Conventions. Certain terms used in this standard that are proper names of signals are printed in uppercase to avoid possible confusion with other uses of the same words; e.g., ATTENTION IN. Any lowercase uses of these words have the normal English meaning.

A number of conditions, sequence parameters, events, English text, states or similar terms are printed with the first letter of each word in uppercase and the rest lowercase; e.g., In, Out, Selective Reset, Bidirectional, Bus Control, Operation Response. Any lowercase uses of these words have the normal English meaning.

2. Referenced and Related American National Standards

2.1 Referenced American National Standard. This standard is intended to be used in conjunction with American National Standard for Information Systems – Intelligent Peripheral Interface – Physical Level, ANSI X3.129-1986. When ANSI X3.129-1986 is superseded by a revision approved by the American National Standards Institute, Inc, the revision shall apply.

2.2 Related American National Standards. The following standards are not essential for the completion of the requirements of this standard and are intended to be used solely for explanation or clarification.

ANSI X3.132, Information Systems – Intelligent Peripheral Interface – Device-Generic Command Set for Magnetic and Optical Disk[1]

ANSI X3.147, Information Systems – Intelligent Peripheral Interface – Device-Generic Command Set for Magnetic Tapes[1]

3. General Description

3.1 Application Environment. The master (controller) provides control of one to eight slaves (disk drives) with a device level set of commands and data transfers. The master and slave of a level 2 environment are the slave and facility of a level 3 environment. There are no facilities in the level 2 environment.

3.2 Logical Interface Characteristics

3.2.1 Slave Control. Control of a slave is exerted by the master through the Bus Controls that initiate the Bus Exchanges for command,

---

[1] These standards are currently under development. Contact the secretatiat for current information about availability.

response, and data transfers. The Bus Controls incorporate coding that identifies the particular command, response, or data type. The Command Controls specify the command type being sent to the slave, whereas the Response Controls specify the response to be read from the slave. The Data Controls specify the operation on fields of a disk sector and their orientation with respect to the start of a sector.

3.2.2 Information Transfer. The interface uses the Double Octet mode to transfer 16-bit information. Data is transferred in the Data Streaming mode; Commands and Response are transferred in the Interlocked mode.

3.2.3 Data Transfers. The master reads and writes on the disk by issuing Data Controls while staying oriented with the rotation of the disk. No addresses or octet counts are involved with the Data Controls. Operations are on whole fields and include field read, write, and skip operations. Field sizes include any master-managed Error Correcting Codes (ECC) and Cyclic Redundancy Codes (CRC).

3.2.4 Format Control. The organization of sectors and the number of sectors per track is controlled by a Format Specification, which may be mutually derived by both the slave and the master. This specification dictates the number of fields per sector, the length of each field, and the number of sectors per track.

3.3 Responsibility

3.3.1 Master. The master performs the following functions:

(1) Data buffering
(2) Data management — Interleaving
(3) Header management
(4) Defect management — defect skipping, revectoring, spare sectors
(5) Error detection and/or correction — Retries, ECC, CRC
(6) Command generation

3.3.2 Slave. The slave performs the following functions:

(1) Sectoring — Hard, soft, or both; variable or fixed
(2) Formatting — Multiple fields, variable or fixed
(3) Gap management — PLO, pads, interfield gaps, write-read recovery
(4) Read Gate and Write Gate control
(5) PLO, bit, and octet synchronization and pattern writing
(6) Header verification for write (Format 1 only)
(7) Command execution
(8) Response generation
(9) Error detection and/or correction (optional)
(10) Dual port (optional)

4. Glossary

bus control. The physical-level octet placed on BUS A by the master during the Bus Control sequence. It is used to define the bus configuration and information type for the subsequent Information Transfer. The three Bus Control types are: Command Control, Response Control, and Data Control.

bus exchange. The Bus Control sequence (initiated by the master) and the Ending Status sequence (initiated by the slave), which are used to frame an Information Transfer.

busy. Busy indicates that the slave is unable to respond to the master's request and may be reported at either the Physical Interface (Selection Sequence) or at the Logical Interface (Ending Status Sequence).

command. The information transferred from the master to the slave following a Command Control. The information contains command-specific parameters.

command control. A Bus Control that specifies that command information is to be transferred from the master to the slave during the Information Transfer and identifies the command type.

data control. A Bus Control that specifies that data information is to be transferred during the Information Transfer, and specifies various data transfer requirements, including the orientation of the data on the disk.

defect map. A list of the defects recorded by the drive manufacturer on the slave.

format specification. An ordered list of parameters that specify the format of the tracks and sectors on the disk.

information transfer. The Physical Interface transfer of commands, responses, and data that are framed by a Bus Exchange.

level 2. A device-level-protocol interface in which the master is aware of the specific characteristics of the device under its control.

level 3. An intelligent interface oriented to the generic characteristics of devices, but not the characteristics unique to the device.

logical interface. All operations above the Physical Interface.

master status. The status placed on BUS A by the master during the Ending Status sequence after the Information Transfer.

parameter. The information octets transferred after a Command or Response Bus Control. Command parameters further define a command or contain specific information about a command, such as head number, cylinder number, and the like. Response parameters contain information on the state of the slave, such as status, current head, current cylinder, and the like.

physical interface. The mechanical, electrical, and bus protocol characteristics specified in ANSI X3.129-1986.

physical sector. This term refers to the sectors identified by the disk drive, relative to index, with zero as the first.

response. The information transferred from the slave to the master following a Response Control. The information contains parameters specific to the response type.

response control. A Bus Control that which specifies that information about a response type is to be transferred from the slave to the master during the Information Transfer and identifies the type of response.

slave status. The status placed on the BUS B by the slave during the Ending Status sequence after an Information Transfer.

status response. The status supplied to the master when executing a Read Status Response that contains the exception status bits.

target sector. The physical sector designated by the master for Rotational Position Sensing (RPS) and Data Controls that operate on target sectors.

5. Physical Interface Considerations

The Physical Interface is used as defined in ANSI X3.129-1986, with the interpretations and additions defined in this section.

5.1 Information Transfer

5.1.1 Octet Mode. All Information Transfers (Commands, Responses, and Data) are transmitted in the double octet (16-bit) mode. Even-numbered octets are transmitted on BUS A and odd-numbered octets are transmitted on BUS B; i.e., the octet on BUS A is considered as being transmitted before the octet on BUS B.

For Command and Response information, bit 7 of an octet is the most significant bit (MSB) and bit 0 is the least significant bit (LSB). The most significant octets are transferred first in multioctet numeric parameters.

For data information, there is no definition of significance to the ordering of bits or octets. The order in which the octets are read is the same as that in which they are written.

(Bit 7 on BUS A is the first recorded bit on the disk).

5.1.2 Transfer Mode. Command/Response information is always transferred in the Interlocked mode. Data information is always transferred in the Data Streaming mode. The transfer modes cannot be changed.

The Request Transfer Settings octet response reflects capabilities and settings of command/response transfers in Double Octet and Interlocked modes.

5.1.3 Termination of Transfers. The Information Transfer sequence may be terminated by either the slave or master.

5.1.4 Data Streaming. Data streaming enables high transfer rates over long cables and is well suited to data transfers by synchronous disk. This is accomplished by not interlocking the SYNC IN and SYNC OUT, which eliminates a round-trip transmission delay.

Data streaming shall be used for data transfers as defined in ANSI X3.129-1986, with the following interpretations:

(1) The SYNC IN pulses occur at the data rate of the disk and are generated at the rate of one per double octet.

(2) The nominal duty cycle of the SYNC IN is 50 percent.

(3) Upon recognizing a SYNC IN, the master returns a SYNC OUT after a delay not exceeding one double octet time at the maximum cable length.

During data output transfers, the slave shall prefetch data by initiating the SYNC IN pulses *n* octet times prior to the writing of the first data octet on the disk in order to account for transmission and master delays. The value of *n* is dependent on master, slave, and transmission delays and the data rate of the slave. The slave shall require at least an *n*-octet first in, first out (FIFO) buffer to account for delays less than maximum (*n* octet times). If data is not available in the slave when it must be written on the disk, a data underrun condition exists, the transfer is terminated and the Sucessful Information Transfer bit in the Slave Status octet may be set to zero.

Drives that support ECC/CRC and have that feature enabled shall complete the sector and write uncorrectable ECC/CRC if a data underrun condition occurs during a disk write.

5.2 Bus Octets. The bus octets defined in ANSI X3.129-1986 shall be used as described, with the additions described in this section. All valid state transitions shall be implemented by the slave.

5.2.1 Unsupported Bus Octets. The following bus octets are not supported:

(1) *Facility Selection.* Slave does not acknowledge selection.

(2) *Request Facility Interrupts.* Slave does not acknowledge request for facility interrupts.

(3) *Bus Acknowledge.* Slave transmits zeroes on BUS B.

5.2.2 Bus Control Octet. The Bus Control octets define the direction and type of information transfer. The Command Control encodes bits 5-0 to specify the command type. The Response Control encodes bits 5-0 to specify the expected response. The Data Control encodes bits 5-0 to specify the type of disk read/write operation.

The Command and Response Controls shall be as described in 7.1, and the Data Controls shall be as described in 7.2 and 7.3.

5.2.3 Master Status Octet. All bits are zero and not used, except for the Successful Information Transfer (bit 7) and the Bus Parity Error (bit 6). If the Master Status Octet indicates an unsuccessful information transfer following a Read Status Response, the slave shall not clear the Status response conditions to be reported.

5.2.4 Slave Status Octet. The Slave Status includes an Operation Ending Status in bits 3-0. These bits shall be as defined in 8.1.

5.2.5 Request Interrupts Octet. The three interrupt classes defined in ANSI X3.129-1986 are as follows:

(1) *Class 3 Status Pending Interrupt.* This interrupt is active when there is a status exception in the Status Response that must be reported to the master. It is reset when the status bits are cleared by a Read Status Response with a Successful Transfer bit set in Master Status, or the slave is reset with a Selective Reset Octet from either port with bit 2 (Slave Reset) set.

(2) *Class 2 RPS Interrupt.* This interrupt is active during the time on each disk revolution that the heads are over the target specified by the Load Position or Load RPS Target Sector Address commands. The RPS is disabled by setting X'FFFF' as the target sector during a seek initiated by the load Position command, or if the slave is reset by a Selective Reset on either port with the Reset Slave bit set. The interrupt is deactivated by the disabling of the RPS, or if the slave is reset with a Selective Reset Octet from either port with the Reset Slave bit set. The interrupt may also be deactivated by any accepted Data Control.

(3) *Class 1 Command Completion Interrupt.* This interrupt becomes active when a command completes successfully and the Time Dependent bit of the Slave Status following the transfer of the command was set. The interrupt is not generated for a Load Position command if the RPS is enabled. It is deactivated by any accepted Bus Control, or the slave is reset with a Selective Reset Octet from either port with bit 2 (Slave Reset) set.

5.2.6 Address Octet Response. The slave shall respond to a Request Interrupt Octet or Selection Octet with an Address Octet Response that has the following organization:

Bit	Description
7	Slave 7
6	Slave 6
5	Slave 5
4	Slave 4
3	Slave 3
2	Slave 2
1	Slave 1
0	Slave 0

5.2.7 Selective Reset Octet. The receipt of this octet causes a Selective Reset of the type

defined in bits 3-0. The following Reset Controls are defined:

Bit	Description
7	Indicates Selective Reset Octet
6-4	Slave address
3	Slave Release: The slave shall release its interface drivers in the same manner as it would upon recognition of the MAINT state. The drivers are to remain released on the port over which the Reset was received until recognition of another Reset with this bit cleared.
2	Reset Slave: The slave shall be reset in the same way as for power on.
1	Reset Logical Interface: The Logical Interface for the port shall be reset. The slave's common resources that are not switched to the alternate port shall be reset. The Format Specification is not affected.
0	Reset Physical Interface: The Physical Interface for the port shall be reset.

5.2.8 Selection Octet. The Selection Octet bits 6-4 specify the slave address. Bits 3-1 are not supported and shall be set to zero. If bit 0, Priority Select, is set, the slave shall cease any operations with the alternate port, cancel any reserve on the alternate port, and attempt to honor selection from the selecting port.

Bit	Description
7	0
6-4	Slave Address
3-1	0
0	Priority Select

5.2.9 Transfer Settings Octet. The slave shall respond to a Request Transfer Settings Octet from the master with a Transfer Settings Octet that has the following organization:

Bit	Description
7 = 0	Indicates Transfer Settings Octet
6 = X	Maintenance Mode 1 or 2
5 = 1	Current Setting is Double Octet Mode
4 = 0	Current Setting is Interlocked Transfer
3 = 0	Slave is not capable of transferring in Streaming
2 = 1	Slave is capable of transferring in Interlocked
1 = 1	Slave is capable of transferring in Double Octet Mode
0 = 0	Slave is not capable of transferring in Single Octet Mode

The transfer settings reported are those reflecting the capabilities and settings for the slave during Command/Responses Information Transfers.

5.2.10 Request Slave Interrupts Octet. The slave shall respond with a Slave Interrupts Octet when it receives a Request Slave Interrupts Octet. The organization of a Request Slave Interrupts Octet shall be as follows:

Bit	Description
7 = 1	Indicates Slave Interrupts Octet
6-4 =	Slave address
3 = 1	Indicates Slave Interrupts Octet
2-0 = 0	Indicates Slave Interrupts Octet

5.2.11 Slave Interrupts Octet. The slave shall respond with a Slave Interrupts Octet when it receives a Request Slave Interrupts Octet. The organization of the Slave Interrupts Octet shall be as follows:

Bit	Description
7	Reserved, must be zero.
6	Busy: Busy is an indication that the slave is unable to respond to the master's request; i.e., it is currently executing a Bus Control (previously issued from this port or the alternate port), or is reserved to the alternate port.
5	Ready: Ready is an indication that the slave is operational and able to accept all Data Bus Controls that are in context.
4	Reserved, must be zero.
3	Priority Select: This interrupt is active when a Priority Select has been received on the alternate port.

NOTE: This interrupt is cleared at the end of the Slave Interrupts sequence during which it was reported.

2	Status Pending Interrupt Active: This interrupt is active when there is active Exception Status to be reported in the Status Response block.

1 RPS Interrupt Active: This interrupt is active during the time on each disk revolution that RPS is enabled and the data heads are over the target.

0 Command Completion Interrupt: This interrupt is active when a command, previously transferred with the Time Dependent bit set in the Slave Status Octet, completes successfully. The Interrupt is not generated for a Load Position command and is cleared by the acceptance of any Bus Control.

5.3 ATTENTION IN Signal. The ATTENTION IN signal shall be activated by an unselected slave when the RPS, Command Completion, or Status Pending Interrupts are active, or a Busy or Not Busy transition has occurred (No Longer Busy), and the interrupt is enabled to generate an Attention. (See 5.2.5 for description of interrupts.) Command functions are provided for enabling and disabling the individual interrupts from generating an Attention, on a port basis. All Attention generation shall be enabled upon Reset except for No Longer Busy.

6. Slave Functions

6.1 Disk Format. There are three possible formats: Format 1 Sector Mode 1, Format 1 Sector Mode 2, and Format 2. At least one format shall be supported.

6.1.1 Format 1 — Fixed Block Format. The fixed block format allows for a slave to be formatted with a variable number of identical sectors per track. The length and organization of a fixed block sector shall be determined by the Fixed Block Format Specification. The fixed block format allows the slave to operate in one of the following two sector modes:

(1) *Sector Mode 1.* Every sector is of the same length and organization, which is fixed by the slave.

(2) *Sector Mode 2.* Every sector is of the same length and organization, but the length and organization is programmable.

The start of a fixed block sector may be determined by one of the methods described in 6.1.1.1 and 6.1.1.2.

6.1.1.1 Fixed Block Hard Sector. A slave in the hard-sectored mode typically derives the start of each sector by counting the number of octets from the index mark at the beginning of a track. The number of sectors is specified by the Format Specification.

6.1.1.2 Fixed Block Soft Sector. A slave in the soft-sectored mode uses Address Marks to mark the start of each sector. The number of sectors is specified by the Format Specification. Each track of the slave shall be formatted with Address Marks by the Perform Sector Marking function.

6.1.2 Format 2 — Variable Block Format. The variable block format allows for a slave to be formatted with a variable number of variable length sectors per track. Each sector may have a different organization.

A slave operating with a variable block format uses Address Marks to mark the start of each sector. The variable block format depends on the controller to determine the length and number of data fields in a sector during a disk write operation.

6.1.3 Format Control. The slave shall be responsible for format control. It shall be capable of performing the following functions under the direction of the Format Specification:

(1) Determining the start of each sector on the track

(2) Determining the start of each field within a sector

(3) Counting the number of octets in each field of a sector

(4) Activating Read Gate and Write Gate in accordance with the current Data Control and the Format Specification

(5) Aquiring PLO, bit, and octet synchronization on read operations

(6) Writing of PLO, bit, and octet synchronization patterns

6.2 Format Specification. The Format Specification is an ordered list of parameters that define for the slave how the tracks and sectors of the disk are to be organized. It is transferred to the slave by the Load Format Specification control, and the master may retrieve it by the Read Format Specification control. Both the master and the slave may establish the parameters during the initialization of the Format Specification.

6.2.1 Fixed Block Mode. The Fixed Block Mode is a mode of operation whereby certain slaves, using the Fixed Block Read/Write Data Bus Control set and Format Specification, allow the master to operate on disk data sectors of a fixed organization. This mode of operation is mutually exclusive with the Variable Block Mode described in 6.2.2 of this document.

6.2.1.1 Fixed Block Format Specification. The Fixed Block Format Specification allows for sectors of any number of fields, but all sectors must have the same organization. The Fixed Block Format Specification has the following format:

Octet	Bit	Description
0-1		Number of octets following: equals n - 1
2		Format type code: 1 = fixed block
3		Flag octet
	7	Initialized (Format Specification Initialization complete)
	6	Manufacturer's Default Format Specification
	5	Sector mode 2
	4	Sector mode 1
	3	Use soft sectoring
	2	Use hard sectoring
	1	Use field Data Controls
	0	Use sector Data Controls
4-5		Number of sectors per track
6-9		Number of physical octets per sector
A-B		Number of beginning header octets to be skipped
C-D		Number of fields per sector (m+1)
E-11		Number of octets per field 0 m=0
12-13		Master turnaround delay 0 m=0
(n-5)-(n-2)		Number of octets per field m
(n-1)-n		Master turnaround delay m

NOTE: m equals the field number of the last field in a sector. Fields are numbered starting from zero. n is the octet number of the last octet transfered.

6.2.1.1.1 Number of Octets. This double-octet parameter contains the octet count for the Format Specification. It does not include itself in the octet count.

6.2.1.1.2 Format Type Code. This single-octet parameter contains the format type code. If this parameter equals 1, then the Format Specification is a fixed block type and is organized as such.

6.2.1.1.3 Flag Octet. This single-octet parameter contains the flag information for the Fixed Block Format Specification and indicates if the Format Specification is initialized, if the Format Specification is a manufacturer's default format or user's format, which sector mode is to be used, and what type of data controls the master intends to use with this format.

6.2.1.1.4 Number of Sectors per Track. The number of sectors per track shall be determined by one of the following schemes:

(1) *Slave with a Fixed Sector.* If the slave has a fixed length sector size, the slave enters this number in the Format Specification every time it is read by the master.

(2) *Defined by the Master.* If the master is to define the number of sectors per track, the master sets the number in the Format Specification every time it is transmitted to the slave. The master specifies as X'FFFF'at least one of the remaining three quantities: the number of fields per sector, one or more field sizes, or one or more of the Master Turnaround Delays. The slave computes the value of the unspecified quantity.

(3) *Calculated by the Slave.* If the slave is to calculate the number of sectors that can be fitted on the track, the master may then supply it with the number of fields, the field sizes, and the Master Turnaround Delays. The master specifies the number of sectors as X'FFFF'. The slave then enters the number of sectors in the Format Specification.

In addition to the values that may be specified, the master has to take into consideration the number of slave-controlled overhead octets in the gaps.

6.2.1.1.5 Number of Physical Octets per Sector. The number of physical octets per sector is typically supplied by the slave and is used by the master to determine the sector number of a defect specified by the Defect Map. It may also be used to infer the size of the field gaps.

6.2.1.1.6 Number of Octets to be Skipped during Header Verify. This parameter shall be supplied by the master to specify the number of octets, starting from the beginning of the header, to be skipped in the slave's verification of the header during the execution of a Verify Header Data Control.

6.2.1.1.7 Number of Fields per Sector. The number of fields per sector may be supplied by the master or left to the slave to compute. The first field in a sector is usually the header identification.

6.2.1.1.8 Number of Octets per Field. The number of octets in each field should be specified by the master based on system considerations. Master-managed ECC and CRC are considered as part of the field length for the user. A field size shall be specified for each of the fields.

As viewed by the slave, the field length is the sum of the preamble used for PLO, bit, and octet synchronization; the data field of the user, and the interfield gap. The interfield gap shall be large enough to handle slave considerations, such as write splice, write pads, write-to-read recovery time, and the Master Turnaround Delays.

6.2.1.1.9 Master Turnaround Delay. The Master Turnaround Delay is the time, measured in octets, required by the master between fields to handle the Ending Status sequence from one Data Control and to initiate another. It is measured at the slave, and therefore includes transmission delays, from the point in time when the slave terminates a transfer (drops SLAVE IN) until the next Data Control is received at the slave. The actual gap used by the slave is the Master Turnaround Delay, increased to account for its own delays including: read propagation delay, transfer termination time, Slave Status turnaround time, and Data Control decoding.

The master shall provide this parameter for each of the fields. If no Data Control is to be sent to the slave after a field, the parameter for that field may be zero. This parameter and the Field Size parameter are repeated until every sector field is defined.

6.2.2 Variable Block Mode. The Variable Block Mode is a mode of operation whereby certain slaves, using an alternate Read/Write Data Bus Control set and Format Specification command, allow the master to write sectors with any number of data fields of any length, up to the track capacity of the slave. This mode of operation is mutually exclusive with the Fixed Block Mode described in 6.2.1 of this document.

6.2.2.1 Variable Block Format Specification. The Variable Block Mode uses the following Format Specification to define the length of fixed data segments, and gaps, and the distance between Sector Marks (RPS).

Octet	Bit	Description
0-1		Number of octets following: equals n - 1
2		Format type code: 2 = variable block
3		Flag octet
	7	Initialized (Format Specification Initialization complete)
	6	Manufacturer's Default Format Specification
	5-0	Reserved, must be zero
4-5		Cell length in octets
6-7		RPS divisions per track
8-9		Index Mark Gap length in cells
A-B		Home Data Segment length in cells
C-D		Home Gap length in cells
E		Home Field Skip length in cells
F		Home Field Retries
10-11		Header Data Segment length in cells
12-13		Header Gap length in cells
14-15		Data Gap length in cells
16-17		Defect Skip length in cells

6.2.2.1.1 Number of Octets. This double-octet parameter contains the octet count for the Format Specification. It does not include itself in the octet count.

6.2.2.1.2 Format Type Code. This double-octet parameter contains the format type code. If this parameter equals 2, then the Format Specification is a variable block type and is organized as such.

6.2.2.1.3 Flag Octet. This single-octet parameter contains the flag information for the Variable Block Format Specification and indicates if the Format Specification is initialized, and if the Format Specification is a manufacturer's default format or a user's format.

6.2.2.1.4 Cell Length in Octets. This double-octet parameter defines the length of a Cell in octets. Most formatting elements are defined in terms of Cells. The Cell length shall be longer in time than a Master Termination Delay (two times the Cable Delay plus the Master Turnaround Delay).

6.2.2.1.5 RPS Divisions per Track. This double-octet parameter defines the number of RPS divisions per track. In the Variable Block Mode, RPS divisions have no relationship with the location of the Data Fields and are only used to generate RPS interrupts and to report the current head position for certain Response Bus Controls.

6.2.2.1.6 Index Mark Gap Length in Cells. This double-octet parameter defines the length of the Gap following the Index Mark. This gap lies between the Index Mark and the beginning of the Home Field.

6.2.2.1.7 Home Data Segment Length in Cells. This double-octet parameter defines the length of the Home Field Data Segment. This field may be set to zero if the master wishes to have no Home Field.

6.2.2.1.8 Home Gap Length in Cells. This double-octet parameter defines the length of the

Home Field Gap Segment. This gap shall be large enough to allow for all re-command delays (e.g., master decision delay, cable delays) between the Home Data Segment and any Header or Data Field that follows. This gap shall also account for head-position uncertainty due to head scatter and system cable configuration signal delay difference. This field may be set to zero only if the Home Data Segment is set to zero.

6.2.2.1.9 Home Field Skip Length in Cells. This single-octet parameter defines the length in cells to be skipped, when a Home Field read fails, before another Home Field read is to be attempted by the slave. This field may be set to zero.

6.2.2.1.10 Home Field Retries. This single-octet parameter defines the number of automatic Home Field read retries that the slave is expected to perform when performing a Read Home Field operation. This field may be set to zero.

6.2.2.1.11 Header Data Segment Length in Cells. This double-octet parameter defines the length of the Header Field Data Segment.

6.2.2.1.12 Header Gap Length in Cells. This double-octet parameter defines the length of the Header Field Gap Segment. This gap shall be large enough to allow for all re-command delays (e.g., master decision delay, cable delays) between the Header Data Segment and any Header or Data Field that follows. This gap shall also account for head-position uncertainty due to head scatter and system cable configuration signal delay difference.

6.2.2.1.13 Data Gap Length in Cells. This double-octet parameter defines the length of the Data Field Gap Segment. This gap shall be large enough to allow for all re-command delays (e.g., master decision delay, cable delays) between the Data Field Data Segment and any Header or Data Field that follows. This gap shall also account for head-position uncertainty due to head scatter and system cable configuration signal delay difference.

6.2.2.1.14 Defect Skip Length in Cells. This double-octet parameter defines the length of the Defect Skip Segment. This segment is used in place of the larger Data Gap Segment when ending certain write operations to allow for skipping over small defects. This segment shall be large enough to allow for all re-command delays (e.g., master turnaround delay, cable delays) between the Data Field Data Segment and the following Data field. This segment shall also account for head-position uncertainty due to head scatter and system cable configuration signal delay difference.

6.2.3 Format Initialization. The Format Specification is not initialized when the slave is powered on, unless the slave has a means of saving it. The master may save the current Format Specifications and load them into the slave. (See Appendix B for description of storage of Format Specifications.)

A Format Specification with its Initialized bit reset is not complete and can not be used by the slave for format control or data transfer.

A Format Specification shall be initialized by the following procedure:

(1) The master loads a Format Specification with its Initialized bit reset to invalidate the old specification, if any, and to condition the slave to start the initialization process. The master provides any of the parameters that it is to contribute to the Format Specification. Parameters not specified by the master shall be X'FFFF'.

Alternatively, the master loads a completely specified Format Specification with the Initialized bit set.

If the slave determines that there are missing or incorrect parameters, it sets the appropriate Command Exception status bit.

(2) The master reads an initialized Format Specification from the slave to obtain the slave's fixed parameters, calculated parameters, or both. The master shall check the Format Specification against its requirements and adjust its own control to the specification.

6.2.4 Manufacturer's Format Specification. The slave shall have a built-in Manufacturer's Format Specification, which is used primarily to write and read the Defect Map. It shall be invoked by the Load Format Specification control with Bit 6 of the Flag Octet set, and shall remain in effect until another Load Format Specification control is received by the slave. It may be read by a Read Format Specification control when it is invoked.

The format shall provide for a header and one or two data fields, each field not to exceed 1024 octets in length. The header contents, the CRC, and the ECC are not specified. The format shall be organized to allow for either sector type or field type of controls. There shall be no sector interleaving.

The master shall be capable of recovering the data from data fields formatted with this

Table 1
Slave Conditions

Power	Ready	Busy	
0	0	0	Nothing happening at the slave (Not P-Available)
0 0	0 1	1 x	Illegal Conditions
1	0	0	The slave is Operational but Not Ready. It is able to handle Selection, Deselection, Bus Control and Ending Status sequences. All Read/Write Data Controls and any Command/Response Bus Controls that require the drive HDA to be accessed will be rejected with a Slave Ending Status of Operation Exception, Bus Control Out of Context.
1	1	0	The slave is Operational and Ready. It is able to handle Selection, Deselection, Bus Control and Ending Status sequences, and Information Transfers in context.
1	0	1	The slave is Operational, Not Ready, and Busy.
1	1	1	The slave is Operational, Ready, and Busy.

specification, without the use of the headers. Sectors shall be read in their physical order from Index.

The Manufacturer's Format Specification may be a Fixed Block or Variable Block Format Specification as indicated by the Format Type Code.

Slaves that operate only in the Variable Block Format shall be capable of reporting the Manufacturer's Default Format in a Fixed Block Format so that the master may determine the number of sectors per track, fields per sector, and octets per field.

6.3 Slave Conditions. The manner in which a slave responds to a command shall be determined by its condition.

The condition of a slave is affected by its intrinsic as well as its operational characteristics. The conditions are:

(1) *P-Available.* The port is capable of responding on the Physical Interface.

(2) *Not P-Available.* The port does not respond on the Physical Interface.

(3) *Operational.* The slave is capable of responding on the Physical Interface and executing commands.

(4) *Ready.* The slave is able to execute any command.

(5) *Not Ready.* The slave is not able to execute any command that requires access to the drive HDA.

(6) *Status Pending.* The slave has a Status Response for the master. The Status Interrupt is active.

(7) *Active.* The slave has accepted a command and has not yet generated the corresponding Slave Status with the Time Dependent bit reset nor activated the Command Completion Interrupt.

(8) *Inactive.* The slave has no command outstanding, but is capable of receiving one from the master.

(9) *Reset.* The slave is in an initialized condition when it has no cognizance of past events. This condition can come about as a result of an external reset by the master, an internal initialization (e.g., power-on), or an unsuccessful internal recovery attempt from a severe slave error.

Slaves provide predictable status on their ability to process commands. This information is obtained via the Request Interrupts sequence (see Table 1).

An Operational slave is able to respond to Master Reset, Selective Reset, Request Interrupts, Request Slave Interrupts, Request Transfer Settings and any Bus Controls that are in context.

A Busy slave may or may not be able to handle Selection, Deselection, Bus Control, and Ending Status sequences.

(1) A physical level dual port switch will indicate Busy during Selection.

(2) A logical level dual port switch will honor the Selection, Deselection, and Bus Control sequences, and will indicate Busy in the Slave Status octet of the Ending Status sequence.

6.4 Dual Port (Optional). The optional dual port consists of manual and programmed Enable/Disable controls for a port (static switching) and the logical constructs for dynamically switching of the slave between two ports. The switching of a slave to a port creates an allegiance of the slave's common resources (slave dependent) to that port.

6.4.1 Port Enable/Disable. Ports may be individually enabled or disabled by command and by optional manual controls. When a port is disabled, it is made Not P-Available on the Physical Interface. The disabling of a port may be either orderly or destructive with the choice being implementation dependent. Both ports are enabled on power-on, if not disabled by manual means.

The orderly disabling of a port takes effect when the Physical Interface on that port is in the IDLE state and the slave, if switched to the port, is inactive. The destructive disabling of a port takes place immediately without regard to the state of the interface.

The enabling of a port takes effect when the Physical Interface on that port is in the IDLE state.

If a port is disabled by command, changing a manual port switch from Disable to Enable shall cause the port to be Enabled. However, the disabling of a port by a manual switch cannot be overridden by a command.

The disabling of a port, either by command or switch, shall cause any explicit or implicit reservation to be cleared and shall cause any solicited status associated with the disabled port to be reset.

6.4.2 Slave Accessibility. The slave can appear in one of two accessibility modes: Neutral and Switched.

Following a Reset condition, the slave enters the Neutral mode. While in Neutral mode, the slave can be accessed via either enabled port and can become switched to an enabled port, either implicitly or explicitly.

(1) *Implicit Switching.* A slave becomes switched to a port implicitly under the following two conditions:

(a) Communication (Selection or Bus Exchange, as determined by the slave) over a port is initiated in accordance with the Physical Interface protocol described in ANSI X3.129-1986. The slave returns to the Neutral mode when the communication ends, unless returned to Neutral by a Reset.

(b) The slave accepts a command. The slave returns to Neutral when the command is completed, unless returned to Neutral by a Reset.

(2) *Explicit Switching.* Explicit allegiance occurs when a Reserve or Priority Reserve function is issued to the port. The slave becomes switched to the port over which the Reserve is received. It remains switched until a Release function is received, the alternate port initiates a Priority Reserve or Priority Select, or the slave is reset with a Selective Reset Octet from either port with bit 2 (Slave Reset) set.

The reserve mechanization is slave dependent. The alternate port is made Busy at either the Physical Interface or the Logical Interface.

An Attention function shall be generated whenever an attempt to access a slave is rejected because it is switched to another port, if enabled by function code 1E (Enable "No Longer Busy Attention"), when the slave goes Not Busy. The Attention will be canceled when the master attempts selection on that port.

The switch mechanism may be implemented within a slave in the following ways:

(1) A Physical Interface Only switch does not permit a reservation (Priority Hold is not supported).

(2) A Logical Interface Only switch can be implemented only if a slave provides the ability to process commands on both ports concurrently.

(3) A combination of both is also possible. In such a case, a Priority Select at the Physical Interface shall override any logical reservation.

6.4.3 Notification of Alternate Port Exception Condition. The occurrence of the following conditions on the alternate port shall be

reported as an Unsolicted Exception in the Status Response:

(1) Priority Select
(2) Format Specification changed
(3) Reset complete
(4) Format complete

6.4.4 Attention. When the slave is in the Neutral mode, Attentions may be sent by the slave over a port, if it has an interrupt condition for that port. When the slave has an allegiance to a port, Attention may be sent to the port to which the slave has the allegiance, provided that the interrupt condition is for that port, but not selected by that port. The generation of Attention by the interrupts may be enabled and disabled on a port basis by command.

6.5 Reset

6.5.1 External Reset. An external reset, in the form of a Master Reset or a Selective Reset received over the Physical Interface, can be presented by the master at any time regardless of the condition of the slave. It shall affect the port over which it is received and the slave, if not Switched to the alternate port. (See also Selective Reset (5.2.7).

The Master Reset and the Reset Physical Interface type of Selective Reset reset the Physical Interface of the port over which they were received and do not affect any port switch, any explicit allegiance, pending interrupts, or the Format Specification.

The Reset Logical Interface type of Selective Reset resets all of the Logical Interfaces of the port over which it was received, including any pending interrupts and any implicit allegiance to this port. It does not affect the state of any port switch or the Format Specification.

The Reset Slave type of Selective Reset resets the entire slave, including pending interrupts, any allegiances, any port switch, any strobes and offsets, and the Format Specification. Following the Reset, the slave enters the Operational condition, except when precluded from doing so by conditions that force it into a Not Operational condition.

6.5.2 Internal Reset. When a slave internal reset occurs, an Unsolicited Exception Status bit is set and the Status Interrupt is activated and sent to all ports that are enabled.

6.6 Head Control. The slave's head addressing register is loaded by the Load Head Address and Load Position controls. The head address register shall be capable of being incremented at the end of a successful data transfer, if bit 4 of a Fixed Block Data Control is set. A successful transfer requires that the Successful Information Transfer bit be set in both the Master Status and the Slave Status.

The address of the first head shall be zero and the address of the last head shall be one less than the number of heads specified in the head Configuration response. The counter increments to zero after the last head address.

The head address shall advance immediately upon receiving the Master Status octet, so as to minimize the head switching gap time.

6.7 Rotational Position Sensing (RPS) (Optional). The optional RPS provides the ability for the slave, when on the addressed cylinder, to generate class 2 RPS Interrupts and, if enabled, an Attention to the master during the time that the heads are over the Fixed Block Mode target sector or Variable Block Mode target RPS division.

6.7.1 Target Sector Address. The target sector is established by transferring a valid physical sector number to the slave by a Load RPS Target Sector or Load Position control. An address of X'FFFF' disables the RPS. The target sector is initialized to X'FFFF' on power-on.

6.7.2 RPS Interrupt. The RPS Interrupt is active for one sector (Fixed Block Mode) or RPS division (Variable Block Mode) time when the sector or division is under the heads. The Interrupt may be extended if the slave supports variable RPS pulse width (see Load Position and Load RPS Target Address Commands). It is deactivated when the slave receives a Data Control or when the RPS is disabled. If the RPS is enabled at the time the slave completes a seek, the normal Command Completion Interrupt is not generated, and the slave remains active until the first RPS Interrupt. If there is status pending, the RPS Interrupt is not generated.

6.8 Slave ECC (Optional). The slave may optionally perform its own ECC on any or all fields of the sector. A Ending Slave Status code is provided for notifying the master of a slave-detected data error. The Read Correction Vectors control is provided to allow the master to obtain the correction vectors from the slave. The ECC function can be disabled and enabled by Load Slave Function codes.

6.9 Power Sequencing (Optional). Power Sequencing provides a means for the master to sequentially start the motors of the slaves, so as to reduce the power line surge current while the slaves are reaching operating speed. The master shall issue a Spin Up function to each slave as it becomes operational. The Command Completion interrupt for this function shall not occur until the slave is up to speed.

This Slave Function is optional for the slave, or in some cases may be bypassed by a switch, if power sequencing is not required or provided by some other means. If power sequencing is supported by the slave and is enabled, the slave shall not turn on its motor when power is applied.

7. Bus Controls

The Bus Controls specify the condition of the bus and the information to be transferred. The three types of Bus Controls are: Command, Response, and Data.

7.1 Command/Response Controls. The Command/Response Controls are forms of the Bus Control octet that allow commands to be transmitted to the slave and responses to be read from the slave. The command and response types are identified in the Bus Control coding. The command and response information transferred consists of ordered lists of parameters. Parameters are transmitted with the most significant octet first.

Commands shall not be stacked in the slave.

The valid Command/Response Controls and their hexadecimal codes are listed in the following table and are described in 7.1.1 through 7.1.15. All other Command/Response Control codes (including 00 and 40 used in IPI Level 3) are rejected as being unsupported.

Bus Control Code	Command/Response Control
01	Load Slave Function
02	Load Format Specification
03	Load Slave Specific Information
04	Load Cylinder Address
05	Load Head Address
06	Load RPS Target Sector Address
07	Load Position
41	Read Configuration
42	Read Format Specification
43	Read Slave Specific Information
44	Read Status
45	Read Correction Vectors
46	Read Current Sector Address
47	Read Position
48	Read Extended Status

7.1.1 Load Slave Function (01). The Load Slave Function control causes the slave to perform the function specified in the function code. The function code is carried in both octets of the two octets of the parameter transmitted to the slave. If the two octets are not the same or the octets contain an invalid value, the Command shall be rejected as containing an invalid parameter. The format of the command shall be as follows:

Octet Parameters	Description
0	Function code
1	Function code repeated

The various functions and their hexadecimal codes shall be as described in 7.1.1.1 through 7.1.1.45.

7.1.1.1 Reserved for Future Use (00-0F)

7.1.1.2 Disable Alternate Port (10) (optional). This function causes the slave to disable the alternate port, making it Unavailable at the Physical Interface. This function may cause an immediate and hard disabling of the alternate port. The function causes an exception if there is only one port.

7.1.1.3 Enable Alternate Port (11) (optional). This function causes the slave to enable the alternate port, thereby making it Available at the Physical Interface. The function causes an exception if there is only one port.

7.1.1.4 Disable Port (12) (optional). This function causes the slave to disable the port over which it received this function, making it Unavailable. This function takes effect when the master deselects from this port.

7.1.1.5 Priority Reserve (13) (optional). This function causes the slave to be reserved to this port, even if already reserved by the other port. The reserve remains in effect until released, until a Priority Reserve is executed by the alternate port, until a selection octet with the Priority Select bit set is issued by the alternate port, or until the slave is reset

with a Selective Reset Octet from either port with bit 2 (Slave Reset) set.

7.1.1.6 Reserve (14) (optional). This function causes the slave to be reserved to this port. The reserve remains in effect until released, until a Priority Reserve is executed by the alternate port, until a selection octet with the Priority Select bit set is issued by the alternate port, or until the slave is reset with a Selective Reset Octet from either port with bit 2 (Slave Reset) set.

7.1.1.7 Release (15) (optional). This function causes the slave to be released from this port and to enter the Neutral mode.

7.1.1.8 Notify Alternate Port of Format Completion (16) (optional). This function sets the Format Completed bit in the Status Response for the alternate port immediately. The function causes an exception if there is only one port.

7.1.1.9 Reserved for Future Use (17)

7.1.1.10 Disable Successful Command Completion Attention (18). The function causes the slave to disable the Command Completion Interrupt from generating an Attention on this port.

7.1.1.11 Enable Successful Command Completion Attention (19). This function causes the slave to enable the Command Completion Interrupt to generate an Attention on this port. It is enabled after a power up reset or a Selective Reset with bit 2 (Slave Reset) of the Selective Reset Octet set.

7.1.1.12 Disable RPS (Class 2) Interrupt Attention (1A). The function causes the slave to disable the RPS Interrupt from generating an Attention on this port.

7.1.1.13 Enable RPS (Class 2) Interrupt Attention (1B). This function causes the slave to enable the RPS Interrupt to generate an Attention on this port. It is enabled after a power up reset or a Selective Reset with bit 2 (Slave Reset) of the Selective Reset Octet set.

7.1.1.14 Disable Status Pending (Class 3) Interrupt Attention (1C). The function causes the slave to disable the Status Pending Interrupt from generating an Attention on this port.

7.1.1.15 Enable Status Pending (Class 3) Interrupt Attention (1D). This function causes the slave to enable the Status Pending Interrupt to generate an Attention on this port. It is enabled after a Power Up Reset or a Selective Reset with bit 2 (Slave Reset) of the Selective Reset Octet set.

7.1.1.16 Disable "No Longer Busy" Attention (1E) (optional). This function causes the slave to disable the slave's "No Longer Busy" Attention. It is disabled after a power up reset or a Selective Reset with bit 2 (Slave Reset) of the Selective Reset octet set.

7.1.1.17 Enable "No Longer Busy" Attention (1F) (optional). This function causes the slave to enable the slave's "No Longer Busy" Attention.

7.1.1.18 No Operation (20). This function performs no operation.

7.1.1.19 Reserved for Future Use (21)

7.1.1.20 Spin Up (22) (optional). This function causes the slave to turn on its motor. If it cannot perform this operation, an exception status is set in the Status Response. The Command Completion interrupt is generated when the slave is up to speed (or nearly so). If the slave motor is on, the Time Dependent bit of the Slave Status need not be used. Resets do not affect the power state of the slave motor.

7.1.1.21 Spin Down (23) (optional). This function causes the slave to turn off its motor. If it cannot perform this operation, an exception status is set in the Status Response. Resets do not affect the power state of the slave motor.

7.1.1.22 Load Heads (24) (optional). This function causes the slave to load its heads. If it cannot perform this operation, an exception status is set in the Status Response. Resets do not affect the load state of the slave heads.

7.1.1.23 Unload Heads (25) (optional). This function causes the slave to unload its heads. If it cannot perform this operation, an exception status is set in the Status Response. Resets do not affect the load state of the slave heads.

7.1.1.24 Lock Carriage (26) (optional). This function causes the slave to lock the carriage. If it cannot perform this operation, an exception status is set in the Status Response. Resets do not affect the lock state of the slave heads.

7.1.1.25 Unlock Carriage (27) (optional). This function causes the slave to unlock the carriage. If it cannot perform this operation, an exception status is set in the Status Response. Resets do not affect the lock state of the slave heads.

7.1.1.26 Recalibrate (28). This function shall cause the slave to move the heads to cylinder 0.

7.1.1.27 Execute Internal Diagnostic (29). This function causes the slave to execute its built-in diagnostics.

7.1.1.28 Reserved for Future Use (2A)

7.1.1.29 Perform Sector Marking (2B) (optional). This function causes the slave to format the currently selected track with Address Marks in accordance with the Format Specification. If the slave cannot perform this function or if the current Format Specification does not indicate that soft sectoring is to be used, an exception status is set in the Status Response.

7.1.1.30 Disable Slave ECC (2C) (optional). This function causes the slave to disable the slave ECC mechanism. If the slave cannot perform this operation, an exception status is set in the Status Response.

7.1.1.31 Enable Slave ECC (2D) (optional). This function causes the slave to enable the slave ECC mechanism. If the slave cannot perform this operation, an exception status is set in the Status Response.

7.1.1.32 Reserved for Future Use (2E-40)

7.1.1.33 Reset Offset (41). This function clears any head offset. If a slave does not support track offsets, it shall respond with an immediate Command Complete Ending Status Octet upon receipt of this function code.

7.1.1.34 Set Positive Offset 1 (42). This function causes the slave to offset its heads by one step (slave dependent) in the positive direction (away from the spindle). If a slave does not support track offsets, it shall respond with an immediate Command Complete Ending Status Octet upon receipt of this function code.

7.1.1.35 Set Negative Offset 1 (43). This function causes the slave to offset its heads by one step in the negative direction (towards the spindle). If a slave does not support track offsets, it shall respond with an immediate Command Complete Ending Status Octet upon receipt of this function code.

7.1.1.36 Set Positive Offset 2 (44). This function causes the slave to offset its heads by two steps in the positive direction. If a slave does not support track offsets, it shall respond with an immediate Command Complete Ending Status Octet upon receipt of this function code.

7.1.1.37 Set Negative Offset 2 (45). This function causes the slave to offset its heads by two steps in the negative direction. If a slave does not support track offsets, it shall respond with an immediate Command Complete Ending Status Octet upon receipt of this function code.

7.1.1.38 Set Positive Offset 3 (46). This function causes the slave to offset its heads by three steps in the positive direction. If a slave does not support track offsets, it shall respond with an immediate Command Complete Ending Status Octet upon receipt of this function code.

7.1.1.39 Set Negative Offset 3 (47). This function causes the slave to offset its heads by three steps in the negative direction. If a slave does not support track offsets, it shall respond with an immediate Command Complete Ending Status Octet upon receipt of this function code.

7.1.1.40 Set Normal Strobe (48). This function causes the slave to set normal data strobe. If a slave does not support early/late data recovery strobes, it shall respond with an immediate Command Complete Ending Status Octet upon receipt of this function code.

7.1.1.41 Set Early Strobe (49). This function causes the slave to set an early data strobe. If a slave does not support early/late data recovery strobes, it shall respond with an immediate Command Complete Ending Status Octet upon receipt of this function code.

7.1.1.42 Set Late Strobe (4A). This function causes the slave to set a late data strobe. If a slave does not support early/late data recovery strobes, it shall respond with an immediate Command Complete Ending Status Octet upon receipt of this function code.

7.1.1.43 Reserved for Future Use (4B-7F)

7.1.1.44 Perform Slave Specific Function *n* (8*n*) (optional). This function causes slave-specific function *n* to be performed. If it can not be performed, the function is rejected or causes an exception.

7.1.1.45 Reserved for Future Use (90-FF)

7.1.2 Load Format Specification (02). The Load Format Specification control transmits a Format Specification to the slave. The Format Specification is described in 6.2. The format of the command parameters shall be as follows:

Octet	Parameters
0-1	Number of octets following: equals *n* - 1
2-3	Format Type and Flag Octet
4-*n*	Remainder of Format Specification (optional)

If bit 6 of the Flag Octet is set, the Manufacturer's Format Specification is invoked and no more parameters need be supplied by the master.

7.1.3 Load Slave-Specific Information (03). The Load Slave-Specific Information transmits slave-specific information (e.g., diagnostics) to the slave. The format of the command parameters shall be as follows:

Octet	Parameters
0-1	Number of octets following: equals *n* - 1
2-*n*	Slave-specific information

7.1.4 Load Cylinder Address (04). The Load Cylinder Address control causes the slave to seek to the cylinder specified in the four octets transmitted to the slave. Any head or strobe offset is cleared. The format of the command parameters shall be as follows:

Octet	Parameter
0-3	Cylinder address

7.1.5 Load Head Address (05). The Load Head Address control causes the slave to select the head specified in the two octets transmitted to the slave. Any head or strobe offset is cleared. The format of the command parameters shall be as follows:

Octet	Parameter
0-1	Head address

7.1.6 Load RPS Target Sector Address (06). The Load RPS Target Sector Address control causes the slave to select the physical sector (Fixed Block Mode) or RPS division (Variable Block Mode) specified in the two octets of RPS target address transmitted to the slave for the target.

The master has several options with this control and may do one of the following:

(1) Send a double-octet parameter to extend the RPS pulse width. For the Fixed Block Mode, this parameter specifies the number of sectors that the RPS Interrupt is extended. For the Variable Block Mode, this parameter specifies the number of RPS Divisions that the RPS Interrupt is extended. The RPS pulse is extended by this parameter in the direction ahead of the target. If this parameter contains a value of zero, then the RPS Interrupt defaults to a length of one sector or RPS division. If the slave does not support the extended RPS pulse width, it shall terminate the transfer after the receipt of the first parameter field.

NOTE: When this field is used, it is a multiplier, i.e., of the number of sectors during which the interrupt is presented.

(2) Send a four-octet field to define the RPS pulse width in octets. If the slave does not support setting of the RPS pulse width in octets, it shall terminate the transfer after the receipt of the second parameter field.

NOTE: When this field is used, the RPS Pulse Width Extension is a multiplier of the value.

(3) Send a four-octet field to define a Class 2 interrupt skew factor ahead of the trailing edge of fixed block sectors in Format 1. This skew factor should typically correspond to the master's estimated reconnect time in octets. In the case of large sectors, this allows the master to set the target address immediately before the desired sector and to define the number of octets prior to its trailing edge where the slave is to negate the Class 2 Interrupt for RPS. If the slave does not support skewing the Class 2 Interrupt, it shall terminate the transfer after the receipt of the third parameter field.

NOTE: When this field is used the master has to define RPS Pulse Width in octets.

The RPS function shall be disabled if the sector address is X'FFFF'. The command completes with the Time Dependent bit of the Slave Status reset.

The format of the command parameters shall be as follows:

Octet	Parameter
0-1	RPS target address
2-3	RPS pulse width extension (optional)
4-7	RPS Pulse Width (Format 1 option)
8-B	RPS Pulse Width Skew (Format 1 option)

If the RPS is not supported, an exception is generated when this command is received by a slave.

If an optional field is not supplied by the master, the value shall remain at its previous setting.

7.1.7 Load Position (07). The Load Position control causes the slave to seek to the specified cylinder and select the specified

head. Any active head offset or data strobe is cleared.

The master has several options with this control and may do one of the following:

(1) Send a double-octet parameter to set the RPS target address. For the Fixed Block Mode, this parameter specifies the RPS target sector. For the Variable Block Mode, this parameter specifies the RPS target divisions. If the RPS target address is set to X'FFFF', RPS Interrupts are disabled. If the slave does not support RPS Interrupts, it shall terminate the transfer after the receipt of the second (Head Address) parameter field.

(2) Send a double-octet parameter to extend the RPS pulse width (see 7.1.6). If the slave does not support the extended RPS pulse width, it shall terminate the transfer after the receipt of the third parameter field.

(3) Send a four-octet field to define finer granularity of the RPS pulse width (see 7.1.6). If the slave does not support the variable RPS pulse width, it shall terminate the transfer after the receipt of the fourth parameter field.

(4) Send a four-octet field to define a Class 2 Interrupt skew factor ahead of the trailing edge of fixed block sectors in Format 1 (see 7.1.6). If the slave does not support skewing the Class 2 Interrupt, it shall terminate the transfer after the receipt of the fifth parameter field.

The command completes with the Time Dependent bit of the Slave Status set. RPS Interrupts, if enabled, are suspended until the slave comes on cylinder. After the slave comes on cylinder, RPS Interrupts are re-enabled if a valid non-X'FFFF' RPS target parameter was received.

The format of the command parameters shall be as follows:

Octet	Parameters
0-3	Cylinder address
4-5	Head address
6-7	RPS target address (optional)
8-9	RPS pulse width extension (optional)
A-D	RPS Pulse Width (Format 1 option)
E-11	RPS Pulse Width Skew (Format 1 option)

If an optional field is not supplied by the master, the value shall remain at its previous setting.

7.1.8 Read Configuration (41). The Read Configuration control causes the slave to transfer configuration information to the master. The format for this transfer shall be as follows:

Octet	Bit	Parameters
0-1		Number of octets following: equals *n* - 1
2		Device Class Code 1 = disk
3		Slave type flag octet
	7	Non-removable disk
	6	Removable disk
	5	Reserved
	4	Fixed head disk
	3	Moving head disk
	2	Reserved
	1	Reserved
	0	Reserved
4		Capability flag octet
	7	Reserved
	6	Variable Block Mode
	5	Fixed Block Mode, Sector Mode 2
	4	Fixed Block Mode, Sector Mode 1
	3	Soft sector
	2	Hard sector
	1	Field Data Controls
	0	Sector Data Controls
5		Feature flag octet
	7	RPS
	6	Dual port
	5	Slave ECC
	4	Reserved
	3	Slave responds to adjacent odd-even select addresses for dual actuators
	2	Slave responds to adjacent odd-even select addresses for dual devices per actuator
	1	Slave restores last loaded Format Specification
	0	Fixed Format Specification
6-9		Address of last data cylinder
A-D		Address of last defect list cylinder
E-F		Number of heads per cylinder
10-11		Number of fixed sectors per track
12-15		Number of octets per track
16-19		Single cylinder seek time (microseconds)
1A-1D		Average seek time (microseconds)
1E-21		Maximum seek time (microseconds)

22-25	Nominal rotation time (microseconds)
26-29	Head switching time (microseconds)
2A-2D	Write-to-read recovery time (microseconds)
2E-31	Manufacturer identification (ASCII)
32-39	Manufacturer model number (ASCII)
3A-3D	Manufacturer revision number (ASCII)
3E-45	Manufacturer unique unit ID number
46-47	Manufacturer switch settings defined field
48-*n*	Manufacturer defined field

7.1.8.1 Number of Octets. This double-octet parameter contains the octet count for the Configuration Response. It does not include itself in the octet count.

7.1.8.2 Device Class Code. This double-octet parameter contains the device class code. This standard defines only a single code, which is for magnetic disk drives.

7.1.8.3 Slave Type Flag Code. This single-octet parameter contains the slave type flag. This parameter indicates whether the slave contains removable media, nonremovable media, or both; and has fixed heads, moving heads, or both.

7.1.8.4 Capability Flag Octet. This single-octet parameter contains the slave capability flag. This parameter indicates whether the slave can operate in the Variable Block Mode, the Fixed Block Mode, or in both modes. If the slave functions in the Fixed Block Mode, then this parameter indicates whether the slave supports sector mode 1, sector mode 2, or both; soft-sector marks, hard-sector marks, or both; and Field Data Controls, Sector Data Controls, or both.

7.1.8.5 Feature Flag Octet. This single-octet parameter contains the slave feature flag. This parameter indicates if the slave supports RPS, Dual Port, Slave ECC, and adjacent odd-even select addresses for dual actuators.

7.1.8.6 Address of Last Data Cylinder. This four-octet parameter contains the address of the last data cylinder on this slave. The first cylinder has an address of zero.

7.1.8.7 Address of Last Defect List Cylinder. This four-octet parameter contains the address of the last defect list data cylinder on this slave.

7.1.8.8 Number of Heads per Cylinder. This double-octet parameter contains the number of heads on this slave.

7.1.8.9 Number of Fixed Sectors per Track. This four-octet parameter contains the number of fixed sectors per track on this slave. If the slave is not fixed sectored, this parameter shall contain X'FFFF'.

7.1.8.10 Number of Octets per Track. This four-octet parameter contains the guaranteed number of physical octets per track on this slave.

7.1.8.11 Single-Cylinder Seek Time. This four-octet parameter contains the value of the single-cylinder seek time in microseconds for this slave.

7.1.8.12 Average Seek Time. This four-octet parameter contains the value of the average seek time in microseconds for this slave. Average seek time equals the sum of the times to perform all possible seeks divided by the number of all possible seeks.

7.1.8.13 Maximum Seek Time. This four-octet parameter contains the value of the maximum cylinder seek time in microseconds for this slave.

7.1.8.14 Rotational Time. This four-octet parameter contains the value of the maximum rotational latency in microseconds for this slave.

7.1.8.15 Head Switching Time. This four-octet parameter contains the value of the maximum head switching time in microseconds for this slave.

7.1.8.16 Write-to-Read Recovery Time. This four-octet parameter contains the value of the maximum write-to-read recovery time in microseconds for this slave.

7.1.8.17 Manufacturer Identification. This four-octet parameter contains the slave manufacturer's Identification in ASCII.

7.1.8.18 Manufacturer Model Number. This eight-octet parameter contains the slave model number in ASCII.

7.1.8.19 Manufacturer Revision Number. This four-octet parameter contains the slave revision number in ASCII.

7.1.8.20 Manufacturer Unit ID Number. This eight-octet parameter contains the unit identification number for the slave. The ID number for each slave, which is provided by the manufacturer, shall either be different from the numbers for any other slave of the same manufacturer, model, and revision number, or contain the value X'FFFF FFFF FFFF FFFF', if not unique.

7.1.8.21 Manufacturer Switch Settings. This double-octet parameter contains the value of any switch settings the slave reports via the Configuration Response.

7.1.9 Read Format Specification (42). The Read Format Specification control causes the slave to transfer the current Format Specification to the master (see 6.2 for a description of the Format Specification). The response shall have the following format:

Octet	Parameters
0-1	Number of octets following: equals *n* - 1
2-3	Format Type and Flag Octet
4-*n*	Remainder of Format Specification

7.1.10 Read Slave Specific Information (43). The Read Slave Specific Information control causes the slave to transfer slave-specific information (e.g., diagnostics) to the master. The response shall have the following format:

Octet	Parameters
0-1	Number of octets following: equals *n* - 1
2-*n*	Slave-specific information

7.1.11 Read Status (44). The Read Status control causes the slave to transfer up to eight octets of status to the master. The response shall have the following format:

Octet	Parameters
0	Exception Status
1	Unsolicited Exception Status
2	Bus Control Exception Status
3-4	Slave Exception Status
5-7	Vendor Unique Status

See 8.2 for detailed definition of the Status Response. The slave may terminate the transfer at any point if the remaining octets are zeros.

Reading the status causes it to be cleared if the Master Status sent to the slave indicates a successful transfer.

7.1.12 Read Correction Vectors (45) (optional). The Read Correction vectors control causes the slave to transfer the ECC correction vectors to the master. If the optional slave ECC is not implemented, this control shall be rejected as unsupported. The format of the response shall be as follows:

Octet	Parameters
0-1	Number of octets following: equals *n* - 1
2	Error pattern
3-5	Error octet location from start of last field transferred
6-*n*	Octets 2-5 repeated as needed for additional vectors

The error pattern octet is exclusive-ORed with the data octet at the specified location. If an error is in the ECC octets, no vectors are returned.

7.1.13 Read Current Sector Address (46). The Read Current Sector Address control causes the slave to transfer the current physical sector address to the master. The response shall have the following format:

Octet	Parameter
0-1	Current sector address

If the slave cannot determine the current sector address, it returns X'FFFF'. The address received by the master may be one sector behind the actual sector because of the time required to transfer the response.

7.1.14 Read Current Position (47). The Read Position control causes the slave to transfer the current position to the master. The format of the response shall be as follows:

Octet	Parameters
0-3	Current cylinder address
4-5	Current head address
6-7	Current RPS target sector address
8-9	Current sector address

If the RPS function is not enabled or not supported, the current RPS target sector address is X'FFFF'. If the slave cannot determine the current sector address, the current sector address is X'FFFF'. The current sector address received by the master may be one sector behind the actual sector because of the time required to transfer the response.

7.1.15 Read Extended Status (48). The Read Extended Status control causes the slave to

transfer up to eight octets of status to the master. The format of the response shall be as follows:

Octet	Parameters
0	Interface Flags
1	Data Recovery Flags
2	Slave Control Flags
3	Slave Status
4	Slave Alarms
5-7	Vendor Unique Status

See 8.3 for detailed definition of the Extended Status Response. The slave may terminate the transfer at any point if the remaining octets are zeros.

7.2 Fixed Block Data Controls. Fixed Block Data Controls enable the users to read and write on the disk in the Fixed Block Format. They specify the direction of the transfer, the fields to be involved, the orientation of the fields, and a Step Head control.

7.2.1 Definitions and Use

7.2.1.1 Fixed Block Data Control Types. There are two types of Fixed Block Data Controls: Field Data Controls and Sector Data Controls. Field Data Controls specify an operation on a single field or a pair of fields. Sector Data Controls are composite controls that specify operations on sectors having a header and data fields as required. When using Field Data Controls, the previous field must have been operated on by a Field or Sector Data Control.

7.2.1.2 Head Advance Control. When bit 4 is set in any Data Control, the head counter advances at the end of a successful transfer. It is also advanced unconditionally by the Step Head control.

7.2.1.3 Orientation with the Disk. It is essential that the master's issuing of Data Controls stays oriented with the disk and that no sectors or fields, which are intended to be operated on, are missed. A Data Control that is not received in time for the slave to act on the next sector or field following the previous Data Control causes orientation to be lost and is rejected. To prevent the first Data Control of a series from being rejected when there is no orientation, orientation shall be established by starting with a target-type Sector Data Control or issuing a read or verify header-type Sector Data Control.

A Field Data Control that operates on the header field shall set the sector orientation for that sector.

Orientation is not lost by a header verify miscompare or a master-initiated termination of a transfer.

7.2.1.4 Target Sector. A group of read and write Data Controls operate on the target sector. The target is the physical sector set by a Load RPS Target Sector Address or Load Position command, if the RPS function is supported. Otherwise, the target is sector 0.

7.2.1.5 Data Control Reject. The Data Control may be rejected for several reasons, including:

(1) The Format Specification is not initialized

(2) The slave is active on the alternate port

(3) The Data Control is a write type and the slave is write protected

(4) The orientation is lost

(5) The Data Control type is invalid

7.2.1.6 Fixed Block Data Control Coding. The Data Control, which is a form of the Bus Control, has the following coding:

Bit	Description
7	1 = Data Control, 0 = Command/Response Control
6	1 = Information In (Read), 0 = Information Out (Write)
5	0 = Fixed Block Data Control
4	Step Head Control
3	Header Field Operation
2	Modifier Bit Selects Target Sector for Read/Write Header operations Selects Verify Header for write operation Selects Skip Header for read operation
1	Data Field 2 operation
0	Data Field 1 operation

This coding results in eight groupings of Data Controls plus a special control as shown in the following table:

Data Control		Description
80-83	91-93	Skip/Write Data Field
84-87	94-97	Verify Header, Write Data
88-8B	98-9B	Write Header, Write Data
8C-8F	9C-9F	Write Header, Write Data At Target

C0-C3	D1-D3	Skip/Read Data Field
C4-C7	D4-D7	Skip Header, Read Data
C8-CB	D8-DB	Read Header, Read Data
CC-CF	DC-DF	Read Header, Read Data At Target

90	Step Head
D0	reserved

7.2.2 Sector Data Controls. The sector Data Controls operate on sectors consisting of a Header Field, and one or more optional Data Fields. These Data Controls shall be as described in 7.2.2.1 through 7.2.2.24. The first of the two hexadecimal codes is for no head step; the second is for head step.

7.2.2.1 Verify Header (84, 94). The Verify Header Data Control causes the slave to verify the next Header Field against data transferred from the master. If the header verifies, the slave shall indicate a Successful Information Transfer in the Slave Ending Status; otherwise, the Verify Header Miscompare status shall be sent.

7.2.2.2 Verify Header and Write Data Field 1 (85, 95). The Verify Header and Write Data Field 1 Data Control causes the slave to verify the next Header Field against data transferred from the master. If the header verifies, the slave continues to write Data Field 1 with data transferred from the master. If the header does not verify, the slave shall terminate the transfer after the header and indicate Verify Header Miscompare in the Slave Ending status.

7.2.2.3 Verify Header and Write Data Field 2 (86, 96). The Verify Header and Write Data Field 2 Data Control causes the slave to verify the next Header Field against data transferred from the master. If the header verifies, the slave continues to write Data Field 2 with data transferred from the master. If the header does not verify, the slave shall terminate the transfer after the header and indicate Verify Header Miscompare in the Slave Ending status.

7.2.2.4 Verify Header and Write Data Fields 1 and 2 (87, 97). The Verify Header and Write Data Fields 1 and 2 Data Control causes the slave to verify the next Header Field against data transferred from the master. If the header verifies, the slave continues to write Data Fields 1 and 2 with data transferred from the master. If the header does not verify, the slave shall terminate the transfer after the header and indicate Verify Header Miscompare in the Slave Ending status.

7.2.2.5 Write Header (88, 98). The Write Header Data Control causes the slave to write the next Header Field with data transferred from the master.

7.2.2.6 Write Header and Data Field 1 (89, 99). The Write Header and Data Field 1 Data Control causes the slave to write the next Header Field and Data Field 1 with data transferred from the master.

7.2.2.7 Write Header and Data Field 2 (8A, 9A). The Write Header and Data Field 2 Data Control causes the slave to write the next Header Field and Data Field 2 with data transferred from the master.

7.2.2.8 Write Header and Data Fields 1 and 2 (8B, 9B). The Write Header and Data Fields 1 and 2 Data Control causes the slave to write the next Header Field and Data Fields 1 and 2 with data transferred from the master.

7.2.2.9 Write Header At Target (8C, 9C). The Write Header At Target Data Control causes the slave to write the Header Field at the target sector with data transferred from the master.

7.2.2.10 Write Header and Data Field 1 At Target (8D, 9D). The Write Header and Data Field 1 At Target Data Control causes the slave to write the Header Field and Data Field 1 at the target sector with data transferred from the master.

7.2.2.11 Write Header and Data Field 2 At Target (8E, 9E). The Write Header and Data Field 2 At Target Data Control causes the slave to write the Header Field and Data Field 2 at the target sector with data transferred from the master.

7.2.2.12 Write Header and Data Fields 1 and 2 At Target (8F, 9F). The Write Header and Data Fields 1 and 2 At Target Data Control causes the slave to write the Header Field and Data Fields 1 and 2 at the target sector with data transferred from the master.

7.2.2.13 Skip Header (C4, D4). The Skip Header Data Control causes the slave to skip the next Header Field. No data is transferred and the slave initiates the Ending Status sequence at the end of the Header Field.

7.2.2.14 Skip Header and Read Data Field 1 (C5, D5). The Skip Header and Read Data Field 1 Data Control causes the slave to skip the next Header Field and transfer Data Field 1 to the master.

7.2.2.15 Skip Header and Read Data Field 2 (C6, D6). The Skip Header and Read Data Field 2 Data Control causes the slave to skip the next Header Field and transfer Data Field 2 to the master.

7.2.2.16 Skip Header and Read Data Fields 1 and 2 (C7, D7). The Skip Header and Read Data Fields 1 and 2 Data Control causes the slave to skip the next Header Field and transfer Data Fields 1 and 2 to the master.

7.2.2.17 Read Header (C8, D8). The Read Header Data Control causes the slave to transfer the next Header Field to the master.

7.2.2.18 Read Header and Data Field 1 (C9, D9). The Read Header and Data Field 1 Data Control causes the slave to transfer the next Header Field and Data Field 1 to the master.

7.2.2.19 Read Header and Data Field 2 (CA, DA). The Read Header and Data Field 2 Data Control causes the slave to transfer the next Header Field and Data Field 2 to the master.

7.2.2.20 Read Header and Data Fields 1 and 2 (CB, DB). The Read Header and Data Fields 1 and 2 Data Control causes the slave to transfer the next Header Field and Data Fields 1 and 2 to the master.

7.2.2.21 Read Header At Target (CC, DC). The Read Header At Target Data Control causes the slave to transfer the Header Field at the target sector to the master.

7.2.2.22 Read Header and Data Field 1 At Target (CD, DD). The Read Header and Data Field 1 At Target Data Control causes the slave to transfer the Header Field and Data Field 1 at the target sector to the master.

7.2.2.23 Read Header and Data Field 2 At Target (CE, DE). The Read Header and Data Field 2 At Target Data Control causes the slave to transfer the Header Field and Data Field 2 at the target sector to the master.

7.2.2.24 Read Header and Data Fields 1 and 2 At Target (CF, DF). The Read Header and Data Field 1 and 2 At Target Data Control causes the slave to transfer the Header Field and Data Fields 1 and 2 at the target sector to the master.

7.2.3 Field Data Controls. The Field Data Controls described in 7.2.3.1 through 7.2.3.8 operate on a single field or a pair of fields. If one of these Data Controls is used for the first field (header) of a sector, it causes orientation to be lost and is rejected. The Sector Data Controls, which operate only on the header, are a form of Field Data Controls and shall be used for the first field of a sector.

The Data Controls shall be as described in the 7.2.3.1 through 7.2.3.8. The first hexadecimal code is for no head step; the second is for head step.

7.2.3.1 Skip Field (80). The Skip Field Data Control causes the slave to skip the next field. No data is transferred.

7.2.3.2 Skip Two Fields (C0). The Skip Two Fields Data Control causes the slave to skip the next two fields. No data is transferred.

7.2.3.3 Write Field (81, 91). The Write Field Data Control causes the slave to write the next field with data transferred from the master.

7.2.3.4 Skip Field and Write Field (82, 92). The Skip Field and Write Field Data Control causes the slave to skip the next field and then write the following field with data transferred from the master.

7.2.3.5 Write Two Fields (83, 93). The Write Two Fields Data Control causes the slave to write the next two fields with data transferred from the master.

7.2.3.6 Read Field (C1, D1). The Read Field Data Control causes the slave to transfer the next field to the master.

7.2.3.7 Skip Field and Read Field (C2, D2). The Skip Field and Read Field Data Control causes the slave to skip the next field and then transfer the following field to the master.

7.2.3.8 Read Two Fields (C3, D3). The Read Two Fields Data Control causes the slave to transfer the next two fields to the master.

7.2.4 Special Data Controls - Step Head (90). The Step Head Data Control causes the slave to step to the next head address. No data is transferred. This Data Control may be combined with the others to step the head address at the end of a successful transfer.

7.3 Variable Block Data Controls. The Variable Block Data Controls provide for reading and writing data sectors with Data Fields of variable length on the disk. They specify the direction of the data transfer, the field type involved, and the orientation requirements of the field.

7.3.1 Variable Block Data Control Coding. A Variable Block Data Control, which is a form of the Bus Control, has the following coding:

Bit	Description
7	1 = Data Control, 0 = Command/Response Control
6	1 = Information In (Read), 0 = Information Out (Write)
5	1 = Variable Block Control
4-0	= Read/Write Variable Block Data Operation Code

The variable block operations shall be defined as follows:

Data Control	Description
A0-A3	Format Field type
A4-A7	Format Field type
A8-AB	Write Field type
AC	Skip Field type
AD-AF	reserved
90	Step Head (See 7.2.4)
B0-BF	reserved
E0-E3	Read Field Type
E4-E7	reserved
E8-EB	reserved
EC-EF	reserved
F0-FF	reserved

7.3.2 Format Home Field (A0). This Data Control causes the slave to write the Home Field following Index Mark and to write a sync pattern until index is encountered a second time. This Data Control does not require orientation and can be issued at any time by the master. The slave locates the Home Field by reference to the Format Specification, which defines the distance in cells between the beginning of the Home Field and the disk Index Mark.

The slave shall terminate the data transfer when the end of the Home Field data segment is encountered, as defined in the Format Specification. The slave shall then begin writing a Home Gap. If another format-type Data Control is received before the Home Gap is completely written, it shall be executed at the end of the Home Gap. Otherwise, a sync pattern shall be written until Index Mark is detected.

7.3.3 Format Header Field (A1). This Data Control causes the slave to format a Header Field. This Data Control requires orientation and can only be issued during the gap following a Home Field, Header Field, Skip Field or another Data Field. When the end of the current field gap is reached, the slave writes an Address Mark, Sync Pattern, and Header data segment.

The slave shall terminate the data transfer when the end of the Header Field Data Segment is reached, as defined in the Format Specification. The slave shall then begin writing a Header Gap. If another format-type Control is received before the Header Gap is completely written, it shall be executed at the end of the Header Gap. Otherwise, a sync pattern shall be written until Index Mark is detected.

7.3.4 Format Data Field (A2). This Data Control causes the slave to write the next Data Field followed by a Data Field Gap Segment. This Data Control requires orientation and can only be issued during the gap following a Home Field, Header Field, Skip Field, or another Data Field. The slave locates the Data Field by assuming it immediately follows the current field.

The end of the operation occurs when the master terminates the data transfer. The slave shall then begin writing a Data Field Gap. If another format-type Control is received before the Data Field Gap is completely written, it shall be executed at the end of the Data Field Gap. Otherwise, a sync pattern shall be written until Index Mark is detected.

7.3.5 Format Data Field — Short Skip (A3). This Data Control causes the slave to write the next Data Field followed by a Data Field Skip Segment. This Data Control requires orientation and can only be issued during the gap following a Home Field, Header Field, Skip Field, or another Data Field. The slave locates the Data Field by assuming it immediately follows the current field.

The end of the operation occurs when the master terminates the data transfer. The slave shall then begin writing a Data Field Skip. If another format-type Data Control is received before the Data Field Skip is completely written, it shall be executed. Otherwise, a sync pattern shall be written until Index Mark is detected.

7.3.6 Format Skip from Index (A4). This Data Control causes the slave to write a sync pattern from index. The length of the sync gap in cells is specified by the double-octet parameter passed to the slave. This Data Control does not require orientation and can be issued at any time by the master.

The slave shall terminate the data transfer after receiving the gap length parameter. The slave shall begin writing an Index Gap after detecting the Index Mark. If another format-type Data Control is received before the Index Gap is completely written, it shall be executed at the end of the Index Gap. Otherwise, a sync pattern shall be written until Index Mark is detected again.

7.3.7 Format Skip (A5). This Data Control causes the slave to write a sync pattern from the end of the gap of the previous format-type Data Control. The length of the sync gap in cells is specified by the double-octet parameter passed to the slave. This Data Control requires orientation and can only be issued during the gap following a Home Field, Header Field, Skip Field or Data Field.

The slave shall terminate the data transfer after receiving the gap length parameter. The slave shall begin writing a Format Gap after detecting the end of the current field gap. If another format-type Data Control is received before the Format Gap is completely written, it shall be executed at the end of the Format Gap. Otherwise, a sync pattern shall be written until Index Mark is detected again.

7.3.8 Write Home Field (A8). The Write Home Field Data Control causes the slave to write the Home Field following index. This Data Control does not require orientation and can be issued at any time by the Master. The slave locates the Home Field by reference to the Format Specification, which defines the distance, in cells between the beginning of the Home Field and disk Index Mark. The end of the operation occurs when the slave terminates the data transfer.

The slave shall terminate the data transfer when the end of the Home Field data segment is encountered, as defined in the Format Specification. If a Data Field follows the Home Field and the master intends to operate on it, the appropriate Field control shall be issued before the end of the Home Field Gap is reached.

7.3.9 Write Header Field (A9). The Write Header Field Data Control causes the slave to write in the next Header Field encountered. This Data Control does not require orientation and can be issued at any time by the master. The slave locates the Header Field by searching for the first occurrence of an Address Mark after receipt of this Data Control. The end of the operation occurs when the slave terminates the data transfer.

The slave shall terminate the data transfer when the end of the Header Field data segment is encountered, as defined in the Format Specification. If any Data Fields follow the Header Field and the master intends to operate on them, the appropriate Field control shall be issued before the end of the Header Field Gap is reached.

7.3.10 Write Data Field (AA). The Write Data Field control causes the slave to write the next Data Field. This Data Control requires orientation and can only be issued during the gap following a Home Field, Header Field, Skip Field, or another Data Field. The slave locates the Data Field by assuming it immediately follows the current field. The end of the operation occurs when the master terminates the data transfer. After Master Termination the slave shall continue to maintain orientation until the end of the Data Field Gap Segment.

7.3.11 Write Data Field - Short Skip (AB). This Data Control causes the slave to write the next Data Field. This Data Control requires orientation and can only be issued during the gap following a Home Field, Header Field, or another Data Field. The slave locates the Data Field by assuming it immediately follows the current field.

7.3.12 Defect Skip (AC). This Data Control causes the slave to skip from the end of the last field gap, by the number of cells specified by the double-octet parameter passed to the slave as write data. This Data Control requires orientation and can only be issued during the gap following a Home Field, Header Field, or a Data Field.

7.3.13 Read Home Field (E0). The Read Home Field Data Control causes the slave to read the Home Field following index. This Data Control does not require orientation and can be issued at any time by the master. The slave locates the Home Field by reference to the Format Specification, which defines the distance in cells between the beginning of the Home Field and the Disk Index Mark. The end of the operation occurs when either the master or the slave terminates the data transfer.

The master may terminate the data transfer operation at any time, and the slave shall continue to maintain orientation until the end of field is reached, as defined in the Format Specification.

The slave shall terminate the data transfer when the end of the Home Field data segment is

encountered, as defined in the Format Specification. If a Header or Data Field follows the Home Field and the master intends to operate on it, the appropriate Field Control shall be issued before the end of the Home Field Gap is reached.

7.3.14 Read Header Field (E1). The Read Header Field Data Control causes the slave to read the next Header Field encountered. This Data Control does not require orientation and can be issued at any time by the master. The slave locates the Header Field by searching for the first occurrence of a Header Address Mark after receipt of this Data Control. The end of the operation occurs when either the master or the slave terminates the data transfer.

The master may terminate the data transfer operation at any time, and the slave shall continue to maintain orientation until the end of the field is reached, as defined in the Format Specification.

The slave shall terminate the data transfer when the end of the Header Field data segment is encountered, as defined in the Format Specification. If any Data Fields follow the Header Field and the master intends to operate on them, the appropriate Field control shall be issued before the end of the Header Field Gap is encountered.

7.3.15 Read Data Field (E2). The Read Data Field Data Control causes the slave to read the next Data Field. This Data Control requires orientation and can only be issued during the gap following a Home Field, Header Field, or another Data Field. The slave locates the Data Field by assuming it immediately follows the current field. The end of the operation occurs when the master terminates the data transfer.

The master may terminate the data transfer operation at any time before the end of the field is reached. However, if any Data Fields follow this one and the master intends to operate on them, termination of the transfer shall occur at the end of the Data Fields, since the slave uses Master Termination to determine the end of the Data Segment and the beginning of the Field Gap. After Master Termination, the slave shall continue to maintain orientation until the end of Field Gap is reached, as defined in the Format Specification.

7.3.16 Read Data Field — Short Skip (E3). The Read Data Field Data Control causes the slave to read the next data field. This Data Control requires orientation and can only be issued during the gap following a Home Field, Header Field, or another Data Field. The slave locates the Data Field by assuming it immediately follows the current field. The end of the operation occurs when the master terminates the data transfer.

The master may terminate the data transfer operation at any time before the end of the field is reached. However, if any Data Fields follow this one and the master intends to operate on them, termination of the transfer shall occur at the end of the Data Fields, since the slave uses Master Termination to determine the end of the Data Segment and the beginning of the Short Skip Field Gap. After Master Termination, the slave shall continue to maintain orientation until the end of the Short Skip Field Gap is reached, as defined in the Format Specification.

8. Status

8.1 Slave Ending Status. The Slave Ending Status is presented to the master during the Ending Status sequence following an Information Transfer and is defined as follows:

Bit	Description
7	Successful Information Transfer
6	Bus Parity Error
5	Odd Byte Transfer
4	Time-Dependent Operation
3-0	Operation Ending Status
	00 xx = Normal end
	00 = Slave available, Bus Control executed
	01 = Slave Busy, Bus Control rejected
	1x = reserved
	01 xx = Data Exceptions
	00 = missed AM
	01 = missed Sync Byte
	10 = ECC error (optional)
	11 = Verify Header Miscompare
	10 00 = Operation Exceptions
	01 = reserved
	1x = reserved
	11 00 = Unsolicited Exceptions
	01 = reserved
	1x = reserved

8.1.1 Successful Information Transfer. This bit is set if the slave determines that the Information Transfer was successful.

8.1.2 Bus Parity Error. This bit is set if there is a parity error on the Bus Control or any information received by the slave.

8.1.3 Odd Byte Transfer. This bit is set if the data transfer was of an odd byte length. The byte on Bus "B" of the last word transferred should be ignored when this bit is true.

8.1.4 Time-Dependent Operation. This bit is set if the command has not been completed by the slave at the time this status is sent to the master. A Command Completion Interrupt or an RPS Interrupt and an Attention, if enabled, will be generated when the command is complete. If this bit is reset, the command is now complete and there will be no Command Completion Interrupt.

8.1.5 Operation Ending Status. This four-bit field indicates if the preceding Bus Control was rejected because of a slave busy condition or outstanding Unsolicited exception, or if the preceding Bus Control incurred a Data or Operation exception. A slave busy condition can result because the slave is reserved to the alternate port or because the slave is currently executing a command that previously returned an Ending Status Octet with the TDO bit set. There are three types of categories of exceptions reported in the Ending Status Octet: Data Exceptions, Operation Exceptions, and Unsolicited Exceptions.

8.1.5.1 Data Exceptions. These exceptions occur when there is a failure to record data sucessfully.

8.1.5.1.1 Missed AM. This exception shows a failure to detect Address Mark.

8.1.5.1.2 Missed Sync Bit. This exception shows a failure to detect Sync Byte.

8.1.5.1.3 ECC Error (Optional). This exception shows that an ECC error was detected.

8.1.5.1.4 Verify Head Miscompare. This exception occurs when there is a mismatch between header and master supplied data.

8.1.5.2 Unsolicited Exception/Operation Exception. An Unsolicited Exception indicates that the previous Bus Control was rejected because outstanding Unsolicited Exception Status exists and the master shall read this status by issuing a Read Status Response Command before any other Bus Controls will be accepted.

An Operation Exception indicates that the previous Bus Control incurred an operation error, and that the nature of this error may be determined by issuing a Read Status Response Command. This captured status may be ignored. If the available status is ignored, it will be clear upon receipt of the next Bus Control.

8.2 Status Response. The status bits of the Status Response indicate exception conditions. They are set on the occurrence of an exception event. The setting of any exception status bit activates the Status Pending interrupt and Attention line if enabled.

The slave transfers the Status Response to the master upon receiving a Read Status Bus Control and clears all exception status bits if the Master Status Octet indicates a successful transfer. The exception status bits are also cleared by a Reset Logical Interface. All exception status bits, except the Unsolicited Exception bits (octet 0, bit 6 and octet 1, bits 7 through 0) are also cleared upon the acceptance of any Bus Control.

The Status Response shall have the following format:

Octet	Bit	Description
0		Exception Status Octet
	7	Status Response
	6	Unsolicited Exception
	5	Bus Control Exception
	4	Read Fault
	3	Write Fault
	2	Seek Fault
	1	Spindle Fault
	0	Execution Fault
1		Unsolicited Exception Status
	7	Reset Complete
	6	Alternate Port Priority Select
	5	Alternate Port Format Change
	4	Alternate Port Format Complete
	3	reserved
	2	Not Ready Transition
	1	Ready Transition
	0	Media Change
2		Bus Control Exception Status
	7	Invalid Bus Control
	6	Invalid Parameter
	5	Unsupported Bus Control
	4	Bus Control Context
	3	Data Bus Control Late
	2	reserved
	1	reserved
	0	reserved

3		Slave Exception Status
	7	Speed Fault
	6	Off Cylinder Fault
	5	Head Select Fault
	4	reserved
	3	reserved
	2	Voltage Fault
	1	Logic Temperature Fault
	0	Actuator Temperature Fault
4		Slave Exception Status
	7	Write Protect Fault
	6	Write Current Fault
	5	Write Transition Fault
	4	Head Offset Fault
	3	Data Strobe Fault
	2	reserved
	1	reserved
	0	reserved
5	7-0	Vendor Unique Status
6	7-0	Vendor Unique Status
7	7-0	Vendor Unique Status

8.2.1 Exception Status Octet (Octet 0)

8.2.1.1 Status Response (Octet 0 Bit 7). This bit is always zero for a Status Response.

8.2.1.2 Unsolicited Exception (Octet 0 Bit 6). This bit is set when the slave has incurred an Unsolicited Exception condition. At least one bit will be set in octet 1, describing the exception type. This bit and the bits in octet 1 can only be reset (on a port basis) by issuing a Read Status Response Bus Control followed by a Master Status Octet with the Successful Information Transfer bit set. As long as an Unsolicited Exception condition exists for a port, the slave shall reject all Bus Controls from that port and only that port, except Read Status Response, with the Unsolicited Exception bit set (bit 3) in the Slave Status Octet.

8.2.1.3 Bus Control Exception (Octet 0 Bit 5). This bit is set when the slave rejects the Bus Control as invalid, since it contains a parameter that is invalid, unsupported, out of context, or late. At least one bit shall be set in octet 2 to describe the type of command exception.

8.2.1.4 Read Fault (Octet 0 Bit 4). This bit is set when the slave detects an error while executing a Read/Verify Data Control. At least one bit shall be set in octets 3 through 7 to describe the type of read exception.

8.2.1.5 Write Fault (Octet 0 Bit 3). This bit is set when the slave detects an error while executing a Write Data Control. At least one bit will be set in octets 3 through 7 to describe the type of write exception.

8.2.1.6 Seek Fault (Octet 0 Bit 2). This bit is set when the slave detects a seek error while executing a Seek Command, Read/Verify Data, or Write Data Control. At least one bit shall be set in octets 3 through 7 to describe the type of seek exception.

8.2.1.7 Spindle Fault (Octet 0 Bit 1). This bit is set when the slave detects a spindle error while executing a Spin Up or Spin Down Command, Read/Verify Data, or Write Data Control. At least one bit shall be set in octets 3 through 7 to describe the type of spindle exception.

8.2.1.8 Execution Fault (Octet 0 Bit 0). This bit is set when the slave detects an Execution error other than Read, Write, Seek, or Spindle Fault during the execution of a Command or Data Control. At least one bit shall be set in octets 3 through 7 to describe the type of exception.

8.2.2 Unsolicited Exception Status (Octet 1)

8.2.2.1 Reset Complete (Octet 1 Bit 7). This bit is set when the slave has executed and completed a reset that may affect this port.

8.2.2.2 Alternate Port Priority Select (Octet 1 Bit 6). This bit is set when the alternate port receives a Selection Octet with the priority Select bit set.

8.2.2.3 Alternate Port Format Change (Octet 1 Bit 5). This bit is set when the alternate port receives a new format specification.

8.2.2.4 Alternate Port Format Complete (Octet 1 Bit 4). This bit is set when the alternate port receives a Load Slave Function code 16, Notify Alternate Port of Format Completion.

8.2.2.5 Reserved (Octet 1 Bit 3).

8.2.2.6 Not Ready Transition (Octet 1 Bit 2). This bit is set when the slave goes from a ready to a not ready condition.

8.2.2.7 Ready Transition (Octet 1 Bit 1). This bit is set when the slave goes from a not ready to ready condition.

8.2.2.8 Media Change (Octet 1 Bit 0). This bit is set with the Ready Transition bit (bit 1) if the slave detects that the media was removed and replaced previous to the Not Ready Transition to Ready Transition.

8.2.3 Bus Control Exception Status (Octet 2)

8.2.3.1 Invalid Bus Control (Octet 2 Bit 7). This bit is set when a Bus Control was received that is not defined in this standard.

8.2.3.2 Invalid Parameter (Octet 2 Bit 6). This bit is set when a valid Command Control was received with an invalid parameter.

8.2.3.3 Unsupported Bus Control (Octet 2 Bit 5). This bit is set when a valid Bus Control is received that is unsupported by this slave.

8.2.3.4 Bus Control Context (Octet 2 Bit 4). This bit is set when a valid Bus Control is received but cannot be executed because it conflicts with the current context of the slave.

8.2.3.5 Data Control Late (Octet 2 Bit 3). This bit is set when a valid Data Control is received later than the orientation window for the next field.

8.2.3.6 Reserved (Octet 2 Bit 2).

8.2.3.7 Reserved (Octet 2 Bit 1).

8.2.3.8 Reserved (Octet 2 Bit 0).

8.2.4 Slave Exception Status (Octet 3)

8.2.4.1 Speed Fault (Octet 3 Bit 7). This bit is set if the slave did not reach the required speed in the required time during the execution of a Spin Up Command, or loses speed during a Data Control.

8.2.4.2 Off Cylinder Fault (Octet 3 Bit 6). This bit is set if the slave did not come on cylinder in the required time during the execution of a Seek Command, or loses on cylinder during a Data Control.

8.2.4.3 Head Select Fault (Octet 3 Bit 5). This bit is set if the slave detects an invalid head selection during the execution of a Head Select Command, Position Command, or Data Control.

8.2.4.4 Reserved (Octet 3 Bit 4).

8.2.4.5 Reserved (Octet 3 Bit 3).

8.2.4.6 Voltage Fault (Octet 3 Bit 2). This bit is set if the slave detects a voltage out of range condition.

8.2.4.7 Logic Temperature Fault (Octet 3 Bit 1). This bit is set if the slave detects an over temperature condition in the slave electronics.

8.2.4.8 Actuator Temperature Fault (Octet 3 Bit 0). This bit is set if the slave detects an over temperature condition in the slave servo actuator.

8.2.5 Slave Exception Status (Octet 4)

8.2.5.1 Write Protect Fault (Octet 4 Bit 7). This bit is set when a Write Data Control is received and the slave is write protected.

8.2.5.2 Write Current Fault (Octet 4 Bit 6). This bit is set when write current was out of range during the execution of a Write Data Control.

8.2.5.3 Write Transition Fault (Octet 4 Bit 5). This bit is set if no write transitions are detected by the slave during the execution of a Write Data Control.

8.2.5.4 Head Offset Fault (Octet 4 Bit 4). This bit is set if the data heads were in an offset position when a Write Data Control was received.

8.2.5.5 Data Strobe Fault (Octet 4 Bit 3). This bit is set if the slave receives a Write Data Control when early or late data strobe is in effect.

8.2.5.6 Reserved (Octet 4 Bit 2).

8.2.5.7 Reserved (Octet 4 Bit 1).

8.2.5.8 Reserved (Octet 4 Bit 0).

8.2.6 Vendor Unique Status (Octet 5 Bits 7-0)

8.2.7 Vendor Unique Status (Octet 6 Bits 7-0)

8.2.8 Vendor Unique Status (Octet 7 Bits 7-0)

8.3 Extended Status Response. The slave transfers Extended Status Response to the master upon receiving a Read Extended Status Bus Control. The Extended Status bits are static indications of current flag states and slave conditions.

The Extended Status Response shall have the following format:

Octet	Bit	Description
0		Interface Flags
	7	Extended Status
	6	Port Number
	5	Alternate Port Enabled
	4	Reserve Active
	3	Command Complete Interrupt Enabled
	2	RPS Interrupt Enabled
	1	Status Pending Interrupt Enabled
	0	Format Specification Present
1		Data Recovery Flags
	7	Offset Direction
	6	Offset MSB
	5	Offset LSB
	4	Early Strobe
	3	Late Strobe
	2	reserved
	1	Header Field ECC/CRC Enable
	0	Data Field ECC/CRC Enable
2		Slave Control Flags
	7	Write Protected
	6	Spindle Power On
	5	Lock Carriage
	4	Load Heads
	3	reserved
	2	reserved
	1	reserved
	0	reserved

3		Slave Status
	7	Speed
	6	On Cylinder
	5	reserved
	4	reserved
	3	reserved
	2	reserved
	1	HDA Ready
	0	Media Present
4		Slave Alarms
	7	reserved
	6	reserved
	5	Illegal Head Select
	4	reserved
	3	reserved
	2	Voltage Range Error
	1	Logic Over Temperature
	0	Actuator Over Temperature
5	7-0	Vendor Unique Extended Status
6	7-0	Vendor Unique Extended Status
7	7-0	Vendor Unique Extended Status

8.3.1 Interface Flags (Octet 0)

8.3.1.1 Extended Status (Octet 0 Bit 7). This bit is always a one for an Extended Status Response.

8.3.1.2 Port Number (Octet 0 Bit 6). 0 = even port, 1 = odd port.

8.3.1.3 Alternate Port Enabled (Octet 0 Bit 5). This bit is set when the alternate port is enabled.

8.3.1.4 Reserve Active (Octet 0 Bit 4). This bit is set when this port has reserved the slave.

8.3.1.5 Command Complete Interrupt Enabled (Octet 0 Bit 3). This bit is set when Command Complete Interrupts are enabled to generate an Attention on this port.

8.3.1.6 RPS Interrupt Enabled (Octet 0 Bit 2). This bit is set when RPS Interrupts are enabled to generate an Attention on this port.

8.3.1.7 Status Pending Interrupt Enabled (Octet 0 Bit 1). This bit is set when Status Pending Interrupts are enabled to generate an Attention on this port.

8.3.1.8 Format Specification Present (Octet 0 Bit 0). The slave currently has a valid format specification loaded.

8.3.2 Data Recovery Flags (Octet 1)

8.3.2.1 Offset Direction (Octet 1 Bit 7). If a head offset is in effect, this bit indicates the direction of offset. This bit is cleared when positive offset (away from the spindle) is in effect, and set when negative offset (towards the spindle) is in effect.

8.3.2.2 Offset MSB (Octet 1 Bit 6). This bit is the most significant bit of the Head Offset magnitude.

8.3.2.3 Offset LSB (Octet 1 Bit 5). This bit is the least significant bit of the Head Offset magnitude. If bits 6 and 5 of this octet are cleared, then no head offset is in effect and bit 7 has no meaning.

8.3.2.4 Early Strobe (Octet 1 Bit 4). This bit is set if Early Data Strobe is active.

8.3.2.5 Late Strobe (Octet 1 Bit 3). This bit is set if Late Data Strobe is active.

8.3.2.6 Reserved (Octet 1 Bit 2).

8.3.2.7 Header Field ECC/CRC Enable (Octet 1 Bit 1). This bit is set if the slave ECC/CRC feature (optional) is enabled.

8.3.2.8 Data Field ECC/CRC Enable (Octet 1 Bit 0). This bit is set if the slave ECC/CRC feature (optional) is enabled.

8.3.3 Slave Control Flags (Octet 2)

8.3.3.1 Write Protected (Octet 2 Bit 7). This bit is set if the slave is currently write protected.

8.3.3.2 Spindle Power On (Octet 2 Bit 6). This bit is set while power is applied to the spindle.

8.3.3.3 Lock Carriage (Octet 2 Bit 5). This bit is set if a Lock Carriage Command is in effect.

8.3.3.4 Load Heads (Octet 2 Bit 4). This bit is set if a Load Heads Command is in effect.

8.3.3.5 Reserved (Octet 2 Bit 3).

8.3.3.6 Reserved (Octet 2 Bit 2).

8.3.3.7 Reserved (Octet 2 Bit 1).

8.3.3.8 Reserved (Octet 2 Bit 0).

8.3.4 Slave Status (Octet 3).

8.3.4.1 Speed (Octet 3 Bit 7). This bit is set if the slave spindle is at operating speed.

8.3.4.2 On Cylinder (Octet 3 Bit 6). This bit is set if the slave actuator is on cylinder.

8.3.4.3 Reserved (Octet 3 Bit 5).

8.3.4.4 Reserved (Octet 3 Bit 4).

8.3.4.5 Reserved (Octet 3 Bit 3).

8.3.4.6 Reserved (Octet 3 Bit 2).

8.3.4.7 HDA Ready (Octet 3 Bit 1). This bit is set when the slave is at speed, the actuator is on track, and the heads are loaded.

8.3.4.8 Media Present (Octet 3 Bit 0). This bit is set when the slave media is present.

8.3.5 Slave Alarms (Octet 4)

8.3.5.1 Reserved (Octet 4 Bit 7).

8.3.5.2 Reserved (Octet 4 Bit 6).

8.3.5.3 Illegal Head Select (Octet 4 Bit 5). This bit is set when the slave has the wrong head selected, multiple heads selected, or no head selected.

8.3.5.4 Reserved (Octet 4 Bit 4).

8.3.5.5 Reserved (Octet 4 Bit 3).

8.3.5.6 Voltage Range Error (Octet 4 Bit 2). This bit is active when the slave has one or more voltages out of range.

8.3.5.7 Logic Over Temperature (Octet 4 Bit 1). This bit is active when the slave logic is over maximum operating temperature.

8.3.5.8 Actuator Over Temperature (Octet 4 Bit 0). This bit is active when the slave actuator is over maximum operating temperature.

8.3.6 Vendor Unique Extended Status (Octet 5 Bits 7-0)

8.3.7 Vendor Unique Extended Status (Octet 6 Bits 7-0)

8.3.8 Vendor Unique Extended Status (Octet 7 Bits 7-0)

Appendixes

(These Appendixes are not part of American National Standard X3.130-1986, but are included for information only.)

Appendix A
Defect Map

The slave will have a Defect Map containing a list of defects found by the manufacturer. This Defect Map will be used by the master to reallocate defective storage when formatting the slave.

A1. Location

The Defect Map is located on the last track of the cylinder identified in the Configuration Information as the Address of Last Defect Map Cylinder. The Defect Map will be recorded successively on each previous track until the map is complete.

A2. Contents

Each data field in the Defect Map will be self-identifying. A Defect Map set will consist of data fields repeated three times, with each data field identified as to whether it is the first, second, or third data field of a set. If a data field is not readable without errors at the factory, then it will be identified. A set may consist of more than three data fields, being made up of three readable data fields and as many data fields as have read errors.

The tracks used for the Defect Map are formatted with the Manufacturer's Format Specification (see 6.2.4).

A3. Formats

A3.1 Defects. The locations of the defects may be specified in terms of a Track Defect List or a Sector Defect List. Each defect entry in the map is 12 octets long.

The defects in the Defect Map should be stored in order of increasing physical addresses. An exception may be made for those that are detected after the initial map was recorded and are added to the end of it by the manufacturer during final test. Any additions that are out of sequence should indicate this by setting the Flag bit within that data field.

A3.1.1 Track Defect List. In this type of list, which is used with slaves that support more than one sector size, the defect locations are specified in terms of octets from Index.

A3.1.2 Sector Defect List. In this type of list, which is used with fixed sectored slaves, the defect locations are specified in terms of sector number and octets from the start of the sector. As an option, the defects can be expressed in terms of field number within the sector.

A3.2 Maps. The map is recorded with each data field recorded error free three times sequentially. Each data field in a set is identical to the other two. The following description describes the contents per data field, with the

implicit understanding that each data field is physically recorded three times.

Each data field of the map starts with a two-octet count of the number of octets used in the field, not including itself. The first data octet is the bit-significant Flag octet, which is followed by the number of this data field within the set. All data fields within the map should either be identified by their data field number within a set, or set to X'FF' to be ignored.

The Number of Sets in the Map and the Number of This Set within the Map occupies the next two fields.

The first set in the Defect Map is numbered as 1. If there are no entries in the Defect Map, the Number of Sets in the Map is set to zero in all three members of the set.

The following two fields provide the address of the first track at which the master may store Format Specifications in the manufacturer's default format.

All but the last two octets remaining are 12-octet entries containing defect information. The last 12-octet defect entry in a data field cannot be split. Therefore, any remainder of less than 12 should be zeros. Similarly, the remainder following the last defect entry in the last data field of the map should be padded with zeros.

The two-octet CRC-16 ($X^{16}+X^{15}+X^{2}+1$) is the last pair of octets in the data field, and is considered part of the data area. It is the master's responsibility to check the CRC.

The format of each sector is as shown below:

Octet	Bit	Description
00-01		Number of octets following: equals n - 1
02		Flag octet
	7	Last data field of Defect Map
	6	Defect Map continues on next lower cylinder
	5	Sector Defect List used
	4	Defect Fields option used
	3	Defect entries in this sector out of sequence
	2-0	Reserved
03		Number of this Sector within a Set 00 = First 01 = Second 02 = Third 03-FE = Illegal FF = Ignore
04-05		Number of Sets in Defect Map
06-07		Number of this Set in the Defect Map
08-0B		Cylinder Address of Format Specification Storage (Note 1)
0C-0D		Head Address of Format Specification Storage
0E-11		Cylinder address of defect
12-13		Head address of defect
14-17		Offset of defect from Index (octets) (Note 2)
18-19		Length of defect (bits)
(m-B):(m-8)		Cylinder address of defect
(m-7):(m-6)		Head address of defect
(m-5):(m-2)		Offset of defect from Index (octets) (Note 3)
(m-1):m		Length of defect (bits)
(m+1):(n-2)		Zero (if any octets not filled by defect entries)
(n-1):n		CRC-16

NOTES:
(1) 'F....F' if not supported
(2) If bit 5 of the Flag octet is set, the Sector Defect List is used, and the Offset field has the following format:

15-16	Sector number
17-18	Offset of defect from start of sector (octets)

(3) If bits 5 and 4 of the Flag octet are set, the Defect Fields List is used, and the Offset field had the following format:

15-16		Sector number
17		n Field n defective
18	7-4	Vendor unique
	3-0	reserved

A4. Protection

It is recommended that the Defect Map be write protected by the manufacturer of the disk so that no action by the master can destroy the factory-supplied defect information.

Appendix B
Storage of Format Specifications and Attributes

A master typically has the need to store Format Specifications and Attributes on the disk.

The storage of the Format Specifications is responsibility of the master. These specifications may be stored anywhere on the media in the preferred format of the master. If the master wishes to record them in an area according to the manufacturer's default specification, the address of the area is defined in each data field of the Defect Map.

The recommended format of each data field is as follows:

Octet	Bit	Description
00-01		Number of octets following: equals n - 1
02		Flag octet
	7	Last data field of storage
	6-0	Reserved
03		Number of this Sector within a Set 00 = First 01 = Second 02 = Third 03-FE = Illegal FF = Ignore
04-05		Number of Sets in Format Specification
06-07		Number of this Set in the Format Specification
08		Partition
09		Alias address
0A-0D		Number of octets per Data Block
0E		Flags
	7	Data Block same size as Physical Block
	6	Data Block multiple of Physical Block
	5	Data Block nonmultiple of Physical Block
	4-1	Reserved
	0	Interleaves performed by master
0F		Cylinder interleave factor
10		Head interleave factor
11		Sector interleave factor
12		Number of alternate cylinders per Partition
13		Number of spare sectors per track
14-17		Partition starting cylinder address
18-1B		Number of cylinders in Partition
1C-1D		Number of octets in Format Specification m - 1
1E-(m+25)		Format Specification (see 6.2) repeat octets 4-(m+25) for additional Partitions
(m+26):(n-2)		Zero (if any octets not filled by entries)
(n-1):n		CRC-16

If the slave is not used with IPI Level 3 interface, octets 08-1B are not used and should be zeros.

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
X3.147-1988 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/TR1-82 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)

October 1987

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

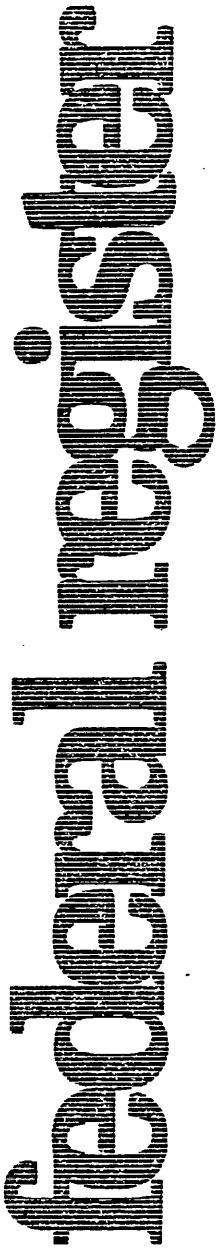
Copies of FIPS are available from:

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 51895-52038



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 ·th extensions for enhanced storage ·dule 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