FUJITSU

M2622Sx/ M2623Sx/ M2624Sx

Intelligent Disk Drives OEM Manual

– SCSI Logical Specification –

41FH5057E-02

Printed on Recycled Paper

REVISION RECORD

Edition	Date Published	Revised contents
01	Jan., 1991	
02	Jul., 1991	Pages1-31, 1-32, 3-24, 3-44, 3-46, 4-7 to 4-10, 4-18 to 4-20, 4-26, 4-27 revised.

Specification No.: 41FH5057E

The contents of this manual may be revised without prior notice.

Address your comments and inquiries on this manual to:

FUJITSU COMPUTER PRODUCTS
OF AMERICA, INC.
2904 Orchard Parkway
San Jose, CA 95134, U.S.A.
TEL: (408) 432-6333
FAX: (408) 894-1709

FUJITSU AUSTRALIA LIMITED
475 Victoria Avenue, Chatswood
N.S.W., 2067, AUSTRALIA
TEL: (61-2) 410-4555
FAX: 411-8603
TLX: 25233

FUJITSU CANADA, INC.
2800 Matheson Blvd. East, Mississauga
Ontario, L4W 4X5, CANADA
TEL: (1-416) 602-5454
FAX: 416-602-5457
TLX: 968132

FUJITSU DEUTSCHLAND GmbH
Frankfurter Ring 211
8000 München 40, F.R. GERMANY
TEL: (49-89) 32378-0
FAX: 32378-100
TLX: 897106

FUJITSU ESPAÑA, S.A.
Edificio Torre Europa
Paseo de la Castellana 95, Madrid 28046, SPAIN
TEL: (34-1) 581-8000
FAX: 581-8300
TLX: 23887

FUJITSU EUROPE LIMITED
2 Longwalk Road, Stockley Park, Uxbridge
Middlesex, UB11 1AB, ENGLAND
TEL: (44-81) 573-4444
FAX: 573-2643
TLX: 263871

FUJITSU FRANCE S.A.
17, rue Olof palme
94006 Créteil cedex, FRANCE
TEL: (33-43) 99 40 00
FAX: 99 07 00
TLX: 262 661

FUJITSU HONG KONG LIMITED
Room 1831, Sun Hung Kai Centre
30 Harbour Road, HONG KONG
TEL: (852-5) 8915780
FAX: 742917
TLX: 62667

FUJITSU ITALIA S.p.A.
Via Melchiorre Gioia, 8-20124 Milano, ITALY
TEL: (39-2) 6572741
FAX: 6572257
TLX: 350142

FUJITSU NORDIC AB
Torggatan 8, S-171 54 Solna, SWEDEN
TEL: (46-8) 764 76 90
FAX: 28 03 45
TLX: 13411

FUJITSU LIMITED
International Operations
Marunouchi 1-6-1, Chiyoda-ku, Tokyo 100, JAPAN
TEL: (81-3) 3216-3211
FAX: 3213-7174
TLX: J22833
Cable: "FUJITSU LIMITED TOKYO"

This page is intentionally left blank.

LIST OF EFFECTIVE PAGES

PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV
Cover	02	1-28	01	3-26	01	3-76	01	3-126	01
Blank	—	1-29	01	3-27	01	3-77	01	3-127	01
i	03	1-30	01	3-28	01	3-78	01	3-128	01
ii	02	1-31	01	3-29	01	3-79	01	4-1	01
iii	02	1-32	01	3-30	01	3-80	01	4-2	01
iv	02	1-33	01	3-31	01	3-81	01	4-3	01
v	02	1-34	01	3-32	01	3-82	01	4-4	01
vi	02	1-35	01	3-33	01	3-83	01	4-5	01
vii	03	1-36	01	3-34	01	3-84	01	4-6	01
viii	02	1-37	01	3-35	01	3-85	01	4-7	01
ix	02	1-38	01	3-36	01	3-86	01	4-8	01
x	02	2-1	01	3-37	01	3-87	01	4-9	01
xi	02	2-2	01	3-38	01	3-88	01	4-10	01
xii	02	2-3	01	3-39	01	3-89	01	4-11	01
xiii	02	2-4	01	3-40	01	3-90	01	4-12	01
xiv	02	2-5	01	3-41	01	3-91	01	4-13	01
xv	02	2-6	01	3-42	01	3-92	01	4-14	01
xvi	02	2-7	01	3-43	01	3-93	01	4-15	01
xvii	02	2-8	01	3-44	01	3-94	01	4-16	01
xviii	02	2-9	01	3-45	01	3-95	01	4-17	01
xix	02	2-10	01	3-46	01	3-96	01	4-18	01
xx	02	2-11	01	3-47	01	3-97	01	4-20	01
1-1	01	2-12	01	3-48	01	3-98	01	4-21	01
1-2	01	2-13	01	3-49	01	3-99	01	4-22	01
1-3	01	2-14	01	3-50	01	3-100	01	4-23	01
1-4	01	3-1	01	3-51	01	3-101	01	4-24	01
1-5	01	3-2	01	3-52	01	3-102	01	4-25	01
1-6	01	3-3	01	3-53	01	3-103	01	4-26	01
1-7	01	3-4	01	3-54	01	3-104	01	4-27	01
1-8	01	3-5	01	3-55	01	3-105	01	4-28	01
1-9	01	3-6	01	3-56	01	3-106	01	4-29	01
1-10	01	3-7	01	3-57	01	3-107	01	4-30	01
1-11	01	3-8	01	3-58	01	3-108	01	4-31	01
1-12	01	3-9	01	3-59	01	3-109	01	4-32	01
1-13	01	3-10	01	3-60	01	3-110	01	4-33	01
1-14	01	3-11	01	3-61	01	3-111	01	4-34	01
1-15	01	3-12	01	3-62	01	3-112	01	5-1	01
1-16	01	3-13	01	3-63	01	3-113	01	5-2	01
1-17	01	3-14	01	3-64	01	3-114	01	5-3	01
1-18	01	3-15	01	3-65	01	3-115	01	5-4	01
1-19	01	3-16	01	3-66	01	3-116	01	5-5	01
1-19	01	3-17	01	3-67	01	3-117	01	5-6	01
1-20	01	3-18	01	3-68	01	3-118	01	5-7	01
1-21	01	3-19	01	3-69	01	3-119	01	5-8	01
1-22	01	3-20	01	3-70	01	3-120	01	5-9	01
1-23	01	3-21	01	3-71	01	3-121	01	Reader Comment Card	
1-24	01	3-22	01	3-72	01	3-122	01		
1-25	01	3-23	01	3-73	01	3-123	01		
1-26	01	3-24	01	3-74	01	3-124	01	Reply Card	
1-27	01	3-25	01	3-75	01	3-125	01		

This page is intentionally left blank.

PREFACE

This manual describes the M2622Sx, M2623Sx and M2624Sx 3.5 inch intelligent disk drives with an embedded SCSI controller.

The purpose of this manual is to explain in detail the specifications of the various commands and their functions for assembling and using the magnetic disk drives in the user system. It is also to provide the required information to generate the host system software. This manual is prepared for users with a basic knowledge on using the intelligent disk drives in the computer system.

The configuration and description range of this manual concerning the disk. Drives are as indicated in the "Manual Organization". Use the other manuals as required.

The contents of this manual are as follows:

Chapter 1 COMMAND PROCESSING

This chapter describes the basic logical specifications concerning the SCSI command processing of the M2622Sx, M2623Sx and M2624Sx intelligent disk drives.

Chapter 2 DATA BUFFER MANAGEMENT

This chapter explains the operations of the data buffer structure, data transfer processing function, and cache feature in the M2622Sx, M2623Sx and M2624Sx intelligent disk drives.

Chapter 3 COMMAND SPECIFICATIONS

This chapter explains in detail the specifications and the procedures to use the SCSI commands in the M2622Sx, M2623Sx and M2624Sx intelligent disk drives.

Chapter 4 SENSE DATA AND ERROR RECOVERY PROCEDURE

This chapter explains the configuration and contents of the sense data that are reported to the host system when an error occurs, key information that are required to recover errors, procedures to recovery errors that the host system software is supposed to execute, and the retry processings to recover errors that the M2622Sx, M2623Sx and M2624Sx intelligent disk drives internally execute.

Chapter 5 DISK MEDIUM MANAGEMENT

This chapter explains the method to initialize the disk medium of the M2622Sx, M2623Sx and M2624Sx intelligent disk drives, procedures to of dispose defective disks, and the procedures to recover data.

The suffixes of the model names of the intelligent disk drives differ depending on the data format at the shipment (e.g.: M2622S, M2623SA and M2624SB). In this manual, M2624S are used as representative model names unless special classification is required. The intelligent disk drives may be referred to as "IDD", "drive", or "device" in this manual.

MANUAL ORGANIZATION

OEM MANUAL Specifications & Installation	1. General Description 2. Specifications 3. Data Format 4. Installation Prerequisites 5. Installation 6. Diagnosis and Maintenance
OEM MANUAL SCSI Physical Specifications	1. SCSI Bus 2. SCSI Message 3. SISI Bus Error Recovery Procedure
OEM MANUAL SCSI Logical Specifications <This manual>	1. Command Processing 2. Data Buffer Management 3. Command Specification 4. Sense Data and Error Recovery Procedure 5. Disk Management
CE MANUAL	1. Specifications and Unit Configuration 2. Diagnosis and Maintenance 3. Troubleshooting and Fault isolation 4. Removal and Replacement Procedures 5. Theory of Operation

FUNCTIONAL LIMITATION

The specifications and functions described in this manual are Limited as follows according to the version of this drive.

Usable version of these function are noticed by ENGINEERING CHANGE REQUEST/NOTICE.

	Function	Usable version of this drive		
		Machine version	EPROM version	Production revision of standard INQUIRY data (ASCII)
1	RECOVER DATA command	Current version of this drive cannot use.		
2	RECOVER ID command			

CONTENTS

		page
CHAPTER 1	COMMAND PROCESSING	1–1
1.1	Command Format	1–1
1.2	Status Byte	1–6
1.3	Command Processing Outline	1–9
1.3.1	Single command	1–9
1.3.2	Command link	1–10
1.3.3	Disconnect/reconnect processing	1–12
1.3.4	Synchronous mode data transfer	1–18
1.4	Command Stack Function	1–20
1.4.1	Command stack management	1–20
1.4.2	Exception of command stack	1–21
1.5	UNIT ATTENTION Condition	1–22
1.5.1	UNIT ATTENTION condition generation	1–22
1.5.2	Response and release conditions in UNIT ATTENTION condition pending state	1–23
1.5.3	UNIT ATTENTION condition multi-pending	1–25
1.6	Sense Data Pending State	1–25
1.6.1	Sense data pending condition	1–25
1.6.2	Response and release conditions in the sense data pending state	1–26
1.7	Command Processing Exception	1–27
1.7.1	Overlap command	1–27
1.7.2	Illegal LUN specification	1–28
1.7.3	Operation codes being reserved	1–28
1.7.4	Command processing in not ready state	1–28
1.7.5	Error recovery	1–30
1.7.6	Reset processing	1–33
1.7.7	Fatal hardware error	1–34
1.8	Data Block Addressing	1–35
1.8.1	Data space definition	1–35
1.8.2	Logical block addressing	1–37

CHAPTER 2 DATA BUFFER MANAGEMENT .... 2−1
2.1 Data Buffer .... 2−1
2.1.1 Data buffer configuration and basic operation .... 2−1
2.1.2 Operation mode control .... 2−5
2.2 Read-Ahead Cache Feature .... 2−8
2.2.1 Read-Ahead Caching operation .... 2−8
2.2.2 Caching parameter .... 2−10
2.2.3 Prefetching operation and prfetching amount .... 2−12
2.3 Write Cache Feature .... 2−13

CHAPTER 3 SPECIFICATIONS OF COMMANDS .... 3−1
3.1 Control/Sense Commands .... 3−1
3.1.1 TEST UNIT READY .... 3−1
3.1.2 INQUIRY .... 3−2
3.1.3 READ CAPACITY .... 3−10
3.1.4 MODE SELECT .... 3−12
3.1.5 MODE SENSE .... 3−47
3.1.6 NO OPERATION .... 3−53
3.1.7 REZERO UNIT .... 3−53
3.1.8 START/STOP UNIT .... 3−54
3.1.9 RESERVE UNIT .... 3−56
3.1.10 RELEASE UNIT .... 3−60
3.1.11 PRIORITY RESERVE .... 3−62
3.1.12 REQUEST SENSE .... 3−63
3.2 Data Access Commands .... 3−65
3.2.1 READ .... 3−65
3.2.2 READ EXTENDED .... 3−66
3.2.3 WRITE .... 3−67
3.2.4 WRITE EXTENDED .... 3−68
3.2.5 WRITE AND VERIFY .... 3−69
3.2.6 VERIFY .... 3−70
3.2.7 SEEK .... 3−71
3.2.8 SEEK EXTENDED .... 3−72
3.2.9 SET LIMITS .... 3−73
3.3 Format Commands .... 3−76
3.3.1 FORMAT UNIT .... 3−76

3.3.2 REASSIGN BLOCKS 3–88
3.3.3 READ DEFECT DATA 3–92
3.4 Maintenance and Diagnostic Commands 3–97
3.4.1 SEND DIAGNOSTIC 3–97
3.4.2 RECEIVE DIAGNOSTIC RESULTS 3–104
3.4.3 WRITE BUFFER 3–114
3.4.4 READ BUFFER 3–117
3.4.5 READ LONG 3–121
3.4.6 WRITE LONG 3–123
3.4.7 WRITE SAME 3–125
3.4.8 RECOVER ID 3–126
3.4.9 RECOVER DATA 3–127

CHAPTER 4 SENSE DATA AND ERROR RECOVERY 4–1
4.1 Sense Data 4–1
4.1.1 Sense data format and basic information 4–1
4.1.2 Sense data basic information 4–3
4.1.3 Sense data additional information 4–11
4.2 Initiator Error Recovery (Recommended) 4–12
4.2.1 END status analysis and error recovery 4–12
4.2.2 Sense data analysis and error recovery 4–15
4.2.3 Error logging 4–25
4.3 Disk Drive Error Recovery 4–26
4.3.1 Error status and retry process 4–26
4.3.2 Automatic alternate block allocation processing 4–30
4.3.3 Error recovery process control 4–32

CHAPTER 5 DISK MEDIUM MANAGEMENT 5–1
5.1 Defect Management 5–1
5.2 Disk Initialization 5–4
5.2.1 Initialization at installation 5–4
5.2.2 Re-initialization 5–6
5.3 Data Block Verification Method (Recommended) 5–7
5.4 Alternate Block Allocation 5–9

This page is intentionally left blank.

FIGURES

		page
1.1	Six-byte length basic CDB format	1–2
1.2	Ten-byte length basic CDB format	1–2
1.3	Status byte	1–7
1.4	Data space configuration	1–36
2.1	Data buffer configuration	2–2
2.2	Data buffer operation in read processing	2–3
2.3	Data buffer operation in write processing	2–5
2.4	Reconnection timing control parameter	2–6
2.5	Caching control parameter	2–10
3.1	Standard INQUIRY data	3–4
3.2	VPD information: page code list	3–8
3.3	VPD information: device serial number	3–9
3.4	READ CAPACITY data	3–11
3.5	Structure of MODE SELECT parameters	3–14
3.6	Parameter configuration of MODE SELECT command	3–16
3.7	MODE SELECT parameter: Read/write error recovery parameter	3–21
3.8	MODE SELECT parameter: Disconnect/reconnect parameter	3–29
3.9	MODE SELECT parameter: Format parameter	3–31
3.10	MODE SELECT parameter: Drive parameter	3–36
3.11	MODE SELECT parameter: Verify error recovery parameter	3–40
3.12	MODE SELECT parameter: Caching parameter	3–42
3.13	MODE SELECT parameter: Additional error recovery parameter	3–44
3.14	MODE SELECT parameter: Reconnection timing parameter	3–46
3.15	MODE SENSE command parameter configuration	3–50
3.16	SET LIMITS command: accessible range setting	3–74
3.17	FORMAT UNIT command parameter list configuration	3–79
3.18	Defect descriptor: block address format	3–82
3.19	Defect descriptor: format of byte distance from index	3–83
3.20	Defect descriptor: physical sector address format	3–84
3.21	REASSIGN BLOCK command: defect data list configuration	3–89
3.22	READ DEFECT DATA command: defect data list configuration	3–93
3.23	SEND DIAGNOSTIC command: parameter list configuration	3–100

3.24	SEND DIAGNOSTIC parameter: page code list	3–101
3.25	SEND DIAGNOSTIC parameter: logical/physical address conversion	3–102
3.26	SEND DIAGNOSTIC parameter: mode setting	3–103
3.27	SEND DIAGNOSTIC parameter: drive-inherent information	3–104
3.28	RECEIVE DIAGNOSTIC RESULTS command: response data configuration	3–105
3.29	RECEIVE DIAGNOSTIC RESULTS response data: page code list	3–106
3.30	RECEIVE DIAGNOSTIC RESULTS response data: logical/physical address conversion	3–107
3.31	RECEIVE DIAGNOSTIC RESULTS response data: mode setting	3–111
3.32	RECEIVE DIAGNOSTIC RESULTS response data: drive-inherent information	3–112
3.33	WRITE BUFFER command: buffer data (Mode = 000, 001)	3–115
3.34	READ BUFFER command: buffer data (Mode = 000, 001)	3–118
3.35	READ BUFFER command: buffer descriptor	3–120
3.36	READ LONG command: transfer data format	3–121
4.1	Extended sense data format	4–2
4.2	Sense key inherent information	4–4
4.3	END status analysis	4–13

TABLES

		page
1.1	Response to link-specified command	1–11
1.2	Command type and disconnect processing	1–13
1.3	Sense data in not ready state	1–29
1.4	Outline of recovery from SCSI bus error	1–31
1.5	Outline of recovery from disk drive error	1–31
1.6	Reset processing during writing	1–34
3.1	Combination of error recovery flags	3–25
3.2	MODE SENSE data type setting	3–49
3.3	FORMAT UNIT command defect process	3–86
3.4	Error recovery control flags in self-diagnostic test	3–98
3.5	Logical/physical address conversion: data block state indication	3–109
4.1	Sense keys	4–5
4.2	Sense and subsense codes	4–6
4.3	Sense data error classification	4–16
4.4	Error recovery procedure	4–21
4.5	Read error retry procedure	4–27
4.6	Disk drive errors and retry counts	4–33

This page is intentionally left blank.

Glossary

Bus condition: Asynchronous condition for causing SCSI bus status transition. There are two types of bus conditions, ATTENTION and RESET.

Bus phase: Name of an SCSI bus state. The SCSI bus is in one of the following phases: BUS FREE, ARBITRATION, SELECTION, RESELECTION, or INFORMATION TRANSFER. The INFORMATION TRANSFER phase is divided into DATA IN, DATA OUT, COMMAND, STATUS, MESSAGE IN, and MESSAGE OUT phases depending on the type of information being transferred.

CCS: Common Command Set which is the standard SCSI logical specification stipulated by a working committee of ANSI. Functions necessary for direct access devices are defined.

CDB: Command Descriptor Block -- a group of data that describes the command for I/O and is transferred from an initiator to a target.

Command: Issued to a target to direct an input/output operation and written as CDB.

Disconnect: Operation performed by the target to free itself from the SCSI bus and the initiator temporarily when SCSI bus operation becomes unnecessary during command processing.

Initiator: SCSI device that has initiated an input/output operation on the SCSI device. This can be abbreviated as INIT.

Logical unit: Simple unit of equipment that can be directed to perform one I/O operation on the SCSI bus.

LUN: Logical unit number used to identify a logical unit.

Message: Information that controls a series of bus phases and I/O sequence between the initiator and the target on the SCSI bus.

Reconnect: Operation performed by the target to reconnect itself with the initiator when operation on the SCSI bus becomes necessary after disconnection.

SCSI: Small computer system interface which is an input/output interface standardized by American National Standard Institute (ANSI). [Standard number: ANSI X3.131-1986]

SCSI device: General term for a device (Input/output device, I/O controller, and host adapter, etc.) connected to on SCSI bus.

SCSI ID: Physical device address used to identify an SCSI device on the SCSI bus. This number is specific to each SCSI device. SCSI IDs are #0 to #7, each corresponding to one bit on the data bus.

Sense code: One-byte of code attached to sense data identify the type of the detected error.

Sense data: Detailed information created by the target when any error is involved in the command termination status. This information is transferred to report the error.

Sense key: Four-bit code attached to sense data to identify the class of the detected error.

Status: One byte of information that is transferred from a target to an initiator on termination of each command to indicate the command termination status.

Target: SCSI device which performs I/O initiated by an initiator. It can be abbreviated as TARG.

REFERENCED STANDARDS

Item	Number	Name	Organization
1	ANSI X3.131-1986	American National Standard for Information Systems — Small Computer System Interface (SCSI)	American National Standards Institute (ANSI)
2	X3T9.2/85-52 Rev 4.B	COMMON COMMAND SET (CCS) of the Small Computer System Interface (SCSI)	American National Standards Institute (ANSI)
3	X3T9.2/86-109 Rev 10. C	Draft proposed American National Standard for Information systems — Small Computer System Interface - 2 (SCSI-2)	American National Standards Institute (ANSI)

This page is intentionally left blank.

CHAPTER 1 COMMAND PROCESSING

1.1 Command Format
1.2 Status Byte
1.3 Command Processing Outline
1.4 Command Stack Function
1.5 UNIT ATTENTION Condition
1.6 Sense Data Pending State
1.7 Command Processing Exception
1.8 Data Block Addressing

This chapter describes the basic logical specifications for command processing of the IDD.

Note:

The IDD operates as a target (TARG) on the SCSI bus. In the explanation of this chapter, the IDD may be described as "TARG" except the case that the IDD must be classified explicitly.

1.1 Command Format

The input/output command from the INIT to the IDD is issued by a Command Descriptor Block (CDB). The CDB is information transferred from the INIT to TARG in the COMMAND phase. In some commands, parameters required to execute the command in DATA OUT phase may be specified besides the CDB. The details are explained in the individual command specifications in Chapter 3.

The CDB of the IDD has two formats: 6-byte length CDB and 10-byte length CDB. Figures 1.1 and 1.2 show the these CDB formats.

	Bit 7	6	5	4	3	2	1	0
Byte 00	Operation code							
01	L U N			Logical block address (MSB)				
02	Logical block address							
03	Logical block address (LSB)							
04	Transfer data length							
05	Control byte							

Figure 1.1 Six-byte length basic CDB format

	Bit 7	6	5	4	3	2	1	0
Byte 00	Operation code							
01	LUN			0	0	0	0	0
02	Logical block address (MSB)							
03	Logical block address							
04	Logical block address							
05	Logical block address (LSB)							
06	0	0	0	0	0	0	0	0
07	Transfer data length (MSB)							
08	Transfer data length (LSB)							
09	Control byte							

Figure 1.2 Ten-byte length basic CDB format

The meaning of fields on CDB are explained below. The definition and meaning of the basic CDB format and field may differ depending on the command type. The details are explained in the individual command specifications in Chapter 3.

(1) Operation code

Bit 7	6	5	4	3	2	1	0
Group code			Command code				

All leading bytes of CDB indicate the format and type of the executed command.

a. Group code

The group code specifies the byte count and format of the CDB. The IDD uses the following group commands:

- Group 0 (000): Six-byte length CDB (See Figure 1.1.)
- Group 1 (001): Ten-byte length CDB (See Figure 1.2.)
- Group 2 (010): Ten byte length CDB (See Figure 1.2.)
- Group 6 (110): Ten-byte length CDB (See Figure 1.2.)
- Group 7 (111): Operation code being reserved (See Subsection 1.7.3.)

b. Command code

The command code specifies the command type in each group.

(2) Logical unit number (LUN)

This field specifies the address of the logical unit (device) connected to the control of TARG when the IDENTIFY message is not used. When the LUN is specified by the IDENTIFY message, the specified value in the LUN field of CDB is ignored.

Note:

The definition of this field may be changed to another meaning in the SCSI standard in the future. It is recommended that the INIT specifies LUN using the IDENTIFY message and sets zero to this CDB field.

(3) Logical block address

This field indicates the address of the leading logical data block of the data block group on the disk which is to be processed by the command. In CDB of group 0, a 21-bit block can be addressed, and in CDB of group 1, group 2, and group 6, 32-bit blocks can be addressed. The logical data block addressing procedure in the IDD are specified in Section 1.8.

(4) Transfer data length

This field specifies the length of the transferred data between the INIT and TARG by executing the command with the number of logical data blocks or number of bytes. Hereafter, the former will be referred to a "transfer block count" and the latter as "transfer byte length" or "parameter list length".

This field may be used for another meaning depending on the command type or may be meaningless. There are commands that allocate three or more bytes as a "transfer data length" field. The details are specified in the individual command specifications in Chapter 3.

a. Transfer block count

When predetermined as the transfer block count, this field specifies the number of logical data blocks to be transferred between the INIT and the IDD with the command.

In the "transfer block count" field of a 1-byte length command, it is assumed that 256 blocks are specified when the specified value of the field is zero. Block count can be specified in the range of 1 to 256. On the other hand, if the "transfer block count" field of a 2-byte length command is zero, no data is transferred. Blocks can be transferred in the range of 0 to 65,535.

b. Transfer byte length or parameter list length

When predetermined as the transfer byte length or parameter list length, this field specifies the length of the data to be transferred between the INIT and the IDD with the command in number of bytes. When zero is specified in this field, no data is transferred except for the case that is allowed specially in the command specification described in Chapter 3.

In the command that transfers the parameter required to execute the command from the INIT to the IDD, this field is referred to as "parameter list length". It specifies the total number of bytes of the parameter list that the INIT sends.

In the command (REQUEST SENSE, INQUIRY) that receives information from the IDD, this field is referred to as "transfer byte length". It specifies the maximum number of bytes (number of bytes in the area secured on the INIT to receive information) that the INIT can receive. The IDD transfers to the INIT only the smaller number of bytes of either the number of effective bytes of information specified in the command type or the specified value of the "transfer byte length" field.

(5) Control byte

Bit 7	6	5	4	3	2	1	0
0	0	0	0	0	0	Flag	Link

a. Link

When this bit is 1, the command link is specified. The details of the command link operation are explained in Subsection 1.3.2.

b. Flag

This bit is effective only when 1 is specified in the Link bit. When 0 is specified in the Link bit, 1 must not be specified in the Flag bit.

This bit specifies the type of message that the TARG transmits to the INIT when the command specified with link ends normally. The IDD notifies the INIT with a LINKED COMMAND COMPLETE (WITH FLAG) message when this bit is 1 and with a LINKED COMMAND COMPLETE message when this bit is 0.

Normally this bit is used to generate an interrupt against to the software in the INIT when the execution of a specific command in a command group that is contiguously linked ends.

c. Bits 7 and 6 (Vendor unique)

The specifications of these bits are meaningless except when specially defined in each command. The IDD ignores the specified value.

Note:

Bits 7 and 6 in the control byte may be used as an unique control field for the future drive specification. The INIT should specify '0' in this field.

(6) Illegal CDB handling

If there is an error in the CDB description (specification) or if there is an error in the specification of a parameter transferred from the INIT according to the CDB specification, the command terminates with a CHECK CONDITION status. For a command that changes data on the disk, the data is not changed in by the command if there is an error in the CDB specification. However, if there is an error in the parameter transferred in the DATA OUT phase, the contents of the disk may be changed within the range specified in the command. Even if there is an error in the CDB specification of the command that involves the DATA OUT phase, the DATA OUT phase may be executed after the COMMAND phase ends. For example, even if there is an error in the specification of the CDB in the WRITE command, the IDD transfers data of so many bytes (the transferred data length is undetermined) but the data is never written on the disk. The details are explained in the individual command specifications in Chapter 3.

In the command with disconnect processing (see Subsection 1.3.3), even if there is an error in the CDB specification, disconnect processing may be performed after the COMMAND phase ends. In this case, reconnect processing is performed thereafter and the CHECK CONDITION status is reported.

Note:

When the CDB which specifies the undefined group code (group 3, 4, or 7) is issued, the IDD requests the 10-byte transfer in the COMMAND phase and reports the CHECK CONDITION status after receiving is completed.

When the CDB group code 5 (undefined group code) is issued, the IDD requests the 12-byte transfer in the COMMAND phase and reports the CHECK CONDITION status after receiving is completed.

1.2 Status Byte

Figure 1.3 shows the status byte format and type of status supported by the IDD.

The status byte is 1-byte information with which the TARG notifies the INIT in the STATUS phase when a command terminates. It indicates the execution result of the command. For a state in which the IDD cannot execute a command when an input/output operation request is accepted, the status byte is reported. However, for an ABORT message, BUS DEVICE RESET message, or RESET condition, or when a command is cleared by a forced conversion to the BUS FREE phase because of an abnormal SCSI bus state, the status byte for that command is not reported.

After reporting the status byte in the STATUS phase, the TARG always sends the COMMAND COMPLETE or LINKED COMMAND COMPLETE (WITH FLAG) message and notifies the INIT on the validity of the status byte.

Bit 7	6	5	4	3	2	1	0
0	0	Status byte code					0

Bit 5 4 3 2 1	
0 0 0 0 0	GOOD status
0 0 0 0 1	CHECK CONDITION status
0 0 1 0 0	BUSY status
0 1 0 0 0	INTERMEDIATE/GOOD status
0 1 1 0 0	RESERVATION CONFLICT status

Figure 1.3 Status byte

(1) GOOD status

This status indicates that the command execution was completed successfully.

(2) CHECK CONDITION status

This status is reported in cases explained below. When the IDD reports this status, it generates sense data and displays the details of the cause. When the INIT receives this status, it issues a REQUEST SENSE command and is supposed to gather the sense data.

a) When the sense key of the sense data indicates a RECOVERED ERROR [=1], the last command which caused the CHECK CONDITION status, indicates that the error recovery processing by the IDD was completed successfully.

b) When the sense key of the sense data indicates a UNIT ATTENTION [=6], it indicates that the IDD had held the UNIT ATTENTION condition. The details of the UNIT ATTENTION condition are explained in Section 1.5.

c) In cases other than those explained above, it indicates that the command is unable to be executed or it terminated abnormally.

(3) BUSY status

This status indicates that the IDD is in a busy state and cannot accept any new commands. Normally, the INIT that receives this status reissues the original command after waiting for an appropriate time.

The IDD reports the BUSY status in the following cases (for the command stack feature, see Section 1.4).

a) When the IDD is executing, stacking a command, executing an initial self-diagnostics, a newly issued command (excluding commands not for disconnect processing explained in Subsection 1.3.3) is received but the INIT that issued the command does not satisfy the disconnect approval condition.

b) When the IDD is executing, stacking a command, executing an initial self-diagnostics, a command other than for link-specified disconnect processing (explained in Subsection 1.3.3) is received.

c) When the DISCONNECT message for command stacking is rejected by the INIT.

(4) INTERMEDIATE status

This status indicates that the link-specified command (a command in which the link bit is 1 and excludes the last command of the command group which is contiguously linked) was completed successfully. The link-specified command terminates abnormally and when the CHECK CONDITION and RESERVATION CONFLICT status are reported, the command link is disconnected and the subsequent linked commands are not executed.

(5) RESERVATION CONFLICT status

This status is a state in which the IDD is reserved by another INIT and cannot be used until the reserved state is released. Normally, the INIT that received this status reissues the original command after waiting for an appropriate time.

1.3 Command Processing Outline

1.3.1 Single command

The following shows the single command processing example which is the most basic operation on the SCSI bus. When the disconnect processing is permitted, disconnect/reconnect processing may be involved until the execution is completed depending on the type of the command, but this operation is omitted in the following explanation. The disconnect function is explained in Subsection 1.3.3.

1) The INIT sets the command, data, and status pointers to the initial value of the command.

2) After acquiring the right to use the SCSI bus in the ARBITRATION phase, the INIT selects the TARG in the SELECTION phase. After the SELECTION phase ends, the control rights of the SCSI bus are entrusted to the TARG.

3) When the TARG responds to the SELECTION phase, the TARG executes the MESSAGE OUT phase if the ATTENTION condition exists. Normally the INIT transmits the IDENTIFY message as the initial message and specifies the device (LUN) to be operated.

4) The TARG executes the COMMAND phase next and receives the CDB from the INIT.

   The TARG judges the length of the CDB by the group code of the first byte of the CDB, and requests the number of required bytes to be transferred.

5) The TARG interprets the contents of the command and executes the requested operation. If the command requires data transfer on SCSI bus, the TARG executes the DATA IN or DATA OUT phase.

6) When the execution of the command is complete, the TARG notifies the INIT of the STATUS byte that indicates the execution result.

7) To terminate the operation, the TARG notifies the INIT of the COMMAND COMPLETE message by the MESSAGE IN phase and enters the BUS FREE phase

1.3.2 Command link

The command link function allows TARG to perform two or more commands sequentially. Examples of command link processing are shown below.

1) The INIT sets the command, data, and status pointers to the initial value of the first command.

2) Acquisition of the right to use the SCSI bus, the TARG selection, and LUN specification by the IDENTIFY message are the same as in the single command.

3) The TARG receives a command from the INIT in the COMMAND phase but 1 is specified in the Link bit of the CDB control byte.

4) The TARG interprets the command and executes the requested processing.

5) The TARG informs the INTERMEDIATE status of the STATUS phase when the command processing is completed successfully. The command link function becomes effective in this case.

6) The TARG notifies the INIT in the MESSAGE IN phase, of the LINKED COMMAND COMPLETE or LINKED COMMAND COMPLETE WITH FLAG message according to the value of the Flag bit of the CDB control byte. The INIT updates the initial value of the next command in which the command, data, and status pointers are linked, when it receives the LINKED COMMAND COMPLETE (WITH FLAG) message.

7) The TARG enters the COMMAND phase following its conversion to the MESSAGE IN phase and receives the command to be executed next. It subsequently performs the single command processing (Link bit=0) or command link processing (Link bit=1) according to the value of the Link bit of the CDB control byte.

The command link continues until a command in which 0 is specified in the CDB Link bit is issued or the command terminates abnormally. The command link function is effective only when the link-specified command is completed successfully. If the link-specified command is completed with an error or in an exceptional state, the command link function becomes ineffective. Table 1.1 shows the IDD response when the link-specified command is completed.

Table 1.1 Response to link-specified command

Flag	Ending state	Status	Message	Link function
0	Normal end	INTERMEDIATE/ GOOD	LINKED COMMAND COMPLETE	Effective
1	Normal end	INTERMEDIATE/ GOOD	LINKED COMMAND COMPLETE WITH FLAG	Effective
×	Abnormal end	CHECK CONDITION	COMMAND COMPLETE	Ineffective
×	Start unacceptable	BUSY	COMMAND COMPLETE	Ineffective
×	Reserved state	RESERVATION CONFLICT	COMMAND COMPLETE	Ineffective

The single logical unit is the only one that can be operated in the series of the linked command. The IDD processes the sequentially linked command in the logical unit specified by the IDENTIFY message or the LUN field of CDB when the first command is received. The specified values in the LUN fields of the second and subsequent CDBs are ignored.

Note:

The INIT that uses the command link must set the ATN signal to TRUE in the SELECTION phase and notify TARG that it is capable of receiving messages other than the COMMAND COMPLETE message. If the INIT does not set the ATN signal to TRUE and specifies 1 for the Link bit of CDB, the IDD completes the command in the CHECK CONDITION status (ILLEGAL REQUEST [=5]/ Invalid Field in CDB [24-00]).

1.3.3 Disconnect/reconnect processing

When processing that is not required the operation on the SCSI bus in the command execution process of the TARG is performed, the TARG can temporarily enter the BUS FREE phase on the SCSI bus by disconnect processing and execute the command in the TARG. With this function, the INIT can perform multiple command processing on the SCSI bus.

(1) Enable disconnect condition and command for disconnect processing

The IDD executes the disconnect processing when all of the following enable disconnect conditions are satisfied. However, as shown in Table 1.2, whether disconnect is to be processed or not and the execution timing of the disconnect processing differs depending on the type of the command.

Enable disconnect condition:

1) The SCSI ID of the INIT is notified by the SELECTION phase.

2) The INIT has generated an ATTENTION condition in the SELECTION phase.

3) The IDENTIFY message has notified that the INIT has permitted the disconnect processing.

Table 1.2 Command type and disconnect processing (1/2)

	Command	Code	Command for not processing disconnect (*1)	Command that executes disconnect processing only when stacked (*2)	Command that executes the disconnect processing (*3)
F	FORMAT UNIT	(04)			○
I	INQUIRY	(12)	○		
M	MODE SELECT	(15)			○
	MODE SENSE	(1A)			○
N	NO OPERATION	(0D)		○	
P	PRIORITY RESERVE	(14)	○		
R	READ	(08)			○
	READ BUFFER	(3C)		○	
	READ CAPACITY	(25)			○
	READ DEFECT DATA	(37)			○
	READ EXTENDED	(28)			○
	READ LONG	(3E)			○
	REASSIGN BLOCKS	(07)			○
	RECEIVE DIAGNOSTIC RESULTS	(1C)		○	
	RECOVER DATA	(D8)			○
	RECOVER ID	(DA)			○
	RELEASE UNIT	(17)	○		
	REQUEST SENSE	(03)	○		
	RESERVE UNIT	(16)		○	
	REZERO UNIT	(01)			○
S	SEEK	(0B)			○
	SEEK EXTENDED	(2B)			○
	SEND DIAGNOSTIC	(1D)			○
	SET LIMITS	(33)		○	
	START/STOP UNIT	(1B)			○
T	TEST UNIT READY	(00)	○		
V	VERIFY	(2F)			○

Table 1.2 Command type and disconnect processing (2/2)

	Command	Code	Command for not processing disconnect (*1)	Command that executes disconnect processing only when stacked (*2)	Command that executes the disconnect processing (*3)
W	WRITE	(0A)			O
	WRITE AND VERIFY	(2E)			O
	WRITE BUFFER	(3B)		O	
	WRITE EXTENDED	(2A)			O
	WRITE LONG	(3F)			O
	WRITE SAME	(41)			O

*1 Command for not processing disconnect: Stack processing (see Section 1.4) and disconnect processing are not performed by these commands in their execution sequence.

*2 Command that executes disconnect processing only when stacked: The disconnect processing is performed only when the commands are stacked (see Section 1.4). When the command execution is started after reconnection, the disconnect processing is not executed until the execution ends.

*3 Command that executes disconnect processing: Regardless of whether the commands are stacked, disconnect processing is performed in the command execution processing (after completion of the COMMAND phase, during the data transfer, or after completion of the data transfer). However, depending on the data state (when cache is hit), or on the specification of the processing contents in commands that have several processing modes (other than the case that the commands are stacked), the disconnect processing may not be executed.

(2) Basic disconnect processing procedure

The disconnect processing is executed in the following manner.

1) When the IDD decides that the SCSI bus can be disconnected during the execution of a command, it sends a DISCONNECT message to the INIT and enters the BUS FREE phase. If necessary, IDD sends a message to operate the INIT pointer before the DISCONNECT message. For details, refer to Chapter 2 of OEM Manual SCSI Physical Specifications.

2) After the IDD enters the BUS FREE phase, the INIT can issue other commands to another TARG or other logical units. The IDD that processed disconnect can accept input/output operation requests from another INIT (see Section 1.4).

3) The IDD executes the command that processed disconnect.

4) Thereafter, the IDD executes the reconnection processing when the operation in SCSI is required, and reconnects the INIT. (The reconnection processing is explained in Item (5) below.)

(3) Disconnect processing procedure after COMMAND phase execution

When a command is stacked (see Section 1.4), when it takes a time to start data transfer requested by the command (example: the READ command that does not hit the cache memory) or when data transfer is not necessary but it takes a time to report the command execution result (status) (example: VERIFY command), disconnect processing is performed just after the execution of the COMMAND phase. In this case, the IDD enters the MESSAGE IN phase from the COMMAND phase and sends the DISCONNECT message to the INIT (SAVE DATA POINTERS message is not sent).

(4) Disconnect processing procedure after data transfer has been executed

In a command that involves data transfer, disconnect may be processed during the execution of DATA IN or DATA OUT phase or after the last data transfer is completed. In this case, it must be noted that the following pointer control is executed.

a. Disconnection during data transfer

After starting data transfer (DATA IN or DATA OUT phase), when it takes time to be able to transfer the subsequent data (example: the data buffer is empty by the READ command, or the data buffer is full by the WRITE command), disconnect processing is performed during the data transfer. The IDD sends the SAVE DATA POINTER message before sending the DISCONNECT message. When the INIT receives the SAVE DATA POINTER message, it must save the current data pointer value and be ready for resumption of the subsequent data transfer when reconnected.

b. Disconnection after the last data transfer is completed

When disconnect processing is performed after the transfer of all data required to execute the command is completed successfully (example: WRITE command), the IDD sends the DISCONNECT message immediately after the SAVE DATA POINTER message sending.

The IDD does not request the data transfer after reconnection, and immediately enters the STATUS phase and reports the status byte.

Note:

In this disconnect processing, since the transfer of all data involved in the execution of the command is completed, essentially, the SAVE DATA POINTER message is not required.

When the SAVE DATA POINTER message is issued, the time to process the message transfer increases. However, the current data pointer can reflect the final result of the data transfer as a result of the restore operation of the pointer executed in the INIT when reconnection is processed to report the status.

c. Disconnection to retry data transfer

When an error occurs on the SCSI bus in the DATA IN or DATA OUT phase and data between the INIT and the IDD is retransferred, the IDD transmits the RESTORE POINTERS message. If data can be retransferred (retried) immediately after that, the IDD enters the DATA IN or DATA OUT phase and retries data transfer without executing the disconnect processing. When it takes time to retransfer the data, the IDD transmits the DISCONNECT message immediately after the RESTORE POINTERS message is sent and performs disconnect processing. In this case, the INIT must hold the saved data pointer value to be ready for data retransfer (retry) after reconnection.

(5) Exception of disconnect processing

When the disconnect processing is executed as previously explained, an ATTENTION condition is generated for the DISCONNECT message that the IDD has transmitted. When the INIT sends back the MESSAGE REJECT message, the disconnect processing is not executed and the IDD executes the command with the SCSI bus connected. In a disconnect processing that requires a pointer operation, it is the same as when the SAVE DATA POINTER message is rejected. However, if disconnect processing can be performed again in the command execution sequence after that, the IDD attempts to perform the disconnect processing again if disconnect processing is permitted at that time.

The INIT enables/disables the control of the disconnect processing by the IDENTIFY message. For example, several IDENTIFY messages may be issued in the execution sequence of a command to enable or disable the disconnect processing.

For details, refer to Chapter 3 of OEM Manual SCSI Physical Specifications.

(6) Reconnection processing procedure

The reconnection processing procedure is as follows:

1) The IDD executes the ARBITRATION phase when processing on the SCSI bus is required, obtains the control of the SCSI bus and reconnects with the INIT in the RESELECTION phase.

2) After that, the IDD sends the IDENTIFY message to the INIT to notify it of the logical unit address (LUN) that the reconnection processing requires. The INIT at this point, fetches the saved pointer (command, data, and status) that corresponds to the specified LUN and restores it in the current pointer.

Notes:

1. In the RESELECTION phase, if there is no response from the INIT within a predetermined time (250 ms), the IDD processes timeout and enters the BUS FREE phase. The IDD retries (reexecution of the RESELECTION phase) a fixed number of times after waiting more than 200 μs. If the IDD still cannot reconnect with the INIT, the IDD clears the command that required reconnection processing and generates sense data that indicates ABORTED COMMAND [=B]/Select/Reselect failure [=45-nn].

2. When the INIT rejects the IDENTIFY message, the IDD clears the command that is processing reconnection and enters the BUS FREE phase. The IDD generates sense data that indicates ABORTED COMMAND [=B]/Message Error [=43-00].

3. The INIT that successfully accepted the IDENTIFY message must change the ACK signal for the IDENTIFY message to false after the restore operation of the pointer is completed. When the ACK signal at the time of the IDENTIFY message transfer changes to false, the IDD assumes that the reconnection processing was completed successfully and starts the subsequent processings if the ATTENTION condition does not exists.

For details, refer to Chapter 5 of OEM Manual Specifications & Installation, and Chapters 1 and 3 of OEM Manual SCSI Physical Specifications.

1.3.4 Synchronous mode data transfer

The IDD supports a synchronous mode data transfer to process data transfer (DATA IN and DATA OUT phases) in the SCSI bus at high speed. Data in the SCSI bus may be transferred either in the asynchronous or synchronous mode. However, when the synchronous mode data transfer is used, the command processing time is reduced and the input/output throughput can be improved by multiple command processing, using the SCSI bus free time.

To use the synchronous mode data transfer function, all of the following functions must be satisfied.

Enable synchronous mode transfer condition is that the INIT generates the ATTENTION condition when the SELECTION phase is executed.

Even if the above condition is satisfied, the IDD data transfer mode (default mode) will be the asynchronous mode after a RESET condition occurs in the SCSI bus after the IDD power is turned on or after one of the INITs issues a BUS DEVICE RESET message. To use the synchronous mode data transfer, the INIT must exchange a SYNCHRONOUS DATA TRANSFER REQUEST message between itself and the IDD to determine in advance the parameters required to execute the synchronous mode transfer.

The INIT that uses the synchronous mode data transfer normally sends the SYNCHRONOUS DATA TRANSFER REQUEST message following the sending of the IDENTIFY message after the first SELECTION phase for the IDD. The INIT then requests TARG (IDD) to change the data transfer mode to the synchronous mode. The data transfer mode once determined between the TARG and the INIT is effective until a RESET condition is generated or a BUS DEVICE RESET message is issued by one of the INITs. Therefore, to avoid the overhead time required to exchange the message, the INIT should not send the SYNCHRONOUS DATA TRANSFER REQUEST message to TARG each time the SELECTION phase is executed.

When the "enable synchronous mode transfer condition" described above is satisfied and the IDD holds the default (asynchronous) transfer mode, and the SYNCHRONOUS DATA TRANSFER REQUEST message is not received from the INIT, and synchronous mode transfer request is permitted in the IDD setting terminal, the IDD sends to the INIT the above message in the MESSAGE IN phase after the COMMAND phase, and tries to set the parameter of the synchronous mode transfer.

The IDD holds a data transfer mode individually for each INIT. Therefore, INITs that use asynchronous mode transfer and synchronous mode transfer may exit mixed on the same SCSI bus. The synchronous mode transfer parameters determined by the SYNCHRONOUS DATA TRANSFER REQUEST message also may differ for each INIT.

Note:

> When the INIT first issues a command after the TARG power is turned on or after the RESET condition is generated, it may transmit the SYNCHRONOUS DATA TRANSFER REQUEST message. However, when a data transfer mode previously established by the BUS DEVICE RESET message issued by another INIT is reset in TARG, the INIT is not aware of it. Since the TARG (IDD) side transmits the SYNCHRONOUS DATA TRANSFER REQUEST message to reestablish the synchronous mode transfer in such a case, and synchronous mode transfer request is permitted in the IDD setting terminal, the INIT must reset the required parameters in response to the message.

For details, refer to Chapter 5 of <u>OEM Manual Specifications & Installation</u>, and Chapters 1 and 2 of <u>OEM Manual SCSI Physical Specifications</u>.

1.4 Command Stack Function

To reduce the overhead of input/output processings under the multiple initiator environment, the IDD is equipped with a command stacker that can accept multiple commands issued by the INITs (1 command/INIT maximum). By using this command stacker feature, an INIT can issue a command to the IDD without considering whether the IDD is executing a command issued by another INIT.

1.4.1 Command stack management

When the IDD receives a new command from an INIT while either processing a command issued by another INIT, is prefetching the minimum amount of data using the Read-Ahead cache feature or is executing an initial self-diagnostics, the IDD stacks the command in the command stacker. After the command stack processing is completed, the IDD performs the disconnect process.

When the processing currently being executed is completed, the IDD retrieves a stacked command to execute it. When two or more commands are stacked, they are retrieved and executed in the stacked order.

When a RESET condition occurs or when a BUS DEVICE RESET message is received from one of the INITs while commands are being stacked, the IDD clears all stacked commands. The IDD then generates an UNIT ATTENTION condition for all INITs.

When a PRIORITY RESERVE command is issued by an INIT other than the INIT that issued the stacked and executed commands while commands are being stacked, the IDD clears all stacked and executed commands see Subsection 1.4.2).

When an ABORT message is sent from the INIT that issued the stacked command and the correct LUN (0) is specified, the commands only which that INIT issued are cleared and the issued ABORT message does not effect other stacked command. When the LUN is not specified or illegal LUN (1 to 7) is specified, the issued ABORT message does not effect the stacked commands.

1.4.2 Exception of command stack

(1) TEST UNIT READY, REQUEST SENSE, INQUIRY, or RELEASE UNIT command

When one of these commands is received and there is no instruction to link the command, the IDD immediately executes the command without executing either the stack processing or disconnect processing. An executed or stacked command issued by another INIT is not affected.

In the case that there is an instruction to link these commands, when the IDD is stacking or executing a command issued by another INIT or is executing an initial self-diagnostics, the IDD does not execute the received command but responds with the BUSY status. When the IDD is not stacking or executing, the command group is immediately executed.

(2) PRIORITY RESERVE command

Even if a command issued by another INIT is already being stacked or executed, the IDD immediately executes the received PRIORITY RESERVE command. All commands being executed or stacked at that time are cleared. The IDD generates sense data that indicates ABORTED COMMAND [=B]/Command Aborted by PRIORITY RESERVE [=88-00] for all commands that are cleared.

(3) When disconnect cannot be processed

When the INIT that issued the command unable to process disconnect because the "enable disconnect conditions" (sse Subsection 1.3.3) are not satisfied or when the DISCONNECT message is rejected by the INIT even if the "enable disconnect conditions" are satisfied, the IDD responds with the BUSY status without stacking the received command if the IDD is executing or stacking other command or the IDD is executing an initial self-diagnostics except when item (1) or (2) is applicable. If the IDD is not executing or stacking, the IDD executes the received command immediately.

(4) When IDD is reserved

When the IDD is reserved by one of the INITs with the RESERVE UNIT command or PRIORITY RESERVE command, RESERVATION CONFLICT status is responded when a newly received command subsequently conflicts with the reserved state. The command is not stacked. Details of reserve state conflict is explained in the explanation of the RESERVE UNIT command (Subsection 3.1.9).

Note:

The IDD does not respond with the BUSY status for commands issued by the INIT because of the command stack feature function except the case as described in this subsection. This function is useful under a multiple initiator environment and eliminates the overhead to reissue commands caused by the BUSY status. Normally, the INIT does not have to be aware of the existence of the command stack feature, but must manage the input/output processing by paying attention to the following items.

1) When commands are stacked, the time to actually execute the stacked commands differs according to the commands stacked previously or according to the contents of the process being executed at that time. Especially when the FORMAT UNIT command, the START/STOP command (Immed = 0), or the data access commands that specifies the large number of blocks to be processed is being stacked or executed, the newly stacked commands must wait a long time before they are executed.

2) Even if commands are stacked, they are not executed in the following cases.

   a) If there is an error in the CDB specification, CHECK CONDITION status is sent when the command is retrieved from the stack.

   b) When the IDD is in a not ready state when the stacked command is retrieved, the CHECK CONDITION status is sent.

   c) When the MODE SELECT parameter (see Chapter 3) that concerns the disk drive data format is changed by a command from another INIT before a stacked command is retrieved, a UNIT ATTENTION condition is generated for all other INITs after execution of the MODE SELECT command is completed. Therefore, the CHECK CONDITION status is sent for all commands being stacked.

1.5 UNIT ATTENTION Condition

The UNIT ATTENTION condition is a function to notify the INIT of an event (change in state) that occurred in asynchronous node to the INIT on the TARG or the logical unit.

1.5.1 UNIT ATTENTION condition generation

One of the following events generates the UNIT ATTENTION condition.

(1) When power-on, RESET or BUS DEVICE RESET occurrence

When the IDD is reset by power-on, a RESET condition, or the BUS DEVICE RESET message, regardless of whether the disk drive is ready, the UNIT ATTENTION condition occurs for all INITs.

(2) Mode parameters changed (when changed by another initiator)

1) When one of the parameters specified in the MODE SELECT command, the parameter that concerns data format (either block descriptor, page 3: format parameter, or page 4: drive parameter) is changed by one of the INITs, this UNIT ATTENTION condition occurs for all INITs except for the one that changed the parameter.

2) When the RESERVE command that specifies the third party reserve feature is executed and the MODE SELECT parameter of the INIT that issued the RESERVE command is duplicated into the MODE SELECT parameter for the third party device, the UNIT ATTENTION condition occurs for the third party device.

(3) Rotational position locking (RPL)

When one of following matter regarding spindle synchronization occurs, the UNIT ATTENTION condition occurs for all INITs.

- Spindle synchronized

  The spindle synchronization, which is specified by the MODE SELECT command (page 4: drive parameter), is completed.

- Spindle not synchronized

  When the spindle synchronization is specified by the MODE SELECT command (page 4: drive parameter), the spindle cannot be synchronized. Or, the spindle synchronization is broken by some causes.

1.5.2 Response and release conditions in UNIT ATTENTION condition pending state

The UNIT ATTENTION condition that the IDD generates as a result of events described above is individually held for each INIT. It is kept until each pending INIT clears the UNIT ATTENTION condition by issuing a command defined below.

When the IDD is holding an UNIT ATTENTION condition and receives a command from the pending INIT, the IDD performs one of the following operations depending on the issued command.

(1) Commands other than INQUIRY, REQUEST SENSE, or PRIORITY RESERVE

The IDD reports the CHECK CONDITION status to the issued command. The UNIT ATTENTION condition for the INIT is cleared when the CHECK CONDITION status is reported. The generated sense key of the sense data is UNIT ATTENTION [=6]. The sense code below shows the events that generated the UNIT ATTENTION condition.

- Power-on, RESET, or BUS DEVICE RESET occurred [=29-00]
- Mode parameters changed [=2A-01]
- Spindle synchronized [=5C-01]
- Spindle not synchronized [=5C-02]

However, when the IDD is reserved by another INIT, the RESERVATION CONFLICT status is reported without clearing the UNIT ATTENTION condition except for the RELEASE UNIT command. For the RELEASE UNIT command, the IDD reports the CHECK CONDITION status and clears the UNIT ATTENTION condition as it is done when the IDD is not reserved. Since the issued command is not accepted, the UNIT ATTENTION condition is not cleared even if the IDD responds with the BUSY status.

(2) INQUIRY command

The INQUIRY command is successfully executed but the UNIT ATTENTION condition is not cleared.

(3) REQUEST SENSE command

One of the following operations is performed depending on the sense data pending state by the IDD (see Section 1.6).

a. For sense data pending state

The IDD successfully executes the REQUEST SENSE command and send the sense data being held to the INIT. The UNIT ATTENTION condition is not cleared in this case.

b. For sense data nonpending state

The IDD successfully executes the REQUEST SENSE command and sends to the INIT, sense data that indicates the UNIT ATTENTION condition being held. The UNIT ATTENTION condition for the INIT is cleared in this case.

(4) PRIORITY RESERVE command

The IDD successfully executes the PRIORITY RESERVE command. The UNIT ATTENTION condition for the INIT is cleared in this case.

1.5.3 UNIT ATTENTION condition multi-pending

When an UNIT ATTENTION condition occurs and other UNIT ATTENTION condition caused by another factor occurs before the first UNIT ATTENTION condition is cleared by the pending INIT, these UNIT ATTENTION conditions are held multiplely and the IDD posts these UNIT ATTENTION conditions sequentically. For example, after the UNIT ATTENTION condition "Power on, RESET or BUS DEVICE RESET occurred" is posted, the UNIT ATTENTION condition "Mode parameters changed", "Spindle synchronized" or "Spindle not synchronized" are posted for the issued command.

1.6 Sense Data Pending State

1.6.1 Sense data pending condition

The IDD generates sense data under the following conditions. The generated sense data is held until the release condition explained in Subsection 1.6.2 is satisfied. It is individually held for each INIT issuing the command, and sense data pending state is also kept for each of the INITs individually.

1) The IDD generates sense data for commands that terminate with CHECK CONDITION status when the IDD sends the CHECK CONDITION status.

2) The IDD generates sense data for commands being executed on the SCSI bus when an unrecoverable error occurs on the bus and the IDD enters from the current phase to the BUS FREE phase forcibly.

   Remark: However, the IDD does not generate sense data when the LUN cannot be specified until the error occurs.

3) The IDD generates sense data for a command requiring reconnection when clearing the command because there is no response from the INIT in the RESELECTION phase and reconnection cannot be executed.

4) The IDD generates sense data for all commands being cleared when commands that are stacked or being executed are cleared by the PRIORITY RESERVE command.

Notes:

1. In items 3) and 4) above, the command terminates abnormally and the IDD holds the sense data without explicitly sending the error occurrence to the INIT. The INIT should issue the REQUEST SENSE command to check for the error contents when recognizing the error occurrence if there is no response from the IDD a for long time after issuing a command.

2. The IDD holds special sense data without a pending INIT when the sense data pending INIT cannot be specified because the INIT SCSI ID is not sent in the SELECTION phase. In such a case, the IDD assumes a new SELECTION phase when no INIT SCSI ID is sent from the sense data pending INIT.

1.6.2 Response and release conditions in the sense data pending state

This subsection explains the response of the IDD when the IDD receives a new command in the sense data pending state and the release condition in the sense data pending state.

1) When there are stacked commands, the IDD executes the commands even if in the sense data pending state, however, this does not influence the pending sense data and the sense data pending state for the pending INITs.

2) The IDD normally accepts and executes any command issued from the INITs other than the sense data pending INITs. However, this does not influence the pending sense data and the sense data pending state for the pending INITs.

3) The IDD normally accepts and executes any command issued from sense data pending INITs to logical units other than sense data pending logical units. However this does not influence the pending sense data and the sense data pending state for the pending INITs.

4) The IDD normally accepts any command issued from the sense data pending INITs to a logical unit in the sense data pending state. In issuing commands other than the NO OPERATION command, the sense data pending state is released. In issuing the NO OPERATION command, the sense data pending state is not released but the command is successfully executed. In issuing the REQUEST SENSE command, the pending sense data is transferred to the INITs and the sense data pending state is released. In issuing any command other than above commands, the command will be successfully executed, the sense data pending state is released and the pending sense data is lost.

5) The sense data pending state is released and the pending sense data is lost when one of the following conditions is satisfied:

   a) The RESET condition occurs on the SCSI bus.

   b) The BUS DEVICE RESET message is issued from an INIT.

   c) The ABORT message is issued from a sense data pending INIT to a logical unit in the sense data pending state.

1.7 Command Processing Exception

1.7.1 Overlap command

After issuing a command to a logical unit, no INIT can issue any other command to the same logical unit until the command is completed.

The command completion is generally defined as when the TARG sends the COMMAND COMPLETE message to the INIT. However, the command execution is also completed with the RESET condition, BUS DEVICE RESET message, or ABORT message (see Subsection 1.7.5 for details).

When a command is issued from an INIT to a logical unit and another command is issued from the same INIT to the same logical unit before the first command is completed, the IDD makes the first command (that is stacked or being executed) and second command terminate abnormally in the following ways.

1) When the second command is issued while the first one is being executed, the IDD stops execution of the first one. When the second command is issued while the first one is stacked but its execution is not yet started, the IDD clears it.

2) The IDD sends the CHECK CONDITION status to the second command issued form the INIT and makes both the first command stopped or cleared in step 1) and the second command terminate abnormally. This status is sent only once for both commands.

3) Sense data generated by the IDD at this time indicates ABORTED COMMAND [=B]/Overlapped Commands Attempted [=4E-00].

Notes:

1. The INIT can be allowed to send the ABORT or BUS DEVICE RESET message with specifying that logical unit to stop the command processing during disconnection (see Subsection 1.7.6).

2. The overlap command rule does not depend on the type of the second command. For example, the IDD sends the CHECK CONDITION status and makes both commands terminate abnormally even if the second command is the INQUIRY, REQUEST SENSE, or PRIORITY RESERVE command.

1.7.2 Illegal LUN specification

The IDD supports only one logical unit number, LUN = 0. When other LUNs (1 to 7) are specified, the IDD operates one of the following ways depending on the type of command issued.

1) The IDD executes the INQUIRY command successfully even if an illegal LUN is specified. Byte 0 ("qualifier" and "device type code" fields) of the standard INQUIRY data which the command transfers to the INIT indicates X'7F'.

2) The IDD executes the REQUEST SENSE command successfully even if an illegal LUN is specified. However, an illegal LUN specification (ILLEGAL REQUEST [= 5]/Logical unit not supported [= 25-00]) is indicated in the sense data which the command transfers to the INIT.

3) When an illegal LUN is specified with commands other than the INQUIRY and REQUEST SENSE commands, the command terminates abnormally with the CHECK CONDITION status. Sense data generated at this time indicates an illegal LUN specification (ILLEGAL REQUEST [= 5]/Logical unit not supported [= 25-00]).

1.7.3 Operation codes being reserved

The operation codes (X'E0' to X'FF') in the group 7 command are reserved by Fujitsu. When the command having these operation codes is issued, the IDD does not always post the CHECK CONDITION status (ILLEGAL REQUEST [= 5]/Invalid command operation code [= 20-00]).

The INIT must not issue the command having these operation codes.

1.7.4 Command processing in not ready state

When the initial self-diagnostics is terminated successfully and spindle motor rotation speed becomes stable after power-on, the IDD reads system information such as the MODE SELECT parameter and defect infor-mation on the disk from the system space in the disk drive and initializes various control information. The IDD becomes usable (ready) after completing this operation. Here, the "not ready state" indicates the following:

Not ready state:

- The spindle motor speed has not become regular.
- Reading of system information is not completed or fails.

Also, accessing data on the disk is disabled until disk initialization (formatting) is completed successfully.

This subsection explains the processing and response in the IDD for commands received while the IDD is in the not ready state or disk initialization is not completed. For spindle motor start control at power-on, use the setting terminal (motor start mode) on the IDD, either starting the rotation at the same time as power-on or controlling the rotation using the START/STOP UNIT command.

(1) General response in not ready state

When receiving a command other than those in item (2) below, the IDD sends the CHECK CONDITION status for the command. Sense data generated at this time indicates the sense key and code depending on the current IDD state as listed in Table 1.3.

Table 1.3 Sense data in not ready state

IDD state	Sense key	Sense code
Rotation speed of spindle motor does not reach the stable speed.	NOT READY [=2]	Logical unit not ready [=04-00]
System information is not completely read out.	NOT READY [=2]	Logical unit has not self-configured yet [=3E-00]
Reading of system information is failed.	HARDWARE ERROR [=4]	Logical unit failed self-configuration [=4C-nn]

(2) Commands that can be executed in not ready state

The IDD can execute the following commands even if they are received in the not ready state.

a. START/STOP UNIT command

b. PRIORITY RESERVE command

c. RESERVE UNIT command

d. RELEASE UNIT command

e. INQUIRY command

This command is successfully executed and the standard INQUIRY data or the VPD information transferred to the INIT at this time indicates X'00' in byte 0 ("qualifier" and "device type code" fields). However, the value of some fields depends on whether system information is completely read. See the explanation of the INQUIRY command (Subsection 3.1.2) for details.

f. REQUEST SENSE command

This command is successfully executed and pending sense data is transferred to the INIT while sense data is held. While no sense data is held, sense data according to the current IDD state (see Table 1.3) is transferred to the INIT.

g. REZERO UNIT, RECOVER ID and RECOVER DATA commands

These commands are successfully executed when reading of system information is failed but the spindle motor rotates at the stable speed in the disk drive. When the spindle motor speed is less than the stable speed or the system information is being read, the CHECK CONDITION status is sent in the same way as in item (1).

(3) Operation when formatting is not completed successfully

After the MODE SELECT command is executed to change the parameters related to the data format when the FORMAT UNIT command is not executed yet or the FORMAT UNIT command terminates abnormally with some reason (example: power-off or RESET condition), data on the disk cannot be accessed normally. In such a case, the IDD sends the CHECK CONDITION status for commands that access the disk.

Sense data generated at this time indicates MEDIUM ERROR [=3]/Medium format corrupted [=31-00] in the former case or MEDIUM ERROR [=3]/Format command failed [=31-01] in the latter case. In both cases, the IDD becomes usable when the FORMAT UNIT command is executed to reinitialize the disk.

1.7.5 Error recovery

If a recoverable error is detected on the SCSI bus or inside the IDD, the IDD tries to perform error recovery during the execution of a command.

The INIT can use the MODE SELECT command to specify detail parameters related to error recovery and whether it is necessary to report an error when recovery from it succeeds. Different error recovery parameter values can be used for different INITs because the IDD holds the MODE SELECT parameter individually for each INIT. The parameters can be saved in the system space on the disk and temporarily changed without being saved depending on the INIT specification. The IDD reads saved parameter values and initializes the error recovery procedure after the power-on or reset operation (RESET condition or BUS DEVICE RESET message).

See Subsection 3.1.4 for details on the MODE SELECT command, and see Section 4.4 and refer to Chapter 3 of OEM Manual SCSI Physical Specifications for details on error recovery process.

The error recovery executed by the IDD is outlined below.

(1) Recovery from SCSI bus error

The IDD tries to perform error recovery for errors detected on the SCSI bus as listed in Table 1.4. The INIT must support the RESTORE POINTERS message to enable all of the retry operations. If recovery from an error is disabled or a serious error related to the SCSI bus protocol is detected, the IDD must enter the SCSI bus phase to BUS FREE forcibly and clear the command that is being executed.

Table 1.4 Outline of recovery from SCSI bus error

Item	Error type	Error recovery
1	MESSAGE OUT phase parity error	Message re-reception
2	COMMAND phase parity error	COMMAND phase reexecution
3	DATA OUT phase parity error	DATA OUT phase reexecution
4	INITIATOR DETECTED ERROR message reception	Reexecution of current bus phase
5	MESSAGE PARITY ERROR message reception	Message retransmission
6	RESELECTION phase timeout	RESELECTION phase reexecution

(2) Recovery from disk drive error

The IDD executes error recovery listed in Table 1.5 for an error detected during access to data on the disk. Note that the recovery increases command processing time because of rotation delay for repositioning to the data block on the disk or initialization of the positioning control system.

The INIT can use the MODE SELECT parameter page 1 (read/write error recovery parameter), page 7 (verify error recovery parameter), and page 21 (additional error recovery parameter) to control the retry count. However, use the default value specified by the IDD in general operation.

Table 1.5 Outline of recovery from disk drive error

Item	Error type	Error recovery
1	Seek error	Return to max cylinder, readjustment of positioning control system, and repositioning to data block
2	ID field read error	Reread
3	Uncorrectable error in data field	Reread
4	Correctable error in data field	Correction with ECC or reread (They can be selected using MODE SELECT parameter.)
5	Undetected object data block	Reread

(3) Automatic readjustment of the positioning control system

To obtain the highest read/write head positioning accuracy regardless of the environmental condition such as temperature, the IDD automatically readjusts the positioning control system at the time intervals listed below after power-on:

- 1st Power-On
- 2nd 5 minutes after power-on
- 3rd 10 minutes after power-on
- 4th 15 minutes after power-on
- 5th 20 minutes after power-on
- 6th 25 minutes after power-on
- 7th 30 minutes after power-on
- 8th 40 minutes after power-on
- 9th 50 minutes after power-on
- 10th 60 minutes after power-on
- 11th 90 minutes after power-on
- Later every 30 minutes

The mean adjustment time is about 100 millisecond but it varies depending on disk drive state.

The readjustment is executed while no command is stacked or being executed in the IDD. However, when a command is issued immediately after readjustment starts and the command requires to access data on the disk, execution of the command cannot start until the readjustment is completed.

The IDD executes readjustment regardless of the above interval condition when the REZERO UNIT command is executed or execution of the FORMAT UNIT command starts. In addition, the IDD may execute readjustment during recovery from a disk drive error as explained in Section 4.3.

Note:

> The INIT cannot estimate the readjustment start timing generally. Readjustment is executed while no command is stacked or being executed but when a command is issued immediately after readjustment starts and the command requires to access data on the disk, execution of the command enters wait state and the command cannot be executed until the readjustment is completed.
>
> The INIT can use the MODE SELECT parameter page 21 (additional error recovery parameter) to inhibit this command execution wait. When command execution wait is inhibited and a received command cannot be executed immediately because of readjustment, the IDD terminates the command with the CHECK CONDITION status. Sense data generated at this time indicates ABORTED COMMAND [=B]/Command execution delay required [=80-00].

However, when consideration is taken for error recovery that possibly occurs while execution wait state is set with the command stack feature or a command is being executed, time before execution of a command issued by the INIT starts and time before the execution is completed are undefined. The command execution start delay caused by readjustment can be contained in these undefined elements. Generally, command execution wait need not be inhibited using the MODE SELECT parameter.

1.7.6 Reset processing

The INIT can execute the following reset processing methods on the SCSI bus:

- RESET condition
- BUS DEVICE RESET message
- ABORT message

When an INIT reserving the IDD enters unrecoverable state under the multiple initiator environment, the PRIORITY RESERVE command can be used to get back the IDD access authority forcibly from the INIT.

The RESET condition resets all SCSI devices connected to the same SCSI bus and sets the IDD to the same state as that at power-on ("hard" RESET).

The BUS DEVICE RESET message can reset only SCSI devices (TARG) selected by an INIT and set the TARGs to the same state as that at power-on. However, note that commands issued by other INITs are also cleared at this time under the multiple initiator environment.

When a specific command issued by the INIT is to be cleared, the INIT must use the ABORT message. When the command is being executed, the INIT generates the ATTENTION condition at optional time and sends the ABORT message to the TARG. When the command is disconnected, the INIT selects a TARG, and then sends the ABORT message following the IDENTIFY message where the object logical unit is specified or sends the ABORT message after a reconnection request is issued.

The IDD clears all commands that are being executed and are stacked when receiving the BUS DEVICE RESET message, the RESET condition occurs, or the PRIORITY RESERVE command is issued. When the ABORT message is issued, the IDD clears only commands (that are stacked or being executed) previously issued for the logical unit specified by an INIT when the INIT issues the ABORT message but no other command is influenced by the ABORT message. No status byte and COMMAND COMPLETE message are reported for cleared commands.

When clearing a command that is writing data onto the disk, the IDD discontinues the command processing in the ways listed in Table 1.6. This applies to an overlap command (see Subsection 1.7.1) that is being executed. The INIT must check the command termination state and if necessary, must execute data recovery.

Table 1.6 Reset processing during writing

Command type	Stop processing of command execution
WRITE WRITE EXTENDED WRITE AND VERIFY SEND DIAGNOSTIC (Write/read test) WRITE LONG WRITE SAME	The data block where data is being written is successfully processed including the ECC field, and then the command execution is stopped. Not all data items transferred from the INIT to IDD may be written onto the disk.
FORMAT UNIT	The data block where data is being written (initialization) is successfully processed, and then the command execution is stopped. The INIT must reissue the command because the formatting results of the whole disk are not guaranteed.
REASSIGN BLOCKS	The current allocation of an alternate block is completed, and then the command execution is stopped. Not all alternate blocks specified by the INIT may be allocated.
MODE SELECT MODE SELECT EXTENDED (Parameter save specification)	The command is completed when the parameter save operation is already started. However, the INIT must execute the MODE SENSE command to check the state or reissue the command because the INIT cannot check whether the save operation is executed.

1.7.7 Fatal hardware error

(1) Self-diagnostic error

When the fatal hardware error is detected during the initial self-diagnostics, offline self-diagnostics, or online self-diagnostics (SEND DIAGNOSTIC command), the IDD indicates the error by blinking the LED on the front panel and stops the spindle motor.

In this error state, the IDD posts the CHECK CONDITION status for all I/O operation requests other than the REQUEST SENSE command. The sense data indicates "HARDWARE ERROR [=4]/Diagnostic failure on component nn [=40-nn] ". The INIT must execute the error recovery by generating the RESET condition or by sending the BUS DEVICE RESET message. For the recommended error recovery procedure, see Section 4.3. For details of the self-diagnostics, refer to Subsection 6.1.1 in OEM Manual Specifications & Installation.

(2) Unrecoverable hardware error

When the fatal hardware error occurs during command execution and the termination process, such as posting the CHECK CONDITION status, cannot be executed, the IDD blinks the LED on the front panel and may stop the spindle motor.

After this state occurs, the IDD posts the CHECK CONDITION status for all I/O operation requests other than the REQUEST SENSE command. The sense data indicates "HARDWARE ERROR [=4]/Internal Target failure [=44-nn]". When this CHECK CONDITION status is posted for the I/O operation request continuously, the INIT must execute the error recovery by generating the RESET condition or by sending the BUS DEVICE RESET message. For the recommended error recovery procedure, see Section 4.3.

1.8 Data Block Addressing

1.8.1 Data space definition

The IDD manages the data storage area on the disk drive by dividing it to the following three data spaces:

- User space: User data storage area
- CE space: Area reserved for diagnosis
- System space: Area only for the IDD

The user can access only the user space and CE space explicitly. The same data format and defect management method are used for these two spaces and the logical data block addressing explained in Subsection 1.8.2 can be used to access these spaces. The system space is accessed at power-on or during the specific command execution in the IDD internally, but the user cannot directly access the system space.

Figure 1.4 shows the data space configuration on the disk drive. The user can specify (MODE SELECT command) up to 1429 cylinders (cylinders 0 to 1428) to be allocated to the user space.

The spare sector area (alternate area) for defective sectors is allocated to the user space and CE space. The user can specify (with MODE SELECT command) several sectors on the last track in each cylinder and several cylinders (alternative cylinders) in the user space as an alternate area.

Alternative data blocks can be allocated in units of defective sectors by the defect management in the IDD for defective sectors on the disk. With this defect management, the IDD can assume all logical data blocks in the user and CE spaces to be error-free blocks and access them.

For details on the data format (cylinder configuration, track format, and sector format) on the disk, defect management method, and alternate block allocation method, refer to Chapter 3 in OEM Manual Specifications & Installation. For outline of the management method, see Chapter 5.

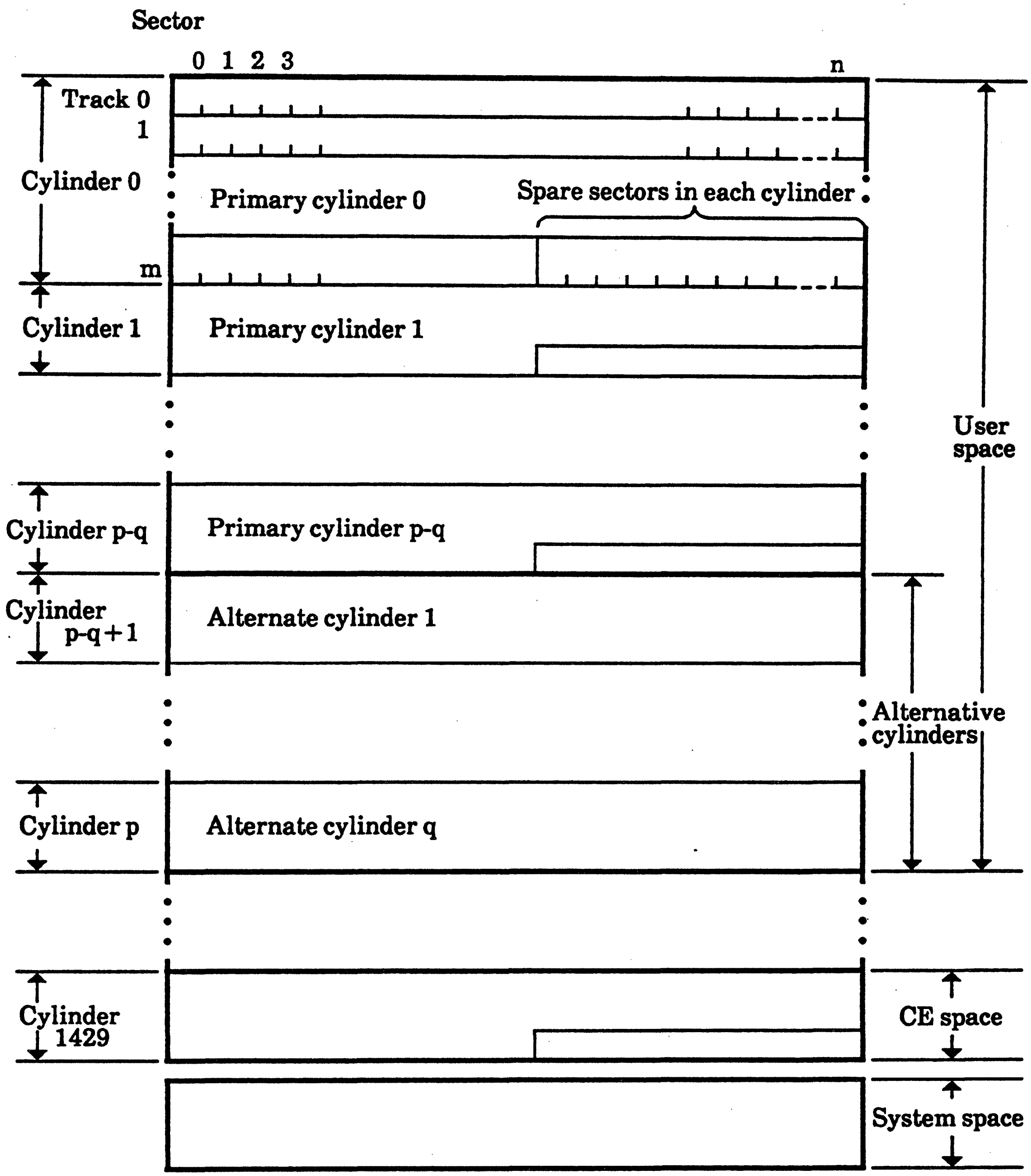


Figure 1.4 Data space configuration

1.8.2 Logical block addressing

The IDD uses the logical data block addressing not depending on the physical disk drive structure as disk data access method. The IDD makes each physical sector correspond to logical block addresses during formatting. Data on the disk is accessed in units of logical data blocks and the INIT uses the logical data block address to specify a target of access.

The logical data block addressing function uses continuous binary numbers for one drive to specify a data block address. Different methods are used to specify the logical data block address between the user space and CE space as explained below. The INIT can use the READ CAPACITY command to recognize the range of logical data block addresses that can be specified in the user and CE spaces.

(1) Logical block address in the user space

'0' is assigned to the logical data block address of the first data block in the user space and continuous logical data block addresses are allocated to data blocks up to the last data block in the user space.

The IDD assumes cylinder 0, track 0, sector 0 in the first logical data block, and then allocates logical data blocks in ascending address order in the following sequence:

① The IDD allocates the logical data blocks in sector number ascending order on the same track.

② The IDD allocates the subsequent logical data blocks to sectors in tracks other than the final one in track number ascending order in the same cylinder in the same way as that in step ①.

③ The IDD allocates the subsequent logical data blocks to sectors other than spare sectors on the last track in the same cylinder in the same way as that in step ①.

④ The IDD allocates the subsequent logical data blocks to cylinders other than the alternative ones up to the final one (cylinder p-q in Figure 1.4) in the cylinder number ascending order in the user space in the same way as that in steps ① to ③.

(2) Logical block address in the CE space

The most significant bit is assumed to be 1 in the 32-bit logical data block address and the logical data block addresses of the CE space data blocks are indicated using unsigned continuous binary numbers starting with X'80000000' (the second number is X'80000001' and the third number is X'80000002'). Therefore, group 0 commands (21-bit addressing only is enabled) cannot be used to access the CE space logical data block.

The data blocks are allocated in the same order as that in the user space. However, there is only one cylinder in the CE space and the final sector on the final track in the cylinder is the final data block in the CE space except for spare sectors.

(3) Alternate area

The above logical data block address is not allocated to any alternative areas (spare sectors in each cylinder and alternate cylinders) in the user and CE spaces. Sectors allocated as alternate blocks in an alternative area are automatically accessed by the IDD defect management (sector slip processing or alternate block processing). Therefore, the user need not consider access to the alternate area and cannot direct-access the data block on the alternate area explicitly.

CHAPTER 2 DATA BUFFER MANAGEMENT

2.1 Data Buffer
2.2 Read-Ahead Cache Feature
2.3 Write Cache Feature

This chapter describes the configuration and operation of the data buffer and the operation of the cache feature.

2.1 Data Buffer

2.1.1 Data buffer configuration and basic operation

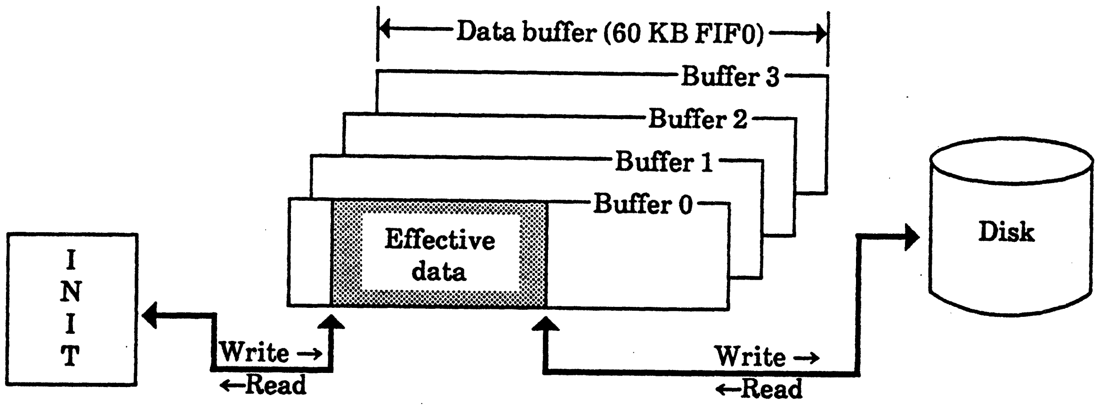
The IDD provides four 60-KB data buffers and they make data transfer efficient between the INIT and the disk drive.

The each data buffer is the First-In-First-Out (FIFO) ring buffer. The data buffer contains two ports. One is used for data transfer with a disk and the other is used for data transfer with the SCSI bus (INIT). These ports operate asynchronously of each other. Since the data buffer can efficiently make up for data transfer speed differences between these ports, the INIT can transfer data with IDD without considering the difference of data transfer speed between the SCSI bus and the disk drive.

The data buffer can contain data consisting of one or more tracks from the disk. The data buffer can transfer data under the best conditions for the system without using the sector interleave even if the SCSI bus (INIT) is inferior to the disk drive in data transfer speed. When the SCSI bus (INIT) is superior to the disk drive in data transfer speed, the SCSI bus data transfer exclusive time can be minimized with making an appropriate amount of data is saved. When one command is executed, one of four data buffers is used and data in the remaining three data buffers are kept. The data buffer to be used is decided by the LRU algorithm. Figure 2.1 shows the data buffer.

Note:

The capacity of each data buffer is 60 KB. However, maximum 56-KB data can be stored in the data buffer.



(When data buffer 0 is selected)

Figure 2.1 Data buffer configuration

The basic function and operation of the data buffer in read/write processing are explained below.

(1) Read operation

Data read from the disk is transferred to the SCSI bus at the timing specified by the MODE SELECT parameter after the data is saved temporarily in the data buffer. The basic procedures of the read operation are explained below.

① The IDD disconnects the SCSI bus and positioning to the data block on the disk specified by the command when receiving a command.

② The IDD reads data from the disk and writes in the data buffer after the positioning to the target data block is completed. If a correctable error is detected, the error recovery or data correction is performed in the data buffer according to the MODE SELECT parameter specification.

③ When data blocks specified by the buffer full ratio (see Subsection 2.1.2) in the MODE SELECT parameter are written in the data buffer, the IDD reconnects the SCSI bus while reading subsequent data blocks from the disk in the data buffer in order to transfer the data from the data buffer to the INIT (SCSI bus).

④ When the INIT (SCSI bus) transfers the data quickly and the data buffer becomes empty before the data blocks specified by the command are transferred, the IDD disconnects the SCSI bus at that time. The IDD repeats reconnection, data transfer, and disconnection until the data blocks specified by the command are transferred.

⑤ When data blocks specified by the command are greater than the maximum capacity can be read in the data buffer (56 KB) and the INIT is inferior to the disk drive in data transfer speed, the free area runs short in the data buffer after the procedure in step ③ and data may not be able to be read from the disk (data overrun). In this case, the IDD reaccesses the overrun block after one rotation of the disk and continues reading.

⑥ The IDD reports status and terminates the command after all specified data blocks are transferred.

Note:

In order to prevent disconnection/reconnection from being repeated frequently and to prevent data overrun from occurring after data is started to be transferred in the SCSI bus, "buffer full ratio" in the MODE SELECT parameter (see Subsection 2.1.2) is required to be set to balance the difference in the data transfer speed between the INIT (SCSI bus) and the disk drive.

Figure 2.2 shows the operation state of the data buffer in read processing. For details, see Subsection 2.1.2.

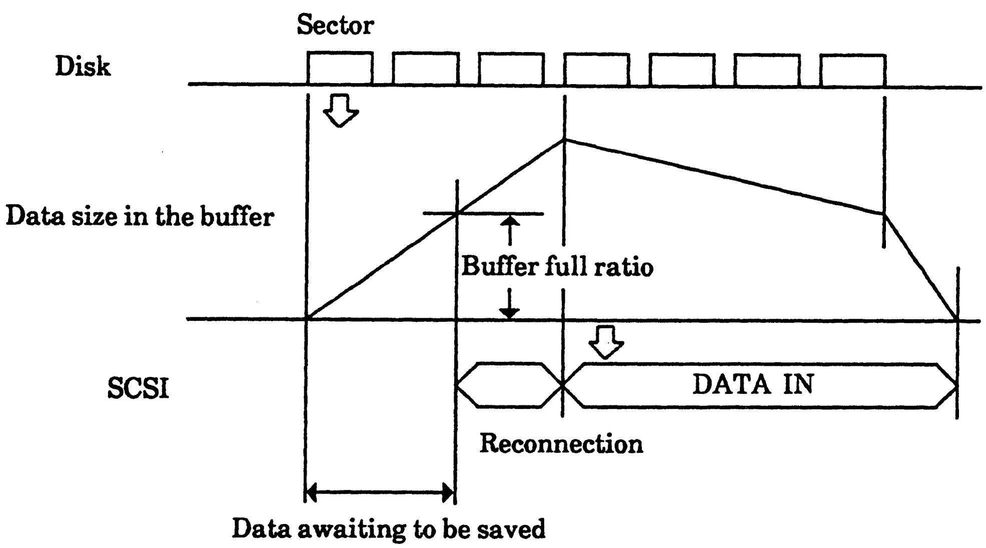


Figure 2.2 Data buffer operation in read processing

(2) Write operation

Data transferred from the INIT is written on the disk after being saved in the data buffer temporarily. The basic procedures of the write operation are explained below.

① The IDD enters the DATA OUT phase immediately after receiving a command, and saves transferred data from the INIT in the data buffer (data prefetching). At this time, the IDD points to a data block on the disk specified by the command in parallel. The IDD disconnects the SCSI bus when all data blocks specified by the command are saved in the data buffer, or when there is no area in the data buffer because of the large amount of specified data.

② The IDD writes the data in the disk from the data buffer after pointing the data block on the disk.

When the data block is pointed during data prefetching, writing data on the disk and prefetching data in the data buffer are executed in parallel.

③ When the data transfer speed of the disk drive is faster than that of the INIT (SCSI bus), the data may not be able to be written continuously (data underrun) because the data to be written in the data block is not prefetched completely at writing the data on the disk. The IDD repoints to a data block in which data underrun occurred after one rotation of the disk while continuing to prefetch the data from the INIT, and continues writing the data block.

④ When the IDD has not finished transferring all data blocks specified by the command from the INIT after disconnecting the SCSI bus, the IDD reconnects the SCSI bus when free areas of the data buffer equal the data with the size specified by the buffer empty ratio of the MODE SELECT parameter as the data is written on the disk, and then the IDD starts to transfer the subsequent data (data fetching).

The data is written on the disk along with data prefetching. The data is transferred with the INIT until free areas of the data buffer are filled with the data. The IDD repeats disconnection, reconnection, and data transfer until all data blocks specified by the command are transferred completely.

⑤ The IDD reports the status and terminates the command after the all specified data blocks are transferred completely.

Notes:

1. When the data transfer size specified by the command is 56 KB or less, the buffer empty ratio value has no meaning because all data blocks required for command execution are prefetched together.

2. In order to process data blocks exceeding 56 KB using a command, the buffer empty ratio of the MODE SELECT parameter must be set to balance the difference in data transfer speed between the INIT (SCSI bus) and the disk drive to prevent disconnection/reconnection from being repeated frequently during command execution, or to prevent data underrun from occurring (see Subsection 2.1.2).

Figure 2.3 shows the operation state of the data buffer in write processing. For details, see Subsection 2.1.2.

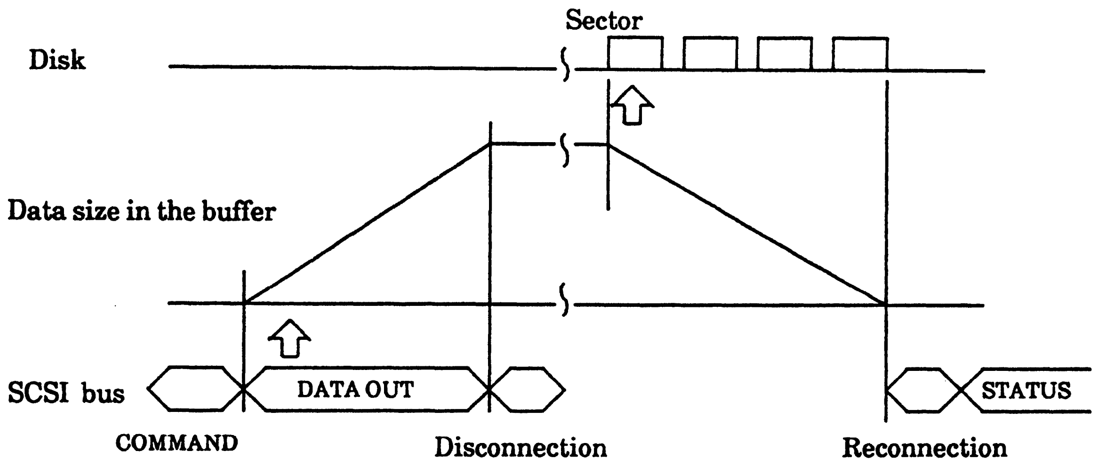


Figure 2.3 Data buffer operation in write processing

2.1.2 Operation mode control

(1) MODE SELECT parameter

The MODE SELECT parameter (disconnect/reconnect parameter: page code 2) shown in Figure 2.4 is provided in the IDD so that the INIT controls the timing to start reconnection that transfers data with the SCSI bus. The user can set the appropriate operation state for the system environment. Since the IDD has different MODE SELECT parameter values for each SCSI ID of the INIT, each INIT can have different parameter values. For the SELECT MODE command, see Subsection 3.1.4.

Disconnect/reconnect parameter (page code 2)

	Bit							
	7	6	5	4	3	2	1	0
Byte 0	0	0	0	0	0	0	1	0
1	X'0E' (page length)							
2	Buffer full ratio							
	Default value: X'00' (=1 logical data block) Variable range: X'00' to X'FF'							
3	Buffer empty ratio							
	Default value: X'80' (=28 KB) Variable range: X'00' to X'FF'							
4								
~	~							
15								

Figure 2.4 Reconnection timing control parameter

a. Buffer full ratio

This parameter specifies the timing the IDD starts reconnection in order to transfer data to the INIT by the READ or READ EXTENDED command.

This parameter value (n) specifies "the amount of the data to be read from the disk to the data buffer" with a ratio of it to "the total capacity of the data buffer" (n/256). The maximum capacity can be read in the data buffer of the IDD is 56 KB. Therefore, when "n" is 32, this parameter specifies 7 KB of data (see Note).

When the data with the size specified with the parameter is enabled to be transferred from the data buffer to the INIT, the IDD executes reconnection and starts transferring the data to the INIT. However, if the total transfer data size specified by the command is less than the data buffer size (56 KB), the IDD executes reconnection and starts transferring the data to the INIT when at least the data blocks with the ratio specified with the parameter in the total transfer data block size specified with the command are enabled to be transferred to the INIT. For example, when sixteen 512-byte logical data blocks (8 KB) are requested to be read and 32 is set to the parameter, two data blocks [16 × (32/256)] are enabled to be transferred and the IDD executes reconnection.

b. Buffer empty ratio

This parameter specifies the timing the IDD starts reconnection in order to request data transfer to the INIT by the WRITE, WRITE EXTENDED, or WRITE AND VERIFY command. The IDD executes first data transfer (data prefetching) along with pointing to a data block to be written on the disk after receiving the command. When the total number of bytes of the data specified by the command is less than the data buffer (56 KB), all data blocks are prefetched together. This parameter specification is not applied.

This parameter value (n) specifies "the size of the free area in the data buffer of the IDD" with a ratio of it to "the total capacity of the data buffer" (n/256). Therefore, when "n" is 32, this parameter specifies 8 KB as the size of the free area on the buffer (see Note).

The IDD uses the data prefetched in the data buffer in order along with writing the data in the disk. When the data in the free data buffer area equals the data with the size specified by the parameter, the IDD executes reconnection and requests the transfer of the subsequent data. If the number of remaining data blocks to be transferred by the command being executed does not reach the data size specified by the parameter, the IDD executes reconnection and requests the transfer of the subsequent data when the data in the free data buffer area equals the number of remaining data blocks.

Note:

When the buffer full ratio or buffer empty ratio specification value is not close to the integral boundary of the logical data block size, the IDD assumes that the data block boundary that is rounded up the specified value is specified. When zero is specified in one of these parameters, the IDD assumes that a logical data block is specified in the parameter.

2.2.2 Caching parameter

The IDD supports the MODE SELECT parameter (caching parameter: page code =8) shown in Figure 2.8 to manage the cache feature. See Subsection 3.1.4 for details of the MODE SELECT parameter.

[Caching parameter (page code = 8)]

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	0	0	0	0	1	0	0	0
	1	X'0A' (page length)							
	2	(Reserved)					WCE	MS	RCD
	Default value	0	0	0	0	0	0	0	0
	Variable value	0	0	0	0	0	1	0	1
	3	X'00' (Reserved)							
	4−5	Number of prefetch suppression blocks							
	Default value	X'FFFF'							
	Variable value	X'0000'							
	6−7	Minimum prefetch size							
	Default value	X'0000'							
	Variable value	X'0000'							
	8−9	Maximum prefetch size							
	Default value	X'0xxx' (60 KB)							
	Variable value	X'0000'							
	10−11	Number of maximum prefetch suppression blocks							
	Default value	X'FFFF'							
	Variable value	X'0000'							

Note:

The variable value column indicates that the parameter can be changed ("1" means changeable).

Figure 2.5 Caching control parameter

(1) Read cache disable (RCD)

This bit specifies enable/disable of the Read-Ahead cache feature.

When 1 is specified in this bit, the Read-Ahead cache feature operation is disabled. The IDD reads all data requested by the READ or READ EXTENDED command from the disk and transfers them to the INIT. The subsequent data blocks are not prefetched.

When 0 is specified in this bit, the Read-Ahead cache feature operation is enabled. The IDD executes the caching operation described before by the READ or READ EXTENDED command .

(2) Multiple selection (MS)

This bit indicates the specifying method of the minimum prefetch (bytes 6 and 7) and maximum prefetch (bytes 8 and 9) parameters.

The IDD supports only "0", and the minimum prefetch and maximum prefetch parameters indicate the number of the logical data blocks to be prefetched.

(3) Number of prefetch suppression blocks

This parameter is used to selectively disable the data prefetching by the READ or READ EXTENDED command. This parameter cannot be changed in the IDD. The IDD operates with the default value (X'FFFF').

(4) Minimum prefetch size (bytes 6 and 7)

This parameter specifies the minimum size (number of logical data blocks) of the data blocks to be prefetched in the data buffer by the READ or READ EXTENDED command. This parameter cannot be changed in the IDD. The IDD operates with the default value (X'0000').

(5) Maximum prefetch size (bytes 8, 9, 10 and 11)

This parameter specifies the maximum size (number of logical data blocks) of the data blocks to be prefetched in the data buffer by the READ or READ EXTENDED command. This size is 60 KB and cannot be changed in the IDD.

2.2.3 Prefetching operation and prefetching amount

① Prefetching is executed regardless of the track or cylinder boundary except for the conditions ③ and ④. When a command is stacked or a new command is received during prefetching, the prefetching operation is immediately stopped except for the following cases.

- The command to be executed next is not needed to stack.
- The command to be executed next is the READ or READ EXTENDED command and the first block specified by that command is already prefetched in the data buffer with the current prefetch operation. Or, all blocks specified by that command exists in the data buffer as a caching data.

② When the data amount specified by the READ or READ EXTENDED command exceeds 30 KB, the maximum prefetching amount is set to 60 KB automatically and data specified by the READ or READ EXTENDED command does not remain is the data buffer.

When the data amount specified by the READ or READ EXTENDED command does not exceed 30 KB, the maximum prefetching amount is calculated by subtracting the data amount specified by the READ or READ EXTENDED command from 60 KB. In this case, data specified by the READ or READ EXTEND command remains in the data buffer and is to be cached.

③ When an error that needs retry occurs during data prefetching, data prefetching is terminated at that time (data correction by the ECC is not performed). For the defective/alternative block processing, data prefetching is continued even if the sector slip processing or alternative sector processing is applied.

④ When the RESET condition occurs on the SCSI bus or when an INIT issues the BUS DEVICE RESET message or PRIORITY RESERVE command, data prefetching is terminated forcededly and the caching data in the data buffer is invalidated.

2.3 Write Cache Feature

The IDD has a write cache feature to reduce the command processing time for the INIT. When the write caching is enabled by the MODE SELECT command (caching parameter: page code = 8, WCE bit), the IDD reports the GOOD status and terminates the command processing at completion of transferring all data specified by the WRITE or WRITE EXTENDED command.

Note:

When the write caching is enabled, the transferred data from the INIT by the WRITE or WRITE EXTENDED command is written on the disk after reporting the GOOD status. Therefore, when an unrecoverable write error occurs at writing, a sense data is generated. When the sense data is kept, the IDD reports the CHECK status for next command generally to report that the sense data is kept. However, it is generally difficult for the INIT to retry the unrecoverable write error in write caching. Take care at using this write cache feature.

This page is intentionally left blank.

CHAPTER 3 SPECIFICATIONS OF COMMANDS

3.1 Control/Sense Commands

3.2 Data Access Commands

3.3 Format Commands

3.4 Maintenance and Diagnositic Commands

3.1 Control/Sense Commands

3.1.1 TEST UNIT READY

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'00'							
	1	LUN			0	0	0	0	0
	2	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	Flag	Link

The TEST UNIT READY command checks the status of the disk drive. This command is not stacked, but is immediately executed.

When the IDD is in ready state and can be used by the INIT that issued this command, the GOOD status is reported to this command.

On the other hand, when the IDD cannot be used, the CHECK CONDITION status is reported to this command. The sense data generated at this time indicates the status of the IDD held at this time.

3.1.2 INQUIRY

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'12'							
	1	LUN			0	0	0	0	EVPD
	2	Page code							
	3	0	0	0	0	0	0	0	0
	4	Transfer byte-length							
	5	0	0	0	0	0	0	Flag	Link

The INQUIRY command transfers the INQUIRY data (the information that indicates the characteristics of the IDD) to the INIT. This command is not stacked, but is immediately executed.

Even if a UNIT ATTENTION condition is being held, this command is executed successfully and the UNIT ATTENTION condition is not cleared. Moreover, even if the disk drive is not ready or an invalid logical unit number (LUN) is specified, this command is executed successfully.

When one of the following occurs, the CHECK CONDITION status is reported to this command and an abnormal termination occurs.

- A specification error was detected in other than the "LUN" field of the CDB.
- The INQUIRY data could not be sent because of an IDD hardware error occurrence.
- An unrecoverable error was detected on the SCSI bus.
- The error detected during command execution was recovered by the IDD error recovery processing (retry), but the mode for RECOVERED ERROR notification was being specified in the MODE SELECT parameter.
- The exception conditions of the overlap command are applied (see Subsection 1.7.1).

The command functions specifiable by the CDB and the data contents to be transferred by this command from the IDD to the INIT depend on the specification of the IDD setting terminal (SCSI level). For setting terminal, refer to Subsection 5.3.2 of <u>OEM Manual Specifications & Installation</u>.

The Enable Vital Product Data (EVPD) bit (bit 0 in CDB byte 1) and the "page code" field (byte 2) are valid only when the setting terminal is set so that INQUIRY data in SCSI-2 mode is to be transferred. When the setting terminal is not set in the above way, the EVPD bit and the "page code" field must be set to 0.

When the EVPD bit is 0, the IDD transfers the standard INQUIRY data to the INIT. When the EVPD bit is 1, the Vital Product Data (VPD) is transferred to the INIT.

When the EVPD bit is 1, the "page code" field specifies the type of VPD to be transferred to the INIT. When the EVPD bit is 0, 0 must be specified in the "page code" field.

The "transfer byte-length" field (CDB byte 4) indicates the byte-length of the standard INQUIRY data or the VPD that can be received by the INIT by using this command. The length of the data to be transferred by the IDD to the INIT is equal to the shorter one of the "transfer byte-length" field value or the length of the standard INQUIRY data or the VPD of the IDD. When 0 is specified in the "transfer byte-length" field, this command is terminated successfully without transferring any data.

(1) Standard INQUIRY data

Figure 3.1 shows the format and contents of the standard INQUIRY data that is to be transferred by this command to the INIT when 0 is specified in the EVPD bit.

<table>
<tr><td></td><td colspan="8">Bit</td><td></td></tr>
<tr><td></td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td></td></tr>
<tr><td rowspan="2">Byte 0</td><td colspan="3" rowspan="2">Qualifier</td><td colspan="5">Device type code</td><td></td></tr>
<tr><td colspan="5">(0,0,0,0,0) or (1,1,1,1,1)</td><td></td></tr>
<tr><td rowspan="2">1</td><td>RMB</td><td colspan="7">Device type qualifier</td><td></td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td></td></tr>
<tr><td rowspan="3">2</td><td colspan="2">ISO version</td><td colspan="3">ECMA version</td><td colspan="3">ANSI version</td><td></td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>(*1)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>(*2)</td></tr>
<tr><td rowspan="3">3</td><td>AENC</td><td>TrmIOP</td><td rowspan="3">0</td><td rowspan="3">0</td><td colspan="4">Response data format</td><td></td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>(*1)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>(*2)</td></tr>
<tr><td>4</td><td colspan="8">X'1F' (Added data length)</td><td></td></tr>
<tr><td>5</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td></td></tr>
<tr><td>6</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td></td></tr>
<tr><td rowspan="3">7</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>(*1)</td></tr>
<tr><td>RelAdr</td><td>WBus32</td><td>WBus16</td><td>Sync</td><td>Linked</td><td rowspan="2">0</td><td>CmdQue</td><td>SftRe</td><td rowspan="2">(*2)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>8
9
15</td><td colspan="8">FUJITSU
(Vendor ID: ASCII)</td><td></td></tr>
<tr><td>16
17
31</td><td colspan="8">M262※S-□□□□
(Product ID: ASCII)
(※: 2, 3 or 4, □: Logical block length)</td><td></td></tr>
<tr><td>32
33
34
35</td><td colspan="8">Microcode version
(Product Revision: ASCII)</td><td></td></tr>
</table>

*1 These values are set when the setting plugs are set so that INQUIRY data in SCSI-1/CCS mode is to be transferred.

*2 These values are set when the setting plugs are set so that INQUIRY data in SCSI-2 mode is to be transferred.

Figure 3.1 Standard INQUIRY data

a. Qualifier

(0,0,0): Indicates that the specified logical unit is an I/O device of the type specified in the "device type code" field. Even if this code is sent, the logical unit is not always ready.

(0,0,1): Indicates that the specified logical unit is an I/O device of the type specified in the "device type code" field, but the actual I/O device is not being connected as this logical unit. The IDD does not send this code.

(0,1,1): Indicates that the specified logical unit is not being supported. When this code is sent, the device type code field indicates X'1F'.

b. Device type code

(0,0,0,0,0): Direct access device

(1,1,1,1,1): Undefined device

When the LUN = 0 is specified, the IDD sends device type code "0,0,0,0,0" (direct access device). In this case, the qualifier field indicates "0,0,0", therefore, byte 0 indicates X'00'. On the other hand, when a value other than 0 is specified in the LUN field, the IDD sends device type code "1,1,1,1,1" (undefined device). In this case, the "qualifier" field indicates "0,1,1", therefore, byte 0 indicates X'7F'.

c. RMB bit

When this bit is 1, the storage medium can be replaced. Since the IDD uses a fixed disk, this bit is always 0.

d. Device type qualifier

0 is always sent to this field.

e. SCSI standard version

Byte 2 indicates the codes of the SCSI standard that apply to the IDD. At the ANSI standard application level, the IDD sends "0,0,1" or "0,1,0" according to the setting terminal specification. The following table lists the ANSI version code definitions.

Code	Definition
0 0 0	Device to which the standard before ANSI X3.131-1986 was applied
0 0 1	Device to which standard ANSI X3.131-1986 was applied
0 1 0	Device to which standard ANSI X3T9.2/86-109 (SCSI-2) was applied

f. Response data format

This field indicates the code that represents the INQUIRY data format. The following table lists the definitions of this code. The IDD sends "0,0,0,1" or "0,0,1,0" according to the setting terminal specification.

Response data format bit: 3 2 1 0	Standard INQUIRY data format
0 0 0 0	Format specified by ANSI X3.131-1986 (SCSI-1)
0 0 0 1	Format specified by ANSI X3T9.2/85-52 (CCS)
0 0 1 0	Format specified by ANSI X3T9.2/86-109 (SCSI-2)

g. Added data length

This field indicates the byte-length of byte 5 and later of the INQUIRY data. Independent of the "transfer byte-length" field specification of the CDB, this field indicates the INQUIRY data length defined by the IDD. This value is always X'1F', therefore, the total length of the INQUIRY data is 36 bytes.

h. Supported functions

Byte 3 bit 7 and byte 7 bits 1 to 7 are valid only when the setting terminal is set to transfer INQUIRY data in SCSI-2 mode. These bits indicate the functions supported in the IDD. A function is supported when the corresponding bit is 1, and is not supported when the corresponding bit is 0. When setting terminal is not set to transfer INQUIRY data in SCSI-2 mode, all of these bits are cleared to 0.

(a) Byte 3 bit 7

AENC: Asynchronous event notification capability (This bit is 0.)
TrmIOP: TERMINATE I/O PROCESS message (This bit is 0.)

(b) Byte 7 bits 1 to 7

RelAdr: Relative logical block addressing (This bit is 0.)
WBus32: 32-bit wide data transfer, that is, data transfer on a 4-byte bus (This bit is 0.)
WBus16: 16-bit wide data transfer, that is, data transfer on a 2-byte bus (This bit is 0.)
Sync: Synchronous-mode data transfer (This bit is 1.)
Linked: Command linking (This bit is 1.)
CmdQue: Command queuing with tag (This bit is 0.)
SftRe: 'Soft' RESET condition (This bit is 0.)

b. Buffer empty ratio

This parameter specifies the timing the IDD starts reconnection in order to request data transfer to the INIT by the WRITE, WRITE EXTENDED, or WRITE AND VERIFY command. The IDD executes first data transfer (data prefetching) along with pointing to a data block to be written on the disk after receiving the command. When the total number of bytes of the data specified by the command is less than the data buffer (56 KB), all data blocks are prefetched together. This parameter specification is not applied.

This parameter value (n) specifies "the size of the free area in the data buffer of the IDD" with a ratio of it to "the total capacity of the data buffer" (n/256). Therefore, when "n" is 32, this parameter specifies 8 KB as the size of the free area on the buffer (see Note).

The IDD uses the data prefetched in the data buffer in order along with writing the data in the disk. When the data in the free data buffer area equals the data with the size specified by the parameter, the IDD executes reconnection and requests the transfer of the subsequent data. If the number of remaining data blocks to be transferred by the command being executed does not reach the data size specified by the parameter, the IDD executes reconnection and requests the transfer of the subsequent data when the data in the free data buffer area equals the number of remaining data blocks.

Note:

When the buffer full ratio or buffer empty ratio specification value is not close to the integral boundary of the logical data block size, the IDD assumes that the data block boundary that is rounded up the specified value is specified. When zero is specified in one of these parameters, the IDD assumes that a logical data block is specified in the parameter.

2.2 Read-Ahead Cache Feature

The IDD provides a simple cache function called Read-Ahead cache feature to use the data buffer effectively and improve the disk drive effective access speed. The Read-Ahead cache feature is an effective cache function for the INIT to read data block groups on the disk in sequence using two or more commands.

2.2.1 Read-Ahead Caching operation

When the READ or READ EXTENDED command is executed, the IDD reads the requested data from the disk and transfers it to the INIT and the IDD reads prefetches the data block following the last logical data block specified by the command in the data buffer. When the READ or READ EXTENDED command issued later specifies one of the prefetched data blocks, the IDD transfers the data block directly from the data buffer to the INIT without accessing the disk. This feature can omit the mechanical access time when the continuous logical data block groups are to be read sequentially using two or more commands and can shorten the effective access time.

The INIT can inhibit the Read-Ahead cache feature operation by the mode setting (see Subsection 2.2.2).

(1) Commands to be cached

Caching operation is executed for the following commands:

- READ
- READ EXTENDED

When all data blocks to be cached and specified by these commands or some of the data blocks containing the first logical block specified by the commands correspond (hit) to the data (see item (2)) to be cached on the data buffer, the IDD outputs the transmittable data immediately from the data buffer to the INIT without disconnecting the SCSI bus after receiving the command.

When the first logical data block specified by the command does not correspond (miss) to the data to be cached on the data buffer, data is read from the disk.

(2) Data to be cached

The following data in the data buffer are to be cached described in item (1) (transferring the data in the data buffer to the INIT without accessing the disk).

① Data prefetched in the data buffer from the disk is to be cached after the READ or READ EXTENDED command. When the transferred data specified by the READ or READ EXTENDED command is less than 30 KB, data read by the READ or READ EXTENDED command is also to be cached.

② Data hit by the READ or READ EXTENDED command and transferred to the INIT is also to be cached even if the data is invalidated.

③ Data transferred to the disk from the INIT by a write command such as WRITE, WRITE EXTENDED, or WRITE AND VERIFY is not to be cached.

(3) Invalidating caching data

Data to be cached in the data buffer is invalidated in the following cases.

① When one of the following commands is issued to the same logical data block as the data to be cached, the data block is to be cached.

- WRITE
- WRITE AND VERIFY
- WRITE EXTENDED

② When one of the following commands is issued, all data items to be cached are invalidated.

- FORMAT UNIT
- MODE SELECT
- PRIORITY RESERVE
- READ BUFFER
- READ DEFECT DATA
- READ LONG
- REASSIGN BLOCKS
- RECEIVE DIAGNOSTIC RESULTS
- RECOVER DATA
- RECOVER ID
- SEND DIAGNOSTIC
- START/STOP UNIT
- WRITE BUFFER
- WRITE LONG
- WRITE SAME

③ When the data buffer having a data to be cached is used by one of the following commands, data to be cached is the specified data buffer is invalidated.

- READ
- READ EXTENDED
- VERIFY
- WRITE
- WRITE EXTENDED
- WRITE AND VERIFY

④ When one of the following events occurs, all data items to be cached are invalidated.

- The RESET condition occurs in the SCSI bus.
- The BUS DEVICE RESET message is issued from one of the INITs.

2.2.2 Caching parameter

The IDD supports the MODE SELECT parameter (caching parameter: page code =8) shown in Figure 2.8 to manage the cache feature. See Subsection 3.1.4 for details of the MODE SELECT parameter.

[Caching parameter (page code = 8)]

Byte		Bit 7	6	5	4	3	2	1	0
0		0	0	0	0	1	0	0	0
1		X'0A' (page length)							
2		(Reserved)					WCE	MS	RCD
	Default value	0	0	0	0	0	0	0	0
	Variable value	0	0	0	0	0	1	0	1
3		X'00' (Reserved)							
4−5		Number of prefetch suppression blocks							
	Default value	X'FFFF'							
	Variable value	X'0000'							
6−7		Minimum prefetch size							
	Default value	X'0000'							
	Variable value	X'0000'							
8−9		Maximum prefetch size							
	Default value	X'0xxx' (60 KB)							
	Variable value	X'0000'							
10−11		Number of maximum prefetch suppression blocks							
	Default value	X'FFFF'							
	Variable value	X'0000'							

Note:

The variable value column indicates that the parameter can be changed ("1" means changeable).

Figure 2.5 Caching control parameter

The "qualifier" field and the "device type code" field in byte 0 contain the same values as those in the standard INQUIRY data. The "page code" field in byte 1 indicates the identifier of this VPD information. The "page length" field in byte 3 indicates the byte-length of byte 4 and later of this data, without regard to the "transfer byte-length" field value of the CDB. This value is always X'02', therefore, the total data length is 6 bytes.

Bytes 4 and later contain a list of all the page codes supported by the IDD. page codes are indicated in ascending order.

b. Device serial number

When X'80' is specified in the "page code" field of the CDB, the device serial number of the IDD is sent. Figure 3.3 shows the format of this information.

	Bit 7	6	5	4	3	2	1	0
Byte 0	Qualifier			Device type code				
1	X'80' (page code)							
2	X'00'							
3	X'0C' (page length)							
4	Device serial number (ASCII)							
5								
≈								
15								

Figure 3.3 VPD information: device serial number

The "qualifier" field and the "device type code" field in byte 0 contain the same values as those in the standard INQUIRY data. The "page code" field in byte 1 indicates the identifier of this VPD information. The "page length" field in byte 3 indicates the byte-length of byte 4 and later of this data, without regard to the transfer byte-length field value of the CDB. This value is always X'0C', therefore, the total data length is 16 bytes.

Bytes 4 to 15 contain a right-justified decimal integer (ASCII code) that indicates the device serial number of the IDD. The remaining high-order digits contain ASCII space characters. When the disk drive is not ready, when system information cannot be read normally from the disk, or when a nonexisting LUN (other than LUN = 0) is specified, all the bytes in this field contain ASCII space characters.

3.1.3 READ CAPACITY

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'25'							
	1	LUN			0	0	0	0	0
	2	Logical block address (MSB)							
	3	Logical block address							
	4	Logical block address							
	5	Logical block address (LSB)							
	6	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	PMI
	9	0	0	0	0	0	0	Flag	Link

The READ CAPACITY command sends information of disk drive capacity and data block size to the INIT.

When the partial medium indicator (PMI) bit (bit 0 in CDB byte 8) is 0, the logical block address and block length (byte-length) of the last data block accessible on the disk drive (user space) are sent to the INIT. In this case, zero must be specified in the "logical block address" field in the CDB.

When the PMI bit is 0 and X'80000000' (starting block address of the CE space) is specified in the "logical block address" field, the logical block address and block length (byte-length) of the last data block accessible on the CE space are sent to the INIT.

When the PMI bit is 1, the logical block address and block length (byte-length) of the data block that is located after the data block specified in the "logical block address" field of the CDB and satisfies one of the following conditions are sent to the INIT. When the address of a block in the CE space is specified in the "logical block address" field, the logical block address and block length (byte-length) of the data block that is located in the CE space and satisfies one of the following conditions are sent to the INIT.

- The data block is located on the same cylinder as that containing the specified data block and is followed by the data block that was first processed by the alternate sector processing (excluding the defective block sector slip processing). Otherwise, the data block is specified in the CDB when the above, specified data block had the alternate sector processing.

- The data block is the last data block when the cylinder that contains the above, specified data block does not have the data block processed by the alternate sector processing.

By using this command in which PMI bit is 1, the INIT can retrieve a continuously usable database of which data transfer is not stopped by alternate sector processing or cylinder switching operation.

Figure 3.4 shows the format of the data to be transferred to the INIT by this command.

	Bit							
	7	6	5	4	3	2	1	0
Byte 0	Logical block address (MSB)							
1	Logical block address							
2	Logical block address							
3	Logical block address (LSB)							
4	Block length (MSB)							
5	Block length							
6	Block length							
7	Block length (LSB)							

Figure 3.4 READ CAPACITY data

3.1.4 MODE SELECT

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'15'							
	1	LUN			PF	0	0	0	SP
	2	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0
	4	Parameter list length							
	5	0	0	0	0	0	0	Flag	Link

The MODE SELECT command sets or changes various parameters of disk drive physical attributes, data format, SCSI bus disconnection/connection timing, and error recovery. Using the MODE SENSE command, the INIT can confirm the types and contents of the IDD-supported parameters, and current value of each parameter or the types and value range of changeable parameters.

The data (MODE SELECT parameters) that is to be transferred to the IDD from the INIT consists of the header, block descriptor, and one or more page descriptors (consisting of various parameters), as explained later.

When the page format (PF) bit in the CDB byte 1 is 1, the MODE SELECT parameters that are to be transferred from the INIT by using this command are in the page descriptor format. The IDD ignores this bit and executes this command with assuming that the MODE SELECT parameters from the INIT are in the page descriptor format.

The save pages (SP) bit in the CDB byte 1 specifies whether the MODE SELECT parameters specified in this command are to be saved onto the disk. The IDD can save all the MODE SELECT parameters that were described in page descriptors. However, the PF bit is valid only about the parameters other than the format parameter (page 3) and drive parameter (page 4: except byte 17).

When the SP bit is 1, all the MODE SELECT parameters (other than the format parameter (page 3) and drive parameter (page 4: except byte 17) that were transferred from the INIT are saved onto the disk at execution of this command. When the SP bit is 0, these parameters are not saved onto the disk. On the other hand, when a FORMAT UNIT command is executed, the format parameter (page 3) and drive parameter (page 4: except byte 17) are always saved onto the disk without regard to the SP bit specification given at issuing this command.

The "parameter list length" field in the CDB specifies the total byte-length of the MODE SELECT parameters transferred from the INIT by using this command. When 0 is specified in the "parameter list length" field, the command is terminated normally without executing the data transfer from the INIT. The format and length of the MODE SELECT parameters are defined as described in each page descriptor explanation of each subsection, and the INIT must always transfer data using this format and length.

When the "parameter list length" field value is less than the total definition length of the MODE SELECT parameters that were actually transferred from the INIT, the whole of the header, block descriptor, and page descriptors cannot be transferred. In this case or when the specified value of the "parameter list length" field exceeds the total definition length of the MODE SELECT parameters transferred from the INIT, this command terminates with the CHECK CONDITION status (ILLEGAL REQUEST [=5] / Invalid Field in CDB [=24-00]) and all the MODE SELECT parameters that were transferred by the command are invalid.

The MODE SELECT parameter values are classified into three types: current values, save values, and default values. The current values of the parameters actually control the IDD operation. The parameters specified in this command are used to change the current values. The save values of the parameters are the parameter values that were specified in this command and then saved on the disk. The default values of the parameters are used as the current values of the parameters by the IDD during the period from IDD power-on to save-value read completion or in the case where there are no save values and the MODE SELECT command is not issued.

For the disk data format parameters (block descriptor, format parameter (page 3), and drive parameter (page 4)), the IDD has a set of current values and save values common to all INITs. Moreover, about other MODE SELECT parameters, the IDD has the current values and save values for each SCSI ID of the INIT (these values are not common to all INITs), and has a set of default values common to all INITs. Therefore, for these parameters, each INIT can have different values.

When the IDD power is turned on, when a RESET condition occurs, or when the IDD receives a BUS DEVICE RESET message, the current values of the MODE SELECT parameters are initialized to the save values. When there are no save values, the current values are initialized to the default values.

When this command changes any of the disk data format parameters (block descriptor (page 3 or 4)), the UNIT ATTENTION condition (MODE parameters changed [=2A-01]) occurs in all the INITs other than the INIT that issued this command.

When a RESERVE UNIT command in which the third party reserve function is specified is issued, the current values of the MODE SELECT parameters of the INIT that issued the RESERVE UNIT command are copied onto the current value fields for the third party device specified with the RESERVE UNIT command. See Subsection 3.1.11 for details of this function.

Figure 3.5 shows the relationship between the current values, save values, and default values of the MODE SELECT parameters.

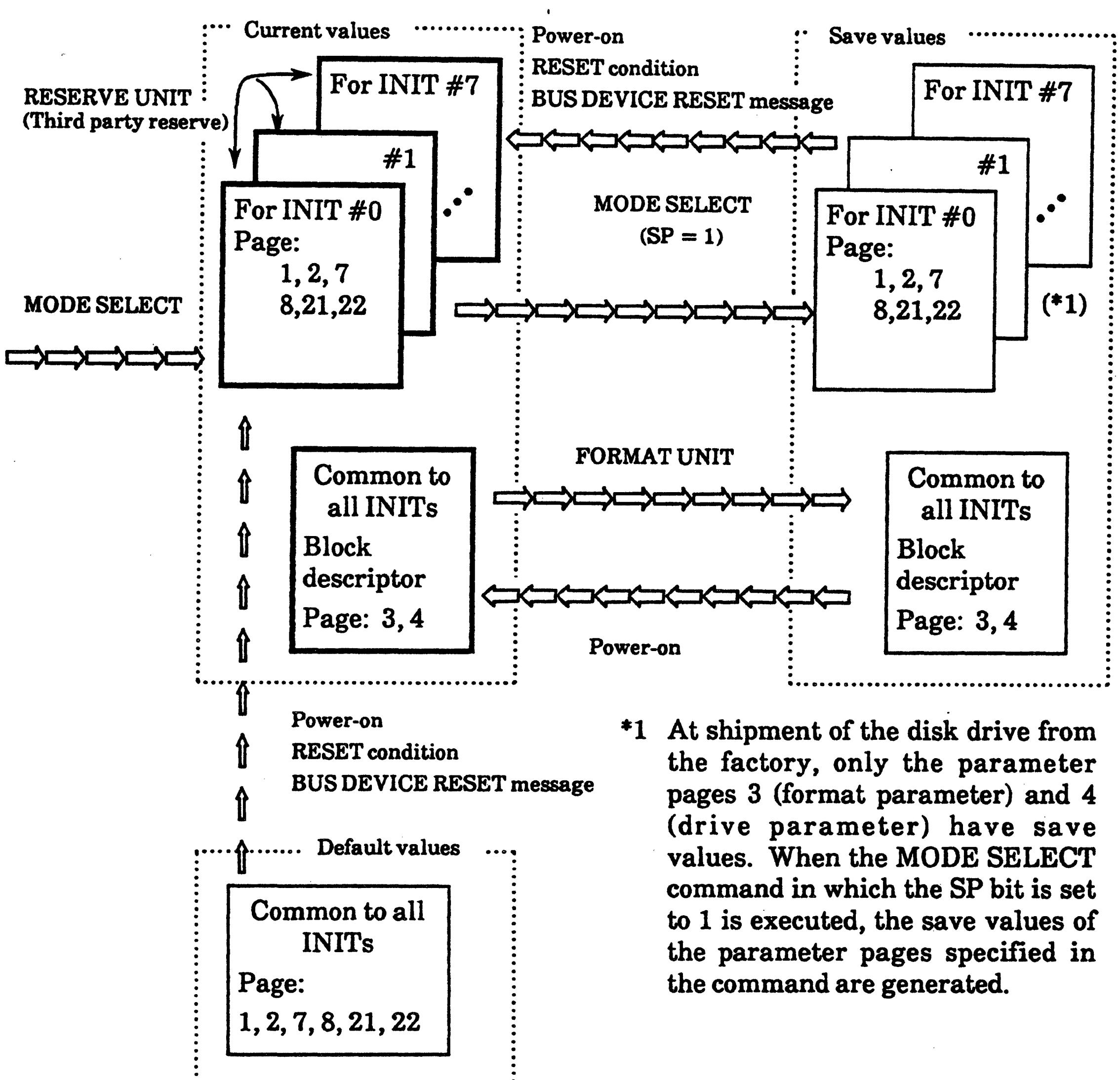


Figure 3.5 Structure of MODE SELECT parameters

The MODE SELECT parameters can be classified into three types: (1) parameters of which values can be arbitrarily changed by the INIT, (2) parameters of which values are determined by the IDD and cannot be changed by the INIT, and (3) parameters of which values can be changed within a specified range by the INIT. The values of some parameter fields are specified by the INIT and then rounded up or down by the IDD so that these values can be used in actual operation.

If changing an unchangeable MODE SELECT parameter transferred from the INIT by this command is requested or if changing a changeable MODE SELECT command into a value outside the changeable range is requested, this command terminates with the CHECK CONDITION status (ILLEGAL REQUEST [=5]/ Invalid field in parameter list [=26-00]) and all the parameters specified in this case by the INIT are invalid. However, some changeable parameters have parameter fields of which values specified by the INIT are ignored. For details, see the explanation below of the page descriptors in this section.

When a parameter specified by the INIT is rounded up or down by the IDD, the IDD posts the CHECK CONDITION status (RESERVED ERROR [=1]/Rounded parameter [=37-00]) for the command according to MODE SELECT parameter (page 21 bit 5: RPR [Rounded Parameter Report]) or terminates the command successfully with posting the GOOD status. In this case, even if the CHECK CONDITION status is posted, all parameters specified by the command are effective. This case are all valid. In this case, the INIT can confirm the rounding up/down result by reading the current values of the parameters by issuing a MODE SENSE command. For an explanation of the parameter fields to be rounded up or down, see the explanation below of the page descriptors.

Figure 3.6 shows the data configuration of a parameter list to be transferred from the INIT by this command. A parameter list consists of a four-byte header, an eight-byte block descriptor, and one or more page descriptors. A parameter list can consist of only the four-byte header and one or more page descriptors. The INIT can transfer only the header or only the header and the block descriptor.

Header

	Bit							
	7	6	5	4	3	2	1	0
Byte 0	X'00'							
1	X'00' (medium type)							
2	×	0	0	×	0	0	0	0
3	X'00' or X'08' (block descriptor length)							

Block descriptor

	Bit							
	7	6	5	4	3	2	1	0
Byte 0	X'00'							
1	Data block count (MSB)							
2	Data block count							
3	Data block count (LSB)							
4	X'00'							
5	Data block length (MSB)							
6	Data block length							
7	Data block length (LSB)							

Page descriptor

		Bit							
		7	6	5	4	3	2	1	0
Page header	Byte 0	0	0	Page code					
	1	Page length							
	2	Parameter field							
	≈								
	n								

Figure 3.6 Parameter configuration of MODE SELECT command

a. Header

(a) Medium type

X'00' (default type) must be specified in this field.

(b) Byte 2

Bits 7 and 4 in byte 2 are valid only when this header is to be transferred to the INIT by a MODE SENSE command. When this header is to be transferred by a MODE SELECT command, these bits are ignored. Other bits must always be cleared to 0 by the INIT.

(c) Block descriptor length

This field specifies the byte-length of the block descriptor that follows the header. This length does not include the length of the page descriptors. For the IDD, only one block descriptor can be defined, and X'00' or X'08' must be specified in the block descriptor length field. When this field specifies X'00', the parameter list transferred from the INIT does not contain a block descriptor and the header is followed by page descriptors.

b. Block descriptor

The block descriptor consists of eight bytes, and specifies the logical attributes of the data format on the disk.

Note:

The parameters contained in the block descriptor are closely related to the format parameter (page 3) and drive parameter (page 4). When the parameter values in the block descriptor are to be changed, the INIT must also change the values of the format parameter and drive parameter.

When only the block descriptor is changed by this command, the IDD changes the sector/track value, the data byte length/physical sector value, track skew factor value, and cylinder skew factor value (in the format parameter (page 3)) and the cylinder count value (in the drive parameter (page 4)) according to the specified block descriptor contents. For other parameters, the IDD uses their current values.

(a) Data block count

This field specifies the total number of logical blocks to be assigned to the user space on the disk. The block length is specified in the "data block length" field. The number of cylinders constituting the user space, and the configuration of the spare area for defect handling are specified in the format parameter (page 3) and drive parameter (page 4). In the "data block count" field, the INIT had better specify zero or the value equal to the total logical data block count determined from the "data block length" field value, format parameter (page 3), and drive parameter (page 4).

When zero is specified in this field, all the user spaces that are determined according to the format parameter (page 3) and drive parameter (page 4) are configured with the logical data blocks, the length of which is specified in the "data block length" field.

When this field contains a nonzero value that is greater or less than the total logical data block count that is determined according to the "data block length" field value, format parameter (page 3), and drive parameter (page 4), the value specified in the "data block count" field is used as the total number of logical data blocks in the user space unless the number of the cylinders required to allocate the logical data blocks (the count of which is specified in this field) exceeds the maximum limit on the disk drive. (The maximum logical data block address is equal to the value acquired by subtracting 1 from the value specified in this field.)

If the number of the cylinders required to allocate the logical data blocks (the count of which is specified in this field) exceeds the maximum limit on the disk drive, the IDD determines a valid data block count value by considering the maximum cylinder count on the disk drive, and sets this value into the "data block count" field. Using a MODE SENSE command, the INIT can confirm the actually determined data block count.

The CE space has the same logical data block format as the user space. The number of the data blocks in the CE space (on one cylinder) is automatically determined according to the format parameter (page 3). The INIT cannot explicitly specify the number of data blocks contained in the CE space.

(b) Data block length

This field specifies the byte-length of a logical data block on a disk. In the IDD, the logical data block length must be equal to the physical data block length. An even numbered byte length value of data block value in the range of 256 to 4096 bytes can be specified in this field.

If 255 or a lower value is specified, the IDD sets 256 into this field. If 4097 or a greater value is specified, the IDD sets 4096 into this field. If an odd value is specified, IDD adds 1 to this value and sets it into this field.

c. Page descriptors

A page descriptor consists of a two-byte length of the page header and a subsequent parameter field. One parameter function attribute corresponds to one page. Using a MODE SELECT command, the INIT can specify one page descriptor or can specify two or more page descriptors in an arbitrary order.

(a) Page code

This field specifies a page descriptor type indication code (page number).

(b) Page length

This field specifies the byte-length of the parameter field from byte 2 (that is, the length of the page header is not included). The value equal to the page length sent by the IDD by using a MODE SENSE command must be specified into this field by the INIT except the case described in the explanation of each page parameter in this subsection.

d. MODE SELECT parameters for IDD

The following table lists the contents and length of the parameter list that can be transferred from the INIT to the IDD by using a MODE SELECT command. If a page descriptor that is not supported by the IDD is specified and a page length is not zero, the command terminates with the CHECK CONDITION status (ILLEGAL REQUEST [=5]/ Invalid field in parameter list [=26-00]) and all the parameters specified in the command are invalid.

Parameter		Byte-length
Header		4
Block descriptor		8 or 0
Page descriptor		
Page 1:	Read/write error recovery parameter	8 or 12
Page 2:	Disconnect/reconnect parameter	12 or 16
Page 3:	Format parameter	24
Page 4:	Drive parameter	12, 20 or 24
Page 7:	Verify error recovery parameter	12
Page 8:	Caching parameter	12
Page 21:	Additional error recovery parameter	4
Page 22:	Reconnection timing parameter	4

Notes:

1. The configuration of the MODE SELECT parameters generally depends on the type of the I/O device and controller used. Also, the current IDD definitions may be extended by future function extensions. To assure independence from the software specifications for each I/O device, the INIT should issue a MODE SENSE command to confirm the types and attributes of the parameters actually supported in TARG, before issuing this MODE SELECT command.

2. When a block descriptor, format parameter (page 3), or drive parameter (page 4) is changed using a MODE SELECT command, a command for accessing data on the disk cannot be executed until the execution of the FORMAT UNIT command is completed. If such a disk data access command is executed, the CHECK CONDITION status (MEDIUM ERROR [= 3]/Medium format corrupted [= 31-00]) is sent.

The following explains the configuration and function of the page descriptor supported by the IDD.

Note:

In the following page descriptor configuration indication figures, the default field indicates the default value of the parameter, and the variable field indicates whether the parameter value can be varied. ('1' indicates that the parameter value can be varied.) The INIT can confirm these values and attributes by using a MODE SENSE command.

(1) Read/write error recovery parameter (page code 1)

Figure 3.7 shows the format of the read/write error recovery parameter in this page descriptor of the MODE SELECT command.

	Bit								
	7	6	5	4	3	2	1	0	
Byte 0	0	0	0	0	0	0	0	1	
1	X'0A' or X'06' (page length)								(*1)
2	AWRE	ARRE	TB	0	EER	PER	DTE	DCR	
Default value	0	0	1	0	1	0/1	0	0	
Variable value	1	1	1	0	1	1	1	1	
3	Retry count at read								
Default value	0	0	0	1	0	0	1	0	(= 18 times)
Variable value	1	1	1	1	1	1	1	1	
4	Correctable bit length								
Default value	0	0	0	0	1	0	0	0	(= 8 bits)
Variable value	0	0	0	0	0	0	0	0	
5	X'00' (reserved)								
6	X'00' (reserved)								
7	X'00' (reserved)								
8	Retry count at write								
Default value	0	0	0	1	0	0	1	0	(= 18 times)
Variable value	1	1	1	1	1	1	1	1	
9	X'00' (reserved)								
10-11	X'0000' (reserved)								

*1 When transfer of this page descriptor is requested by a MODE SENSE command, the IDD posts X'0A' as the page length (byte 1). However, when a MODE SELECT command is executed and X'0A' or X'06' is specified as the page length, the IDD assumes that the correct page length is being specified. When X'06' is specified as the page length, the value specified in the retry count at read field is used as the retry count at write field value, without change. Page length X'06' is supported by considering the compatibility with the conventional models. However, the INIT should use page length X'0A' when possible.

Figure 3.7 MODE SELECT parameter: Read/write error recovery parameter

The error recovery parameter defined in this page descriptor can be applied to the following commands, except in specified cases.

- READ
- READ EXTENDED
- READ LONG
- RECOVER DATA
- RECOVER ID
- SEND DIAGNOSTIC (Write/read test)
- WRITE
- WRITE AND VERIFY (Write operation)
- WRITE EXTENDED
- WRITE LONG
- WRITE SAME

a. AWRE: Automatic write reallocation enabled

1: Enables execution of the automatic alternate block assignment function during write operation.

0: Disables execution of the automatic alternate block assignment function during write operation.

Note:

For the automatic alternate block assignment, see Subsection 4.3.2.

b. ARRE: Automatic read reallocation enabled

1: Enables execution of the automatic alternate block assignment function during read operation.

0: Disables execution of the automatic alternate block assignment function during read operation.

Note:

For the automatic alternate block assignment, see Subsection 4.3.2.

c. TB: Transfer block

1: Specifies to transfer to the INIT, a data block in which an uncorrectable error was detected during read operation.

0: Specifies not to transfer to the INIT, a data block in which an uncorrectable error was detected during read operation.

Note:

The data block that was error-corrected or recovered by the error recovery function of the IDD is always transferred to the INIT without regard to the specification of this bit.

d. EER: Enable early recovery

1: Specifies immediate execution of the data correction function of ECC without executing a read retry up to the number of times specified in the "retry count at read" parameter, if a correctable data check is detected.

0: Specifies execution of a read retry up to the number of times specified in the "retry count at read" parameter if a correctable data check is detected. If possible, ECC corrects the data later.

e. PER: Post error

1: Specifies sending the CHECK CONDITION status when the execution of the command in which an error (regarding to the disk drive) was detected but corrected by the error recovery function of the IDD is terminated. Sense data is generated to send the contents of the last corrected error, and the sense key indicates RECOVERED ERROR [=1].

0: Specifies termination of a command with the GOOD status even if an error (regarding to the disk drive) is detected and corrected by the error recovery function of the IDD. (The contents of the corrected error are not sent.)

Note:

The default value of the PER flag can be set to 1 or cleared to 0 by using the setting terminal on the IDD. However, the current value is initialized after the power is turned on, a RESET condition occurs, or a BUS DEVICE RESET message is received. When there is a save value for this page descriptor, this save value is used, instead of the default value specified with the setting terminal, as the current value. Refer to Subsection 5.3.2 of OEM Manual Specifications & Installation for an explanation of the setting terminal.

f. DTE: Disable transfer on error

1: Specifies termination of the execution of a command even if an error detected on the disk drive is corrected by the error recovery function of the IDD.

0: Specifies continuation of the execution of a command when an error detected on the disk drive is corrected by the error recovery function of the IDD.

g. DCR: Disable correction

1: Specifies not to execute the error correction by ECC even if a correctable error is detected.

0: Specifies execution of the error correction by ECC when a correctable error is detected.

h. Retry count at read

This parameter specifies the number of times a retry is to be executed for a recovery from a data check error detected during a disk read operation. The "retry count" specified in this parameter is the maximum read retry count for the ID field and data field of each logical data block. If an error is detected, the IDD executes a read retry up to the specified number of times to recover the ID field of the data block. When data is not immediately corrected by ECC, the IDD executes a read retry up to the specified number of times to recover the data field of the data block. Before the retry specified in this field, the IDD performs retry once internally. Therefore, the actual retry count is specified number plus one. Even if zero is specified in this parameter, the read retry is performed automatically. In this case, data field correction by ECC is executed according to the EER and DCR control flags.

i. Correctable bit length

This parameter indicates the maximum burst error length (bit length) that can be corrected by ECC. This value cannot be changed by the INIT. The IDD ignores the set value of this parameter and operates according to the default value.

j. Retry count at write

This parameter specifies the maximum number of times a read retry is to be executed when a data check error is detected during the ID field verification at write on the disk. The retry count specified in this parameter is applied in logical data block units. Before the retry specified in this field, the IDD performs retry once internally. Therefore, the actual retry count is specified number plus one. Even if zero is specified in this parameter, the read retry at ID field verification is performed automatically.

Table 3.1 Combination of error recovery flags (1/4)

EER	PER	DTE	DCR	Error recovery procedure
0	0	0	0	① A read retry is executed up to the number of times specified in the "retry count at read", "retry count at write", or "retry count at verify" parameter. ECC corrects the error when possible. ② When the data is recovered from the error, the command execution is continued. ③ The contents of the recovered error are not sent to INIT. ④ If an unrecoverable error is detected, the command execution is terminated at once. ⑤ The data in the block containing an unrecoverable error is transferred to the INIT according to the TB bit specification. (read command)
0	0	0	1	① A read retry is executed up to the number of times specified in the "retry count at read", "retry count at write", or "retry count at verify" parameter. ECC does not correct the error. ② When the data is recovered from the error, the command execution is continued. ③ The contents of the recovered error are not sent to INIT. ④ If an unrecoverable error is detected, the command execution is terminated at once. ⑤ The data in the block containing an unrecoverable error is transferred to the INIT according to the TB bit specification. (read command)
0	0	1	0	----- Unspecifiable flag combination ----- (*1)
0	0	1	1	----- Unspecifiable flag combination ----- (*1)
0	1	0	0	① A read retry is executed up to the number of times specified in the "retry count at read", "retry count at write", or "retry count at verify" parameter. ECC corrects the error when possible. ② When the data is recovered from the error, the command execution is continued. ③ If an unrecoverable error is detected, the command execution is terminated at once. ④ The data in the block containing an unrecoverable error is transferred to the INIT according to the TB bit specification. (read command) ⑤ When the data is recovered from all the detected errors, the CHECK CONDITION status (RECOVERED ERROR [=1]) is sent at the termination of all the processing of the command. In this case, the sense data indicates the contents of the last recovered error and the address of the error-detected data block.

Table 3.1 Combination of error recovery flags (2/4)

EER	PER	DTE	DCR	Error recovery procedure
0	1	0	1	① A read retry is executed up to the number of times specified in the "retry count at read", "retry count at write", or "retry count at verify" parameter. ECC does not correct the error. ② When the data is recovered from the error, the command execution is continued. ③ If an unrecoverable error is detected, the command execution is terminated at once. ④ The data in the block containing an unrecoverable error is transferred to the INIT according to the TB bit specification. (read commands) ⑤ When the data is recovered from all the detected errors, the CHECK CONDITION status (RECOVERED ERROR [=1]) is sent at the termination of all the processing of the command. In this case, the sense data indicates the contents of the last recovered error and the address of the error-detected data block.
0	1	1	0	① A read retry is executed up to the number of times specified in the "retry count at read", "retry count at write", or "retry count at verify" parameter. ECC corrects the error when possible. ② Without regard to the error recovery success/failure, the command execution is terminated with the CHECK CONDITION status when the error recovery is terminated. In this case, the sense data indicates the address of the data block in which the error was detected. ③ The data block recovered from the error is transferred to the INIT. The data block in which an unrecoverable error was detected is transferred to the INIT according to the TB bit specification. (read commands)
0	1	1	1	① A read retry is executed up to the number of times specified in the "retry count at read", "retry count at write", or "retry count at verify" parameter. ECC does not correct the error. ② Without regard to the error recovery success/failure, the command execution is terminated with the CHECK CONDITION status when the error recovery is terminated. In this case, the sense data indicates the address of the data block in which the error was detected. ③ The data block recovered from the error is transferred to the INIT. The data block in which an unrecoverable error was detected is transferred to the INIT according to the TB bit specification. (read commands)

Table 3.1 Combination of error recovery flags (3/4)

EER	PER	DTE	DCR	Error recovery procedure
1	0	0	0	① If a correctable data check is detected, it is corrected by ECC at once. If an uncorrectable data check is detected, a read retry is executed up to the number of times specified in the "retry count at read", "retry count at write", or "retry count at verify" parameter. However, when error correction is enabled during the retry, the error is corrected by ECC at once. ② When the error recovery succeeds, the command execution is continued. ③ The contents of the recovered error are not sent to the INIT. ④ If an unrecovered error is detected, the command execution is terminated at once. ⑤ The data in the data block containing the unrecoverable error is transferred to the INIT according to the TB bit specification. (read commands)
1	0	0	1	----- Unspecifiable flag combination ----- (*1)
1	0	1	0	----- Unspecifiable flag combination ----- (*1)
1	0	1	1	----- Unspecifiable flag combination ----- (*1)
1	1	0	0	① If a correctable data check is detected, it is corrected by ECC at once. If an uncorrectable data check is detected, a read retry is executed up to the number of times specified in the "retry count at read", "retry count at write", or "retry count at verify" parameter. However, when error correction is enabled during the retry, the error is corrected by ECC at once. ② When the error recovery succeeds, the command execution is continued. ③ If an unrecovered error is detected, the command execution is terminated at once. ④ The data in the data block containing the unrecoverable error is transferred to the INIT according to the TB bit specification. (read commands) ⑤ When the data is recovered from all the detected errors, the CHECK CONDITION status (RECOVERED ERROR [=1]) is sent at the termination of all the processing of the command. In this case, the sense data indicates the contents of the last recovered error and the address of the error-detected data block.
1	1	0	1	----- Unspecifiable flag combination ----- (*1)

Table 3.1 Combination of error recovery flags (4/4)

EER	PER	DTE	DCR	Error recovery procedure
1	1	1	0	① If a correctable data check is detected, it is corrected by ECC at once. If an uncorrectable data check is detected, a read retry is executed up to the number of times specified in the retry count at read, retry count at write, or retry count at verify parameter. However, when error correction is enabled during the retry, the error is corrected by ECC at once. (The ID field is subject to only the read retry.) ② Without regard to the error recovery success/failure, the command execution is terminated with the CHECK CONDITION status when the error recovery is terminated. In this case, the sense data indicates the address of the data block in which the error was detected. ③ The data block recovered from the error is transferred to the INIT. The data block in which an unrecoverable error was detected is transferred to the INIT according to the TB bit specification. (READ or READ EXTENDED)
1	1	1	1	----- Unspecifiable flag combination --- -- (*1)

*1 If an unspecifiable error recovery flag combination is specified, the MODE SELECT command terminates with the CHECK CONDITION status (ILLEGAL REQUEST [=5]/ Invalid field in parameter list [=26-00]) and all the parameters specified in it are invalid.

(2) Disconnect/reconnect parameter (page code 2)

Figure 3.8 shows the format of the disconnect/reconnect parameter in this page descriptor of the MODE SELECT command.

	Bit								
	7	6	5	4	3	2	1	0	
Byte 0	0	0	0	0	0	0	1	0	
1	X'0E' or X'0A' (page length)								(*1)
2	Buffer full ratio								
Default value	0	0	0	0	0	0	0	0	(= 1 block)
Variable value	1	1	1	1	1	1	1	1	
3	Buffer empty ratio								
Default value	1	0	0	0	0	0	0	0	(= 28 KB)
Variable value	1	1	1	1	1	1	1	1	
4-5	Bus inactive limit								
Default value	0	0	1	1	0	0	1	0	(= 5 ms)
Variable value	0	0	0	0	0	0	0	0	
6-7	X'0000' (reserved)								
8-9	X'0000' (reserved)								
10-11	X'0000' (reserved)								
12	X'00' (reserved)								
13-15	X'000000' (reserved)								

*1 When the transfer of the page descriptor is requested by the MODE SELECT command, the IDD reports X'0E' as a page length (byte 1). However, when X'0E' or X'0A' is specified in the page length for the MODE SELECT command, the IDD assumes that the correct page length is specified. X'0A' of the page length is supported by considering the compatibility with the conventional models. It is recommended that the INIT uses X'0E' for the page length to prepare for furture specification extension.

Figure 3.8 MODE SELECT parameter: Disconnect/reconnect parameter

See Section 2.1 for details of the specification method of the "buffer full ratio" parameter and the "buffer empty ratio" parameter contained in this page descriptor and the data buffer operation.

a. Buffer full ratio

This parameter specifies when the IDD should start reconnection in order to transfer data to the INIT by using a READ or READ EXTENDED command.

This parameter value (n) specifies "the amount of the data to be read from the disk to the data buffer" with a ratio of it to "the total capacity of the data buffer" (n/256). The maximum transferable capacity of the each data buffer of the IDD is 56 KB. Therefore, when "n" is 32, this parameter specifies 7 KB of data (*1).

When transferring the data (the length of which is specified in this parameter) from the data buffer to the INIT is enabled, the IDD executes the reconnection and starts the data transfer to the INIT.

The default value of this parameter is X'00'. Since the actual, minimum value of the "buffer full ratio" is one logical data block, X'00' is the same as the specification of one logical data block (*1).

b. Buffer empty ratio

This parameter specifies when the IDD should start reconnection in order to request the INIT to restart data transfer by using a WRITE or WRITE EXTENDED command.

This parameter value (n) specifies "the size of the free area in the data buffer of the IDD" with a ratio of it to "the total capacity of the data buffer" (n/256). The maximum transferable capacity of the each data buffer of the IDD is 56 KB. Therefore, when "n" is 64, this parameter specifies 14 KB as the size of the free area on the buffer (*1).

To write data onto the disk, the IDD sequentially transfers the data that was prefetched into the data buffer. When the size of the free area on the data buffer equals the value specified in this parameter, the reconnection is executed and transferring the next data is requested.

When the total transfer data block byte-length specified in the command is 56 KB or less, all the data is prefetched. Therefore, this parameter is ignored.

*1 When the value specified in the "buffer full ratio" or "buffer empty ratio" parameter is not on the boundary of a multiple of the logical data block length, it is assumed that the data block boundary that is rounded up the specified value is being specified. When zero is specified in one of these parameters, the IDD assumes that the length of one logical data block is being specified in the parameter. The parameter values specified in the MODE SELECT command by the INIT are retained without change, and the MODE SENSE command sends these values. (These values are not rounded by the IDD.)

c. Bus inactive limit

This parameter specifies the maximum time, that the TARG can make the SCSI bus the busy state (BSY signal is true) without executing the REQ/ACK handshaking, as a multiplier of 100 μs. For the IDD, this value is 5 ms in the normal operation. This parameter cannot be changed by the INIT. The IDD ignores this parameter value and operates according to the default value.

(3) Format parameter (page code 3)

Figure 3.9 shows the format of the format parameter contained in this page descriptor of the MODE SELECT command.

	Bit								
	7	6	5	4	3	2	1	0	
Byte 0	0	0	0	0	0	0	1	1	
1	X'16' (page length)								
2 – 3	Number of tracks per zone								
Default value	X'000x'								(= 1 cylinder)
Variable value	X'0000'								
4 – 5	Number of alternate tracks per zone								
Default value	X'0003'								(= 3 sectors)
Variable value	X'FFFF'								
6 – 7	Number of alternate tracks per zone								
Default value	X'0000'								
Variable value	X'0000'								
8 – 9	Number of alternate tracks per zone								
Default value	X'000x'								(= 1 cylinder)
Variable value	X'FFFF'								
10–11	Number of sectors per track								
Default value	X'00xx'								
Variable value	X'0000'								

Figure 3.9 MODE SELECT parameter: Format parameter (1/2)

12–13	Data byte-length per physical sector							
Default value	X'xxxx'							
Variable value	X'FFFF'							
14–15	Interleave factor							
Default value	X'0001'							
Variable value	X'0000'							
16–17	Track skew factor							
Default value	X'00xx'							
Variable value	X'0000'							
18–19	Cylinder skew factor							
Default value	X'00xx'							
Variable value	X'0000'							
20	SSEC	HSEC	RMB	SURF	0	0	0	0
Default value	0	1	0	0	0	0	0	0
Variable value	0	0	0	0	0	0	0	0
21–23	X'000000' (reserved)							

Figure 3.9 MODE SELECT parameter: Format parameter (2/2)

a. Parameters for specifying alternate areas for faulty blocks (bytes 2 to 9)

The following four parameters specify the number and positions of spare sectors required to assign alternate blocks to faulty blocks on a disk. See Chapter 3 of OEM Manual Specifications & Installation for details of the IDD alternate block processing.

(a) Number of tracks per zone

This parameter specifies the alternate-block spare-sector assignment units (zone). In the IDD, this value is fixed to the number of tracks per cylinder, and cannot be changed by the INIT. In this parameter, the INIT must specify zero or the same value as the number of the heads of the disk drive. When a nonzero value that is not equal to the number of the heads is specified in this parameter, the IDD sets the value that is equal to the number of tracks per cylinder into this parameter (this processing is called parameter rounding).

(b) Number of alternate sectors per zone

This parameter specifies the number of spare sectors (per zone) to be reserved for alternate blocks. In the IDD, this parameter indicates number of the spare sectors that are prepared on each cylinder. The maximum allowable value of this parameter is the smaller one of "32" or "(Number of sectors per track in zone 4) - 1". The INIT can change this parameter value within the range of 0 to the maximum allowable value. If a value greater than the maximum allowable value is specified, the IDD sets the maximum allowable value into this parameter (parameter rounding down).

(c) Number of alternate tracks per zone

This parameter specifies the number of the tracks (per zone) required for reserving an alternate block area. This parameter value cannot be changed. When a nonzero value is specified in this parameter, the IDD sets zero in this parameter with executing the parameter rounding down process.

(d) Number of alternate tracks per drive

This parameter specifies the number of tracks (per disk drive) for reserving an alternate block area. In the IDD, this parameter indicates number of spare sectors that are prepared on each cylinder. In this parameter, the INIT must specify a multiple of the number of the heads of the disk drive. The number of tracks on zero to seven cylinders can be specified. If the specified value is not equal to a multiple of the number of the heads, the IDD rounds up the value and sets the valid track count (that satisfies the cylinder boundary conditions) into this parameter. If a track count exceeding the number of tracks on seven cylinders is specified, the IDD sets the number of tracks on seven cylinders into this parameter (parameter rounding up).

Note:

When zero is set in both the "number of sectors per zone" and the "number of alternate tracks per drive" parameters, the IDD sets values that are equal to the default values (3 is set into the "number of alternate sectors per zone" parameter, and the number of tracks required for one cylinder is set into the "number of alternate tracks per drive" parameter) into these parameters.

b. Parameters for specifying track format (bytes 10 and 11)

(a) Number of sectors per track

This parameter indicates the number of physical sectors per track. In the IDD, the number of sectors on the track is determined as an unique value, depending on the "data block length" parameter of the block descriptor and the data format specified by the "data byte length per physical sector" parameter of this page descriptor. Therefore, this parameter cannot be changed. The IDD ignores this parameter value. The default value of this parameter indicates the mean value of four zones with the current data format.

c. Parameters for specifying sector format (bytes 12 to 19)

(a) Data byte-length per physical sector

This parameter specifies the byte-length of the data to be stored in one physical sector. In the IDD, the byte-length of the data to be stored in one physical sector is the same as the byte-length of the data to be stored in one logical data block. In this parameter, the INIT must specify zero or the same value as the "data block length" parameter value specified in the block descriptor.

When zero is specified in this parameter, the data block length value specified in the current block descriptor is used as the value of this parameter.

The nonzero value that is specified in this parameter by the INIT must be an even number in the range of 256 to 4096. If 255 or a lower value is specified, the IDD sets 255 into this parameter. If 4096 or a greater value is specified, the IDD sets 4096 into this parameter. If an odd number is specified, the IDD adds 1 to the number, and sets the resultant value into this parameter.

When both this page descriptor and block descriptor are specified by the same INIT and a nonzero value that is not equal to the value of the "data block length" parameter in the block descriptor is specified, the IDD sets the value as same as the specified value of the "data block length" parameter in the block descriptor by the parameter rounding process.

On the other hand, when this page descriptor is specified without specifying the block descriptor, the data format on the disk (number of data blocks and data block length) is determined by the specified value of this parameter and the value of the "number of cylinders" parameter in the drive parameter (page 4).

The default value of this parameter indicates the value coincidence with the current data format.

(b) Interleave factor

This parameter is valid only when a MODE SELECT command is executed. The interleave factor of the current data format of the disk drive is sent to this parameter. In the IDD, this parameter always contains X'0001'. It means no interleave. When a MODE SELECT command is executed, the value specified in this parameter is ignored.

(c) Track skew factor

This parameter specifies the number of physical sectors (track skew) existing between the data block having the highest logical block address on a track and the data block having the next logical block address on the next track on the same cylinder. This parameter cannot be changed. The IDD ignores the value specified in this parameter, and sets the track skew value that is optimal for the specified data block length. Refer to Chapter 3 of OEM Manual Specifications & Installation for details on track skew. The default value of this parameter indicates the mean value of four zone with the current data format.

(d) Cylinder skew factor

This parameter specifies the number of physical sectors (cylinder skew) existing between the data block having the highest logical block address on a cylinder and the data block having the next logical block address on the next cylinder. This parameter cannot be changed. The IDD ignores the value specified in this parameter, and sets the cylinder skew value that is optimal for the specified data block length. Refer to Chapter 3 of OEM Manual Specifications & Installation for details on cylinder skew. The default value of this parameter indicates the mean value of four zone with the current data format.

d. Parameters about device type (byte 20)

(a) SSEC: Soft sectoring

When this bit is 1, the soft sectoring method is used as the data formatting method of the disk. The hard sectoring method is used in the IDD (see paragraph (6)), and the value of this bit cannot be changed. The INIT ignores the value of this bit.

(b) HSEC: Hard sectoring

When this bit is 1, the hard sectoring method is used as the data formatting method of the disk. The value of this bit cannot be changed. The IDD can use only the hard sectoring method as the data formatting method. The IDD ignores the value of this bit.

(c) RMB: Removable medium

When this bit is 1, the disk on the disk drive can be replaced. When this bit is 0, the disk on the disk drive cannot be replaced (fixed disk). The fixed disk is used in the IDD, and the value of this bit cannot be changed. The INIT ignores the value of this bit.

(d) SURF: Surface addressing

When this bit is 1, logical data blocks are addressed so that all the sectors on one storage side (one head) are addressed and then the sectors on the next storage side (next head) are addressed. On the other hand, when this bit is 0, logical data blocks are addressed so that all the sectors on one cylinder (all heads) are addressed and then the sectors on the next cylinder are addressed. Only latter addressing can be used for the IDD and the value of this parameter cannot be changed. The INIT ignores the value of this bit.

(4) Drive parameter (page code 4)

Figure 3.10 shows the format of the drive parameter in the page descriptor of the MODE SELECT command.

	Bit								
	7	6	5	4	3	2	1	0	
Byte 0	0	0	0	0	0	1	0	0	
1	X'16', X'12' or X'0A' (page length)								(*1)
2–4	Cylinder count								
Default value	X'000595'								(= 1429)
Variable value	X'FFFFFF'								
5	Head count								
Default value	X'000x'								
Variable value	X'0000'								
6–8	"Write precompensation" start cylinder								
Default value	X'000000'								
Variable value	X'000000'								
9–11	"Reduced write current" start cylinder								
Default value	X'000000'								
Variable value	X'000000'								

Figure 3.10 MODE SELECT parameter: Drive parameter (1/2)

12–13	Drive step rate							
Default value	X'0000'							
Variable value	X'0000'							
14–16	Landing zone cylinder							
Default value	X'000000'							
Variable value	X'000000'							
17	(reserved)						RPL	
Default value	0	0	0	0	0	0	0	0
Variable value	0	0	0	0	0	0	1	1
18	X'00' (reserved)							
19	X'00' (reserved)							
20–21	Medium rotation rate							
Default value	X'112E' (4,396 rpm)							
Variable value	X'0000'							
22–23	X'0000' (reserved)							

*1 When transfer of this page is requested by a MODE SENS command, the IDD posts X'16' as the page length. When X'16', X'12', or X'0A' is specified in the page length parameter of the MODE SELECT command, the IDD assumes that the correct page length is being specified. Page length X'12' and X'0A' are supported by considering the compatibility with the conventional models, but the INIT should use page length X'16' for future specification extension.

Figure 3.10 MODE SELECT parameter: Drive parameter (2/2)

a. Cylinder Count

This parameter specifies the total number of cylinders that constitute the user space on a disk. Note that this value includes the number of cylinders to be used for the alternate block area specified in the "number of alternate tracks per drive" parameter contained in the format parameter (page 3). The value of the cylinder count parameter can be changed. In this parameter, the INIT must specify zero or a value that is equal to or less than the maximum number of cylinders (=1429) usable for the user space.

When zero is specified in this parameter, it is assumed that the smaller value of the maximum number of cylinders in the user space or the number of the cylinders required to allocate the logical data blocks (the count of which is specified in the "data block count" parameter contained in the block descriptor) is being specified. When the "data block count" parameter value in the block descriptor is zero in this case, the maximum number of cylinders (=1429) is set into this parameter.

On the other hand, when a nonzero value is specified in this parameter, the IDD processes this parameter in the following way.

- When this page descriptor is specified with the block descriptor which contains a nonzero value of the "data block count" parameter by the same MODE SELECT command and the value set in this parameter does not coincide with the value specified in the "data block count" parameter in the block descriptor, the IDD sets the number of cylinders required to allocate the number of logical blocks, that is specified by the "data block count" parameter in the block descriptor, into this parameter after executing the parameter rounding in the limit of the maximum number of cylinders.

- When this page descriptor is specified with the block descriptor which contains a zero value of the "data block count" parameter, or when this page descriptor is specified without the block descriptor, the number of allocatable logical data blocks is determined from the specified value of this parameter and the specified value of the format parameter (page 3). When the value exceeding the maximum number of cylinders is specified, the IDD sets the maximum number of cylinders into this parameter by the parameter rounding (down). When the value of this parameter is equal to or smaller than the number of alternate cylinders specified in the "alternate track count per disk drive" parameter in the format parameter (page 3), the IDD sets the "number of alternate cylinders + 1" into this parameter by the parameter rounding (up).

When the number of cylinders which constructs the user space is specified in this parameter explicitly, the INIT sets zero into the "data block count" parameter in the block descriptor or transfers this page descriptor without specifying the block descriptor.

b. Head count

This parameter specifies the number of the data read/write heads of the disk drive (the number of servo heads is not included). The value of this parameter cannot be changed. When a zero or same value as the default value is specified, the IDD sets the same value as the default value by the parameter rounding.

c. Bytes 6 to 16

The parameters defined in bytes 6 to 16 of this page descriptor need not be explicitly specified for the IDD by the INIT. The IDD ignores the values specified in these parameters.

d. Rotational position locking (RPL)

This parameter specifies the spindle synchronization control. The definition is as follows.

RPL bit		Operation
1	0	
0	0	Spindle synchronization is inhibited.
0	1	The IDD performs the spindle synchronization according to the master signal from the external device. (slave mode)
1	0	The IDD sends the master signal to the external device for spindle synchronization (master mode)
1	1	not used (master control mode)

e. Medium rotational rate

This parameter specifies the disk rotational speed in rpm. The INIT cannot change this parameter. The IDD ignores the specified value in this field.

(5) Verify error recovery parameter (page code 7)

Figure 3.11 shows the format of the verify error recovery parameter in this page descriptor of the MODE SELECT command.

	Bit								
	7	6	5	4	3	2	1	0	
Byte 0	0	0	0	0	0	1	1	1	
1	X'0A' (page length)								
2	0	0	0	0	EER	PER	DTE	DCR	(See Note)
Default value	0	0	0	0	1	0/1	0	0	
Variable value	0	0	0	0	1	1	1	1	
3	Retry count at verify								
Default value	0	0	0	1	0	0	1	0	(= 18 times)
Variable value	1	1	1	1	1	1	1	1	
4	Correctable bit length								
Default value	0	0	0	0	1	0	0	0	(= 8 bits)
Variable value	0	0	0	0	0	0	0	0	
5–9	X'0000000000' (reserved)								
10–11	X'0000' (reserved)								

Figure 3.11 MODE SELECT parameter: Verify error recovery parameter

The error recovery parameter defined in this page descriptor is applied to the following commands:

- VERIFY
- WRITE AND VERIFY (verify operation)

Note:

The default value of the PER flag can be set to 1 or cleared to 0 by using the setting terminal on the IDD. This terminal-setup value must be the same as the default value of the PER flag contained in the read/write error recovery parameter (page 1). Refer to Subsection 5.3.2 of OEM Manual Specifications & Installation for an explanation of the setting terminal.

a. Error recovery flags

- EER: Enable early recovery
- PER: Post error
- DTE: Disable transfer on error
- DCR: Disable correction

The definitions and functions of these control flags are the same as those of the read/write error recovery parameter. See item (1) above for details.

b. Retry count at verify

This parameter specifies the number of times a retry is to be executed for a recovery from a data check error detected during a disk read operation. The "retry count" specified in this parameter is the maximum read retry count for the ID field and data field of each logical data block. If an error is detected, the IDD executes a read retry up to the specified number of times to recover the ID field of the data block. When data is not immediately corrected by ECC, the IDD executes a read retry up to the specified number of times to recover the data field of the data block. When zero is specified in this parameter, the read retry is inhibited. In this case, data field correction by ECC is executed according to the EER and DCR control flags.

c. Correctable bit length

This parameter indicates the maximum burst error length (bit length) that can be corrected by ECC. This value cannot be changed by the INIT. The IDD ignores this parameter value and operates according to the default value.

(6) Caching parameter (page code 8)

Figure 3.12 shows the format of the caching parameter in this page descriptor of the MODE SELECT command.

		Bit							
		7	6	5	4	3	2	1	0
Byte 0		0	0	0	0	1	0	0	0
1		X'0A' (page length)							
2		Reserved					WCE	MS	RCD
	Default value	0	0	0	0	0	0	0	0
	Variable value	0	0	0	0	0	1	0	1
3		X'00' (reserved)							
4–5		Prefetch-suppressed block count							
	Default value	X'FFFF'							
	Variable value	X'0000'							
6–7		Minimum prefetch block count							
	Default value	X'0000'							
	Variable value	X'0000'							
8–9		Maximum prefetch block count							
	Default value	X'xxxx' (60 KB)							
	Variable value	X'0000'							
10–11		Maximum prefetch restriction block count							
	Default value	X'FFFF'							
	Variable value	X'0000'							

Figure 3.12 MODE SELECT parameter: Caching parameter

The caching parameter in this page descriptor controls the range of the data to be preread by the Read-Ahead cache feature and controls caching operation validation/invalidation. See Sections 2.2 and 2.3 for details of the cache feature operation and the parameter setup.

a. Read cache disable (RCD)

1: Disables the operation of the Read-Ahead cache feature.

0: Enables the operation of the Read-Ahead cache feature.

b. Multiple selection (MS)

This bit specifies how to specify the "minimum prefetch count" (bytes 6 and 7) and "maximum prefetch count" (bytes 8 and 9) parameters in this page descriptor.

1: The "minimum prefetch count" and "maximum prefetch count" parameters indicate a multiplier. The number of data blocks to be prefetch is calculated that the value in the "transfer byte count" in the CDB specified by the READ or READ EXTENDED command X multiplier.

0: The "minimum prefetch count" and "maximum prefetch count" parameters indicate the data block count to be prefetched with the logical data block count.

This bit cannot be changed, and the IDD ignores this bit specification and operates according to the default value ('0').

c. Write cache enable (WCE)

1: Enables the operation of the write cache feature.

0: Disables the operation of the write cache feature.

d. Prefetch-suppressed block count

This parameter selectively inhibits the data prefetch operation that is to be caused by a READ or READ EXTENDED command. When the transfer block count specified in the CDB of the READ or READ EXTENDED command is greater than the value of this parameter, the data prefetch operation is not performed. This parameter cannot be changed, and the IDD ignores this parameter specification and operates according to the default value (X 'FFFF').

e. Minimum prefetch count

This parameter specifies the minimum amount of the logical data blocks to be prefetched into the data buffer by the READ or READ EXTENDED command.

This parameter cannot be changed, and the IDD ignores this parameter specification and operates according to the default value (X '0000').

f. Maximum prefetch count

This parameter specifies the maximum amount of the logical data blocks to be prefetched into the data buffer by the READ or READ EXTENDED command.

The data amount to be prefetched in the IDD is varied according to the data amount required by the READ or READ EXTENDED command, and is 60 KB in maximum. The number of blocks equivalent to 60 KB is always reported by the MODE SENSE command.

This parameter cannot be changed, and the IDD ignores this parameter specification.

g. Maximum prefetch restriction block count

This parameter specifies the maximum number of the logical data blocks to be prefetched into the data buffer by the READ or READ EXTENDED command. In the IDD, the maximum amount to be prefetched cannot be restricted.

This parameter cannot be changed, and the IDD ignores this parameter specification.

(7) Additional error recovery parameter (page code 21)

Figure 3.13 shows the format of the additional error recovery parameter in this page descriptor of the MODE SELECT command.

[Fujitsu unique parameter]

	Bit								
	7	6	5	4	3	2	1	0	
Byte 0	0	0	1	0	0	0	0	1	
1	X'02' (page length)								
2	DCED	PSER	RPR	0	Retry count at seek error occurrence				
Default value	0	0	0	0	0	0	1	0	(= 2 times)
Variable value	1	1	1	0	1	1	1	1	
3	RFJ	(reserved)							
Default value	0	0	0	0	0	0	0	0	
Variable value	0	0	0	0	0	0	0	0	

Figure 3.13 MODE SELECT parameter: Additional error recovery parameter

a. DCED: Disable command execution delay

When the IDD is executing internal processing such as automatic readjustment of positioning, a newly received command cannot be executed at once. The DCED bit specifies how to process such a command received from the INIT in the above case.

1: The IDD does not wait for the termination of the internal processing, and terminates the received command with the CHECK CONDITION status (ABORTED COMMAND [=B]/Command Execution Delay Required [=80-00]).

0: The IDD temporarily stacks the received command, and executes it after the internal processing is terminated.

Note:

When this bit is 1, the INIT can know the execution wait of the command that was generated asynchronously at an unexpected time. Even if the execution of the command is delayed because of the internal processing of the IDD, the command is executed normally after the internal processing is terminated. Therefore, this bit should be set to 0.

b. PSER: Post SCSI error

This bit specifies the reporting method of the error information when an error detected on the SCSI bus is corrected by the error recovery process of the IDD. For the error recovery process for the SCSI bus error, refer to Chapter 3 in OEM Manual SCSI Physical Specifications.

1: The CHECK CONDITION status is reported after the command execution terminates. The sense key of the sense data indicates "RECOVERED ERROR [=1]" and the information of the latest recovered error is reported.

0: The command being executed terminates with the GOOD status and the recovered error for the SCSI bus is not reported.

c. Rounded parameter report (RPR)

This bit specifies the operation when the rounding-up or -down is performed to the MODE SELECT parameter.

1: When rounding-up or -down is performed, the IDD reports the CHECK CONDITION status (RECOVERED ERROR [=1]/Rounded parameter [=37-00]).

0: Even if rounding-up or -down is performed, the IDD reports the GOOD status and terminates successfully.

d. Retry count at seek error occurrence

This parameter specifies the maximum number of times a positioning retry is to be executed at seek error occurrence. When 0 is specified, this retry is inhibited. The value of this parameter is applied to all the commands that causes a seek operation.

e. RFJ (Reserved by Fujitsu)

This bit is reserved by Fujitsu. This bit should be set to 0.

f. Byte 3 bit 6 to 0 (reserved)

This parameter is provided for matching with the conventional models, and is defined as an unchangeable parameter. The value specified in this parameter by the INIT is ignored. To the MODE SENSE or MODE SENSE EXTENDED command, default value X'00' is always sent.

(8) Reconnection timing parameter (page code 22)

Figure 3.14 shows the format of the reconnection timing parameter in this page descriptor of the MODE SELECT command.

[Fujitsu unique parameter]

		Bit							
		7	6	5	4	3	2	1	0
Byte 0		0	0	1	0	0	0	1	0
1		X'02' (page length)							
2		X'00' (reserved)							
	Default value	0	0	0	0	0	0	0	0
	Variable value	0	0	0	0	0	0	0	0
3		X'00' (reserved)							
	Default value	0	0	0	0	0	0	0	0
	Variable value	0	0	0	0	0	0	0	0

Figure 3.14 MODE SELECT parameter: Reconnection timing parameter

This page descriptor is provided for matching with the conventional models. No parameters in this descriptor are valid for the IDD. All the parameters in this descriptor are defined as changeable parameters. However, the values specified by the INIT are ignored. The IDD always sends default value X'00' to the MODE SENSE command.

3.1.5 MODE SENSE

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'1A'							
	1	LUN			0	DBD	0	0	0
	2	PC		Page code					
	3	0	0	0	0	0	0	0	0
	4	Transfer byte-length							
	5	0	0	0	0	0	0	Flag	Link

This command sends values and attributes of various parameters related to the physical attributes and data format of the disk drive or the SCSI bus disconnect/ reconnect process timing and error recovery procedure to INIT.

Data (MODE SENSE data) transferred by this command from IDD to INIT consists of the header, block discriptor, and one or more page discriptors describing various parameters as explained below.

When 1 is specified for the disable block discriptors (DBD) bit of CDB byte 1, the MODE SENSE data to be transferred to the INIT by this command does not contain the block discriptor. When 0 is specified for this bit, the IDD transfers the MODE SENSE data, containing the header, a block discriptor, and the specified page descriptor, to the INIT. When 1 is specified for this bit, the IDD transfers the MODE SENSE data, containing only the header and the specified page descriptor, to the INIT.

The "page code" field of CDB byte 2 specifies the page code of the page descriptor to be transferred to INIT by this command. The page code length depends on the SCSI level with the setting terminal. Types, page codes and length of the page descriptor supported by IDD are as follows:

Page code	Page descriptor	Byte-length	
		SCSI level	
		SCSI-2	SCSI-1/CCS
0	(No page descriptor is transferred.)		
1	Read/write error recovery parameter	12	8
2	Disconnect/reconnect parameter	16	12
3	Format parameter	24	24
4	Drive parameter	20	20
7	Verify error recovery parameter	12	12
8	Caching parameter	12	12
21	Additional error recovery parameter	4	4
22	Reconnection timing parameter	4	4
3F	All page descriptors supported by the IDD	108	72

When the specified value of the "page code" field is other than X'3F', the specified page descriptor only is transferred to INIT. When X'3F' is specified and the SCSI-2 mode is enable with setting on the IDD, all page descriptors supported by IDD are transferred to INIT in ascending order of page codes. When X'3F' is specified and the SCSI-1/CCS mode is enable with setting on the IDD, the page 7 and the page 8 are not transferred to INIT. When X'00' is specified for the "page code" field, the page descriptor is not transferred. When any other page code (i.e. page descriptor not supported by the IDD) is specified, the header (and the block descriptor) and the specified page descriptor with page length zero are transferred to INIT.

As shown in Table 3.2, the page control (PC) field of CDB byte 2 specifies the type of the parameter value in the page descriptor transferred to the INIT by this command.

Note:

Although Table 3.2 explains "the value specified with the MODE SELECT command" as a value reported to the INIT by this command, a value that is different from the value specified by INIT is reported when "parameter rounding" by the IDD is applied or for the "parameter" field or bit whose specified value is ignored. For details, see the explanation of each page descriptor of the MODE SELECT command (see Subsection 3.1.4).

For the byte position or bit position reserved in each page descriptor, 0 is reported.

Table 3.2 MODE SENSE data type setting

PC	Type of parameter to be transferred to INIT
00	Current value: The current value of each parameter is reported. The current value is either value explained below. ● Value specified by the latest MODE SELECT command that successfully terminated. ● Same value as the saved value when the MODE SELECT command has not been executed after power on, after the RESET condition, or after the BUS DEVICE RESET message from any INIT is received. However, this value is same as the default value for parameters other than page 3 and page 4 (except Byte 17) unless the saved value is present.
01	Changeable value: The "parameter" field and "parameter" bit that can be changed by the INIT are reported by the MODE SELECT command. For the field and bit position that can be changed in each page descriptor, 1 is reported. For the field and bit position that cannot be changed, 0 is reported. Refer to the explanation of each page descriptor in Subsection 3.1.4 for whether each parameter can be changed.
10	Default value: The default value of each parameter is reported. Refer to the explanation of each page descriptor in Subsection 3.1.4 for whether each parameter can be changed.
11	Saved value: The saved value of each parameter is reported. The saved value is either value explained below. ● For parameters other than page 3 and page 4 (except Byte 17), the value specified with the MODE SELECT command with the "SP" bit set to 1 that completed execution last. The value is same as the default value when the MODE SELECT command with the "SP" bit set to 1 has not been executed (i.e. the saved value is not present). ● For page 3 and page 4 (except Byte 17)parameters, the value saved on the disk when the FORMAT UNIT command is executed.

The "transfer byte-length" field of the specifies the total number of bytes of MODE SENSE data that can be transferred to the INIT with this command. The IDD transfers the number of bytes of all MODE SENSE data specified with the "page code" field or the MODE SENSE data that corresponds to the length specified with the "transfer byte-length" field, whichever is smaller, to the INIT. When 0 is specified for the "transfer byte-length", no data is transferred and this command terminates.

Figure 3.15 shows the data configuration of the parameter list (MODE SENSE data) transferred to the INIT by this command. The 4-byte header, 8-byte block descriptor, and one or more page descriptors specified with CDB are transferred in order as the parameter list. When 1 is specified for the DBD bit of CDB, the block descriptor is not transferred. When X'00' is specified for the "page code" field of CDB, the page descriptor is not transferred.

Header

	Bit 7	6	5	4	3	2	1	0
Byte 0	Sense data length							
1	X'00' (medium type)							
2	WP	0	0	0	0	0	0	0
3	X'00' or X'08' (block descriptor length)							

Block descriptor

	Bit 7	6	5	4	3	2	1	0
Byte 0	X'00'							
1	Number of data blocks (MSB)							
2	Number of data blocks							
3	Number of data blocks (LSB)							
4	X'00'							
5	Data block length (MSB)							
6	Data block length							
7	Data block length (LSB)							

Page descriptor

		Bit 7	6	5	4	3	2	1	0
Page header	Byte 0	PS	0	Page code					
Page header	1	Page length							
	2	Parameter field							
	≈	Parameter field							
	n	Parameter field							

Figure 3.15 MODE SENSE command parameter configuration

(1) Header

a. Sense data length

This field indicates the length (number of bytes) of the parameter list (MODE SENSE data) that can be transferred to the INIT by this command. This value does not include the length of the "sense data length" field itself. Regardless of the setting of the "transfer byte-length" field of CDB, the parameter list of the type specified with CDB that corresponds to the length supported by IDD is reported to this field. To check that all parameter lists requested by the command have been transferred, the INIT should check that the value indicated in this field plus the sense data length field length itself is smaller than the value specified for the "transfer byte-length" field of CDB.

b. Medium type

X'00' (default type) is always reported to this field.

c. WP bit

When this bit is 1, writing to the disk is disabled. When this bit is 0, the write operation is enabled. Disabling or enabling the write operation can be specified with the external operator panel which can be connected to the IDD. For details of the external operator panel, refer to Section 4.4 of <u>OEM Manual Specifications & Installation</u>. This write inhibit state set by the SET LIMIT command is not reflected to this bit.

d. Block descriptor length

This field indicates the length (byte-length) of the block descriptor that follows the header. This value does not include the page descriptor length. The IDD always reports X'08' to this field in order to indicate that a set of block descriptors follows the header when CDB of this command has specified 0 for the DBD bit. When 1 has been specified for the DBD bit of CDB, the value of this field is X'00'.

(2) Block descriptor

The 8-byte block descriptor indicates the logical attribute of the data format on the disk.

a. Number of data blocks

This field indicates the total number of logical data blocks (i.e. the block length is indicated in the "data block length" field) that are present in the user space on the disk. This value does not include the number of spare sectors reserved for the alternative block process.

When the default value transfer is specified by this command, the value of this field is X'000000' (this means the maximum logical data block count which is alignable in the user space).

b. Data block length

This field indicates the length (byte-length) of one logical data block on the disk.

(3) Page descriptor

The page descriptor consists of the 2-byte page header and the parameter field that follows, and is categorized in units of "page" for each functional attribute of the parameter. For configuration and contents of each page descriptor, see the explanation of the MODE SELECT command in Subsection 3.1.4.

a. PS bit

When this bit is 1, the parameter defined with the page descriptor can be saved on the disk. When this bit is 0, the parameter page that cannot be saved is indicated. All page descriptors supported by the IDD can be saved. For any page descriptor transferred by this command, 1 is indicated on this bit.

b. Page length

This field indicates the parameter field length (byte-length) on and after byte 2 excluding the page header of the page descriptor. The IDD always indicates the value same as the defined length of the page descriptor in this field regardless of the type of the parameter requested in the page control field of CDB, and reports all parameter fields of the page descriptor to byte 2 and following bytes.

c. Parameter field

The parameter field on and after byte 2 indicates the parameter value of the type (current value, changeable value, default value, or saved value) requested in the "page control" field of CDB. For the definition of each parameter and the default value and changeable value. see the explanation of the MODE SELECT command in Subsection 3.1.4.

3.1.6 NO OPERATION

[Fujitsu unique command]

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'0D'							
	1	LUN			0	0	0	0	0
	2	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	Flag	Link

This command executes disk drive status check only. When this command is executed while the IDD holds the sense data, sense data that is pending is not cleared.

When the IDD is ready and can be used from the INIT that issued this command, the GOOD status is reported to this command. When the IDD is not usable, the CHECK CONDITION status is reported to this command and the sense data indicates the current IDD state.

3.1.7 REZERO UNIT

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'01'							
	1	LUN			0	0	0	0	0
	2	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	Flag	Link

This command moves the read/write head of the disk drive to the initial position. A data block whose logical block address is zero is present at the initial position (cylinder 0/track 0). The initialization and the automatic readjustment of the positioning control of the disk drive are performed by this command.

3.1.8 START/STOP UNIT

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'1B'							
	1	LUN			0	0	0	0	Immed
	2	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	×	Start
	5	0	0	0	0	0	0	Flag	Link

This command controls disk drive spindle motor start and stop.

The spindle motor is controlled with the start bit of bit 0 in CDB byte 4. When this command is issued with the start bit set to 0, the spindle motor can be stopped. When 1 is specified for the start bit, the spindle motor can be started.

Bit 1 [LoEj (Load/Eject)] of CDB byte 4 is the control bit for the removable medium unit only and is not significant for the IDD. The IDD ignores the value specified for this bit and controls start and stop of the spindle motor according to setting of the start bit only.

The spindle motor start mode of the disk drive is selectable with the setting terminal on the IDD. When the setting terminal is set to disable motor start control with this command, the spindle motor automatically starts when the power to the IDD is turned on. When the setting terminal is set to enable motor start control with this command, the spindle motor does not start even though the power to the IDD is turned on. In this case, INIT should start the spindle motor by issuing this command. For the setting terminal, refer to Subsection 5.3.2 of OEM Manual Specifications & Installation.

Note:

The setting terminal only specifies the spindle motor starting method upon power-on. This command is valid when either mode has been set.

Regardless of the setting terminal state, INIT can issue this command at any timing to stop or start the spindle motor.

The termination timing of this command (status byte report) is dependent upon the value specified for the Immed (Immediate) bit of bit 0 in CDB byte 1 as explained below.

a. When start is specified (start bit = 1)

- When the Immed bit is 1, the GOOD status is reported and command execution is terminated without waiting the disk drive becomes ready after being instructed to start the spindle motor.

- When the Immed bit is 0, the status byte is reported and command execution is terminated at the point the disk drive becomes ready after the spindle motor is started.

b. When stop is specified (start bit = 0)

- When the Immed bit is 1, the GOOD status is reported and command execution is terminated after spindle motor stop is issued.

- When the Immed bit is 0, the status byte is reported and command execution is terminated after the spindle motor stops.

Notes:

1. When another command is linked and issued with this command that specifies start, the Immed bit should be set to 0. When the Immed bit is set to 1, the disk drive is not ready in general at the time that this command terminates. Therefore the linked command cannot be successfully executed.

2. When another command being stacked or executed is present even though the Immed bit has been set to 1 with this command, this command is stacked and the disconnect process is performed. The status is not always reported immediately after the COMMAND phase even though 1 is specified for the Immed bit.

3.1.9 RESERVE UNIT

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'16'							
	1	LUN			3rd Pty	3rd Pty Dev ID			0
	2	×	×	×	×	×	×	×	×
	3	×	×	×	×	×	×	×	×
	4	×	×	×	×	×	×	×	×
	5	0	0	0	0	0	0	Flag	Link

This command controls exclusive access of the logical unit (IDD) in the multiple initiator environment along with the RELEASE UNIT command.

With this command, the IDD is reserved by the INIT which issued this command or for another SCSI device specified on CDB.

The INIT that issues this command must send the SCSI ID of the own INIT upon execution of the SELECTION phase. When the INIT ID cannot be identified, this command is not executed but terminates with the CHECK CONDITION status (ILLEGAL REQUEST [=5]/Initiator's SCSI ID not identified [=90-00]). Values specified in CDB bytes 2 to 4 are meaningless and ignored, but the INIT should specify X'00' for these bytes.

(1) Logical unit reserve function

This command reserves the entire IDD (logical unit) for a particular SCSI device. The reserved state established by this command is kept until one of the following conditions occur.

1) Reservation condition change by the INIT which issued this command (superseding reserve)

2) Reservation releasing with the RELEASE UNIT command from the INIT which issued this command

3) PRIORITY RESERVE command execution by any INIT

4) BUS DEVICE RESET message from any INIT

5) RESET condition

6) IDD power off/on

When this command is issued from an INIT which does not have the "reservation right" for the IDD while the IDD has been reserved for any SCSI device, this command terminates with the RESERVATION CONFLICT status. For details of the reservation right, see the Note below.

After the reserve state is established, the command other than the INQUIRY, REQUEST SENSE, PRIORITY RESERVE, and RELEASE UNIT commands, issued by an INIT other than the SCSI device which reserved the IDD are rejected. The RESERVATION CONFLICT status is reported to the INIT which issued the command. The INQUIRY, REQUEST SENSE, or PRIORITY RESERVE command is successfully executed even though the IDD has been reserved for another SCSI device. Although the RELEASE UNIT command terminates with the GOOD status, the release operation request by the RELEASE UNIT command issued from a SCSI device which does not have the reservation right for the IDD is ignored.

(2) Reservation right and third party reserve function

When the "3rd Pty" bit of CDB byte 1 is 0, the IDD is reserved for the INIT which issued this command and this INIT has the reservation right for the IDD.

When the "3rd Pty" bit is 1, the third party reserve function is specified. The INIT which issued this command by specifying the third party reserve function can reserve the IDD for another SCSI device. In this case, this command reserves the IDD for the SCSI device with the SCSI ID specified in the "3rd Pty Dev ID" field of CDB byte 1 (called the third party device). The reservation right for the IDD belongs to the INIT which issued this command even though the IDD is reserved for another SCSI device by using the third party reserve function. To reset the reserved state, the INIT that issued this command should issue the RELEASE UNIT command (see Subsection 3.1.12) which specifies the third party release function. The condition to maintain the reserved state established by the third party reserve function is the same as that provided when the third party reserve function is not used. (See item (1).)

When the reserved state is established by the third party reserve function, the current value of the MODE SELECT parameter of the INIT that issued the RESERVE UNIT command is copied to the current value for the third party device. As a result, commands that are issued by the third party device are executed according to the MODE SELECT parameter specified by the INIT which issued the RESERVE UNIT command. When the MODE SELECT parameter is duplicated in the third party reserve function, the IDD generates the UNIT ATTENTION condition (Mode parameter changed [=2A-01]) for the third party device. The IDD retains the MODE SELECT parameter related to the error recovery procedure, etc. for each INIT. With this parameter copy function, INIT that issues the RESERVE UNIT command can explicitly specify the MODE SELECT parameter for the third party device. When the third party device issues the MODE SELECT command later, the MODE SELECT parameter is changed to the value specified by the third party device.

(3) Reservation condition change (superseding reserve)

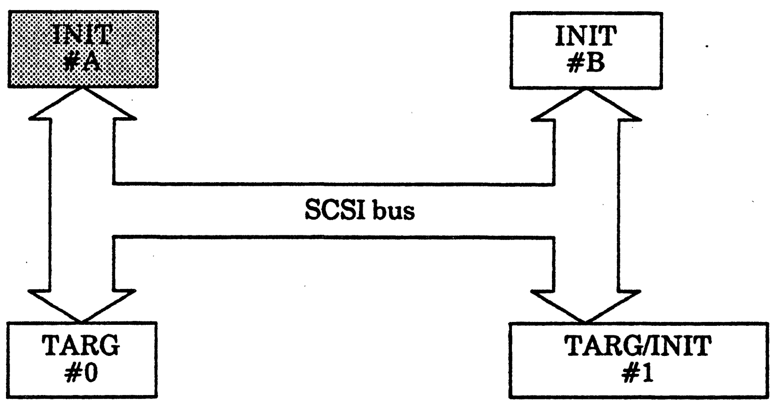
The INIT that has the reservation right for the IDD (i.e., the INIT that established the IDD reserved state by issuing this command) can change the IDD reservation condition by issuing another RESERVE UNIT command. (Superseding reserve)

When a superseding reserve is executed, the IDD resets the existing reserved state and establishes the new reserved state according to the new setting of this command.

By using this function, the INIT can change the SCSI device (SCSI ID) that reserves the logical unit while holding the reserved state for the logical unit for which the reserved state was established by using the third party reserve function.

Notes: Reservation right and third party reserve function

To clarify the extent of reserve and release, this manual uses the "reservation right".



1. When INIT#A issues the RESERVE UNIT command without specifying the third party reserve function to TARG#0, TARG#0 is reserved by INIT#A and the reservation right for TARG#0 is retained by INIT#A. In this case;

   - INIT#A exclusively occupies TARG#0.

   - Commands, except for the INQUIRY, REQUEST SENSE, PRIORITY RESERVE, and RELEASE UNIT commands, issued from another SCSI device (INIT#B, TARG/INIT#1) to TARG#0 are rejected with the RESERVATION CONFLICT status. The RELEASE UNIT command terminates successfully, but the TARG#0 reserved state is not affected.

2. When INIT#A issues the RESERVE UNIT command with the third party reserve function ("3rd Pty Dev ID"=TARG/INIT#1) to TARG#0, TARG#0 is reserved by TARG/INIT#1, but the reservation right for TARG#0 is retained by INIT#A. In this case;

   - TARG/INIT#1 occupies TARG#0 exclusively. When TARG/INIT#1 issues the RESERVE UNIT command, the command is rejected with the RESERVATION CONFLICT status. The RELEASE UNIT command terminates successfully, but the TARG#0 reserved state is not affected.

   - INIT#A can issue the INQUIRY, REQUEST SENSE, PRIORITY RESERVE, RELEASE UNIT, and RESERVE UNIT commands to TARG#0, but other commands are rejected with the RESERVATION CONFLICT status. The RELEASE UNIT command issued by INIT#A resets the reserved state of TARG#0. The RESERVE UNIT command changes the reserved state of TARG#0.

   - Commands, except for the INQUIRY, REQUEST SENSE, PRIORITY RESERVE, and RELEASE UNIT commands, issued from INIT#B to TARG#0 are rejected with the RESERVATION CONFLICT status. The RELEASE UNIT command terminates successfully, but the TARG#0 reserved state is not affected.

3. The example of the third party reserve function in Note 2 above is useful when the COPY command is used. For instance, when TARG/INIT#1 has the COPY command and executes data transfer between TARG#0 and TARG/INIT#1 by using the COPY command, execution of third party reserve in item 2 before INIT#A issues the COPY command to TARG/INIT#1 can disable access of INIT#B to TARG#0 during execution of the COPY command. Third party reserve in Note 2 copies the MODE SELECT parameter specified by INIT#A to the current value for TARG/INIT#1. INIT#A can therefore explicitly specify the operation parameter for access of TARG/INIT#1 to TARG#0 during execution of the COPY command.

3.1.10 RELEASE UNIT

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'17'							
	1	LUN			3rd Pty	3rd Pty Dev ID			0
	2	×	×	×	×	×	×	×	×
	3	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	Flag	Link

This command resets the reserved state of the IDD for the INIT which issued this command.

When the reserved state related to the INIT that issued this command, or the reserved state of release object type specified with CDB of this command is not present in the IDD, or the IDD is reserved by another SCSI device, this command terminates successfully with the GOOD status but the IDD reserved state is not affected at all.

The INIT that issues this command must send the SCSI ID of the INIT upon execution of the SELECTION phase. When the INIT ID cannot be identified, this command is not executed but terminates with the CHECK CONDITION status (ILLEGAL REQUEST [=5]/Initiator's SCSI ID not identified [=90-00]). The value specified for CDB byte 2 is meaningless and ignored, but INIT should specify X'00' for this byte.

(1) Release function

This command resets the reserved state of the entire IDD (logical unit) for which the INIT that issued this command has the reservation right. For the definition of reservation right, see the Note in Subsection 3.1.9.

(2) Release object and third party release function

When the "3rd Pty" bit of CDB byte 1 is 0, this command resets the reserved state of the IDD for which reservation was made with the RESERVE UNIT command without the third party reserve function by the INIT which issued this command.

When the "3rd Pty" bit is 1, the third party release function is specified. This command can reset the reserved state that was established by the third party reserve function. When the third party release function is specified, this command resets the reserved state only when the INIT that issued this command had reserved the IDD for the same SCSI device as that (third party device) which had the SCSI ID specified in the "3rd Pty Dev ID" field of CDB byte 1 in this command with the RESERVE UNIT command with the third party reserve function.

Note:

The RESERVE UNIT command that specified the third party reserve function changes the MODE SELECT parameter for the third party device to the value which is equal to the parameter for INIT that issued the RESERVE UNIT command. The MODE SELECT parameter value is not changed even though the reserved state is reset by the third party release function (the parameter value cannot be restored to the original value).

3.1.11 PRIORITY RESERVE

[Fujitsu unique command]

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'14'							
	1	LUN			0	0	0	0	0
	2	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	Flag	Link

This command forcibly reserves the entire IDD (logical unit) for the INIT which issued this command, even if the IDD has been reserved for another SCSI device or is being used by another SCSI device.

The INIT that issues this command must send the SCSI ID of the INIT upon execution of the SELECTION phase. When the INIT ID cannot be identified, this command is not executed but terminates with the CHECK CONDITION status (ILLEGAL REQUEST [=5]/Initiator's SCSI ID not identified [=90-00]).

This command is used for the hardware error recovery process, etc. in the multiple initiator environment. For instance, when an INIT falls in an unrecoverable state while reserving the IDD, another INIT can forcibly recover the access right for the IDD by using this command.

With execution of this command, the IDD is reserved by the INIT that issued this command. The IDD operation and reserved state resetting condition after the reserved state is established are the same as those for the RESERVE UNIT command (see Subsection 3.1.9).

When the IDD holds the UNIT ATTENTION condition for the INIT that issued this command, this command is successfully executed and the UNIT ATTENTION condition is cleared.

This command is not stacked but immediately executed. In this case, all commands that are being stacked or executed are cleared and the reconnection request and status report for the INITs that issued those commands are not made. The IDD generates the sense data indicating "ABORTED COMMAND [=B] / Command aborted by PRIORITY RESERVE [=88-00]" for each command cleared by this command. When the IDD has been reserved for an SCSI device other than the INIT that issued this command, the reserved state is unconditionally cleared and the IDD is reserved by the INIT which issued this command.

Note:

When this command is issued, any command being executed by the IDD is unconditionally cleared and the execution result is not ensured. For details of the resetting process with this command, see Subsection 1.7.6.

3.1.12 REQUEST SENSE

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'03'							
	1	LUN			0	0	0	0	0
	2	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0
	3	Transfer byte-length							
	5	0	0	0	0	0	0	Flag	Link

This command transfers sense data to an INIT. This command is not stacked and immediately executed.

The IDD sense data consists of 48 bytes. The "transfer byte-length" field of CDB indicates the number of bytes that can be received by an INIT. The IDD transfers the sense data with one of the following methods by the setting terminal specification (SCSI level). For the setting terminal, refer to Subsection 5.3.2 in OEM Manual Specification & Installation.

- When the SCSI-2 mode is specified by the setting terminal, the IDD transfers the sense data with the number of bytes specified in the "transfer byte-length" field or of the sense data length (48 bytes) retained by the IDD, whichever is smaller, to the INIT. When zero is specified in the "transfer byte-length" field, no sense data is transferred and this command is terminated. In this case, the sense data held by the IDD is cleared.

- When the SCSI-1/CCS mode is specified by the setting terminal, the IDD transfers the leading 4 bytes of the sense data if zero is specified in the "transfer byte-length" field. When the number other than zero is specified in the "transfer byte-length" field, the IDD transfers the sense data with the number of bytes specified in the "transfer byte-length" field or of the sense data length (48 bytes) retained by the IDD, whichever is smaller, to the INIT.

When the command execution terminates abnormally, the IDD generates a sense data for the INIT that issued the command and enters the sense data pending state as described in Section 1.6.

The sense data being held is valid until the pending INIT reads it with the REQUEST SENSE command or until the INIT issues a command other than the NO OPERATION command to the IDD. The sense data being held is cleared by the ABORT message from the pending INIT, BUS DEVICE RESET message from any INIT, or by the RESET condition.

INIT should issue this command to take out the sense data when a command terminates with the CHECK CONDITION status, when the SCSI bus inadvertently enters the BUS FREE phase during command execution, or when no reconnection request from the IDD is detected.

Note:

> For details of the sense data pending state, see Section 1.6. For the format and contents of sense data to be transferred from IDD to INIT with this command, see Section 4.1.

When this command is issued while the IDD holds the UNIT ATTENTION condition, this command sends the sense data indicating the UNIT ATTENTION condition to INIT and the UNIT ATTENTION condition is cleared unless the IDD holds the sense data at that point. For details, see Section 1.5.

When this command is issued while valid sense data is not held, the IDD sends sense data with the sense key indicating "NO SENSE [=0]" and the sense code indicating "No additional sense information [00-00]" to the INIT.

This command reports the CHECK CONDITION status and terminates abnormally only when one of the following conditions is detected. In this case, new sense data is generated and sense data that has been held is lost.

- When setting other than the LUN field of CDB is incorrect
- When sense data cannot be sent due to an IDD hardware error
- When an unrecoverable error is detected on the SCSI bus
- When the exception conditions of the overlap command are applied (see Subsection 1.7.1).

In any other case, this command sends the sense data indicating the error state to the INIT and terminates with the GOOD status when an error was detected while this command was being executed whether or not sense data was being held. When an error that was recovered by retry is detected during execution of this command and the mode reporting RECOVERED ERROR is specified, this command sends the sense data to the INIT then terminates with the CHECK CONDITION status. Sense data indicating the error contents (RECOVERED ERROR [=1]) is generated.

3.2 Data Access Commands

3.2.1 READ

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'08'							
	1	LUN			Logical block address (MSB)				
	2	Logical block address							
	3	Logical block address (LSB)							
	4	Transfer block count							
	5	0	0	0	0	0	0	Flag	Link

This command reads the continuous logical data blocks corresponding to the number of blocks specified in the "transfer block count" field, beginning with the logical data block on the disk specified in the "logical block address" field of CDB, and transfers the data to the INIT.

For the number of transfer blocks, a maximum of 256 logical data blocks can be specified. When the "transfer block count" field in CDB byte 4 is set to zero, transfer of 256 logical data blocks is specified. When the "transfer block count" field is set to other than zero, the specified number of logical data blocks are transferred.

When this command specifies transfer of two or more data blocks, and the data block to be processed reaches the track boundary or cylinder boundary, the head switch or cylinder switch is automatically executed and the corresponding blocks are read.

When settings in the "logical block address" field of CDB and "transfer block count" field exceeds the maximum logical block addresses on the IDD, the command terminates with the CHECK CONDITION status (ILLEGAL REQUEST [=5]/ Logical block address out of range [=21-00]) and disk read operation is not executed.

The error recovery process can be specified with the MODE SELECT parameter while this command is being executed. Unless the retry process and data correction process are disabled, data transferred to the INIT by this command is free of errors when this command successfully terminates or when this command terminates by reporting the sense key of RECOVERED ERROR [=1]. When correctable data check is detected, the IDD automatically corrects the data error on the data buffer, then sends the data to the INIT.

When the disconnect process has been enabled, the start timing of the reconnection process to execute data transfer on the SCSI bus can be specified with the MODE SELECT parameter (buffer full ratio). If operation of the Read-Ahead cache feature has been enabled, this command performs the caching operation using the IDD data buffer. For details of the data buffer operation and Read-Ahead cache feature, see Chapter 2.

3.2.2 READ EXTENDED

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'28'							
	1	LUN			×	×	0	0	0
	2	Logical block address (MSB)							
	3	Logical block address							
	4	Logical block address							
	5	Logical block address (LSB)							
	6	0	0	0	0	0	0	0	0
	7	Transfer block count (MSB)							
	8	Transfer block count (LSB)							
	9	0	0	0	0	0	0	Flag	Link

This command reads the continuous logical data blocks corresponding to the number of blocks specified in the "transfer block count" field, beginning with the logical data block on the disk specified in the "logical block address" field of CDB, and transfers the data to the INIT.

Functions of this command are same as those of the READ command of group 0 except that the 4-byte logical block address and the number of transferred blocks in 2 bytes can be specified. When zero is specified in the "transfer block count" field, the seek operation to the cylinder/track where the data block specified in the "logical block address" field is present and the read ahead cache operation are executed (if read ahead cache is enable).

Settings in bits 4 and 3 of CDB byte 1 are invalid for the IDD, and specified values are ignored.

3.2.3 WRITE

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'0A'							
	1	LUN			Logical block address (MSB)				
	2	Logical block address							
	3	Logical block address (LSB)							
	4	Transfer block count							
	5	0	0	0	0	0	0	Flag	Link

This command writes data transferred from the INIT in the continuous logical data blocks on the disk, beginning with the logical data block specified in the "logical block address" field of CDB.

For the number of transferred blocks, a maximum of 256 logical data blocks can be specified. When the "transfer block count" field in CDB byte 4 is set to zero, transfer of 256 logical data blocks is specified. When the "transfer block count" field is set to other than zero, the number of logical data blocks are transferred.

When this command specifies transfer of two or more data blocks, and the data block to be processed reaches the track boundary or cylinder boundary, head switching or cylinder switching is automatically executed and the specified blocks are written.

When settings of the "logical block address" field of CDB and "transfer block count" field exceeds the maximum logical block address on the IDD, the command terminates with the CHECK CONDITION status (ILLEGAL REQUEST [=5]/ Logical block address out of range [=21-00]) and disk write operation is not performed.

The IDD executes data transfer from the INIT to the data buffer (data prefetch) immediately after CDB is received. When the disconnect process has been enabled, it is executed at the point all data transfer specified with the command terminates or there is no space in the data buffer. Positioning operation to the specified data block is performed concurrently with this data transfer. The write process from the data buffer to the data block on the disk is performed immediately after positioning is complete. For details of the data buffer operation and the reconnection process start timing control with the MODE SELECT parameter (buffer empty ratio), see Section 2.1.

When the write cache feature is enabled, the IDD reports the status byte at completion of receiving all data transferred from the INIT. When an error occurs during writing data on the disk, the IDD reports the CHECK CONDITION status for the next command. When the write cache feature is disabled, the IDD reports the status byte after writing all data transferred from the INIT and terminates the command execution.

Note:

When the setting on CDB is incorrect or when the write operation onto the disk cannot be successfully executed for some reason, data transfer from the INIT to the IDD (data prefetch to the data buffer) may be executed. In this case, the length of data transferred from the INIT to the IDD is undefined. All data transferred to the IDD may not be actually written on the disk. When the command terminates with the CHECK CONDITION status and the sense key of sense data indicates ILLEGAL REQUEST [=5], the command has not performed the data write operation onto the disk.

3.2.4 WRITE EXTENDED

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'2A'							
	1	LUN			×	×	0	0	0
	2	Logical block address (MSB)							
	3	Logical block address							
	4	Logical block address							
	5	Logical block address (LSB)							
	6	0	0	0	0	0	0	0	0
	7	Transfer block count (MSB)							
	8	Transfer block count (LSB)							
	9	0	0	0	0	0	0	Flag	Link

This command writes data transferred from the INIT to the continuous logical data blocks on the disk, beginning with the logical data block specified in the "logical block address" field of CDB.

The functions of this command are the same as those of the WRITE command of group 0 (see Subsection 3.2.3) except that the 4-byte logical block address and the number of transferred blocks in 2 bytes can be specified. When the number of transfer blocks is set to zero, only the seek operation to the cylinder/track where the data block specified in the "logical block address" field is present is executed.

The settings in bit 4 and bit 3 of CDB byte 1 are invalid for the IDD and the specified values are ignored.

3.2.5 WRITE AND VERIFY

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'2E'							
	1	LUN			×	0	0	0	0
	2	Logical block address (MSB)							
	3	Logical block address							
	4	Logical block address							
	5	Logical block address (LSB)							
	6	0	0	0	0	0	0	0	0
	7	Block count (MSB)							
	8	Block count (LSB)							
	9	0	0	0	0	0	0	Flag	Link

This command writes data transferred from the INIT in the continuous logical data blocks on the disk corresponding to the number of data blocks specified in the "block count" field, beginning with the logical data block specified in the "logical block address" field of CDB, then reads and verifies the data.

The functions of this command related to the write operation are the same as those of the WRITE EXTENDED command except for disabled the write caching operation (see Subsection 3.2.4). When zero is specified in the "block count" field of CDB, this command executes only the seek operation to the cylinder/track where the logical data block specified in the "logical block address" field is present.

The setting in bit 4 of CDB byte 1 is invalid for the IDD and the specified value is ignored.

The verification to be executed with this command is only checking validity of CRC (ID field) and ECC (data field). The error recovery process upon verification complies with the mode specified by the MODE SELECT parameter (verify error recovery parameter). For instance, verification is assumed to have been successful even though correctable data check is detected in verification unless the data correction process is disabled.

3.2.6 VERIFY

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'2F'							
	1	LUN			×	0	0	BytChk	0
	2	Logical block address (MSB)							
	3	Logical block address							
	4	Logical block address							
	5	Logical block address (LSB)							
	6	0	0	0	0	0	0	0	0
	7	Block count (MSB)							
	8	Block count (LSB)							
	9	0	0	0	0	0	0	Flag	Link

This command reads and verifies the continuous logical data blocks corresponding to the number of data blocks specified in the "block count" field, beginning with the logical data block on the disk specified in the "logical block address" field of CDB. The data transfer is not performed with this command.

The "block count" field of CDB specifies the number of data blocks to be verified. When the "block count" field is set to zero, this command executes only the seek operation to the cylinder/track where the logical data block specified in the "logical block address" field is present.

The setting in bit 4 of CDB byte 1 is invalid for the IDD and the specified value is ignored.

When the BytChk bit is specified '1' , the IDD compares the data written on the drive and the data transferred from the INIT. If the comparison is unsuccessful for any reason, the IDD returns CHECK CONDITION status (MISCOMPARE [=E]/Miscompare during verify operation [=1D-00]).

When the BytChk bit is specified '0', the verification to be executed with this mode is only checking validity of CRC (ID field) and ECC (data field). The error recovery process upon verification complies with the mode specified by the MODE SELECT parameter (verify error recovery parameter). For instance, verification is assumed to have been successful even though correctable data check is detected in verification unless the data correction process is disabled.

3.2.7 SEEK

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'0B'							
	1	LUN			Logical block address (MSB)				
	2	Logical block address							
	3	Logical block address (LSB)							
	4	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	Flag	Link

This command executes the seek operation to the cylinder/track where the logical data block specified in the "logical block address" field of CDB is present.

When the disconnect process has been enabled, the IDD performs the disconnect process after the CDB is received. The IDD executes the reconnect process at the point the seek operation terminates and then reports the status.

When the disconnect process has not been enabled, the IDD reports the GOOD status and terminates this command immediately after the CDB is received and the seek operation is started unless linking is specified for this command.

When linking is specified for this command even though the disconnect process has not been enabled, the IDD executes the seek operation in the state that the IDD is connected with the SCSI bus. After the completion of the seek operation, the IDD reports the status of this command and executes the linked next command.

3.2.8 SEEK EXTENDED

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'2B'							
	1	LUN			0	0	0	0	0
	2	Logical block address (MSB)							
	3	Logical block address							
	4	Logical block address							
	5	Logical block address (LSB)							
	6	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	Flag	Link

This command executes the seek operation to the cylinder/track where the logical data block specified in the "logical block address" field of CDB is present.

The functions and operation of this command are the same as those of the SEEK command of group 0 (see Subsection 3.2.7) except that the 4-byte logical block address can be specified.

3.2.9 SET LIMITS

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'33'							
	1	LUN			0	0	0	RdInh	WrInh
	2	Logical block address (MSB)							
	3	Logical block address							
	4	Logical block address							
	5	Logical block address (LSB)							
	6	0	0	0	0	0	0	0	0
	7	Number of blocks (MSB)							
	8	Number of blocks (LSB)							
	9	0	0	0	0	0	0	Flag	Link

This command specifies the address range of the logical data block on the IDD that is accessible with the command linked following to this command and the type of operation that can be executed. This command can be issued only once among a series of linked commands.

Note:

> Setting of this command is valid only for the series of linked commands following to this command. When the command linkage is broken, the setting is invalid.

The "logical block address" field of CDB specifies the address of the logical data block as the starting point of the accessible range. When the user space address (X'00000000' and later) is specified in the "logical block address" field, access to only the user space is enabled for commands linked following to this command, and access to the CE space is disabled. On the other hand, when the CE space address (X'80000000' and later) is specified, access to only the CE space is enabled for commands linked following to this command, and access to the user space is disabled.

The size of the accessible range to be specified with this command is specified in the "block count" field of CDB by using the number of logical data blocks from the starting point. When the "block count" field is set to zero, access to the specified data space (user space or CE space) that begins with the logical data block specified in the "logical block address" field and ends with the last logical data block is enabled.

Figure 3.16 shows how to specify the accessible range with this command.

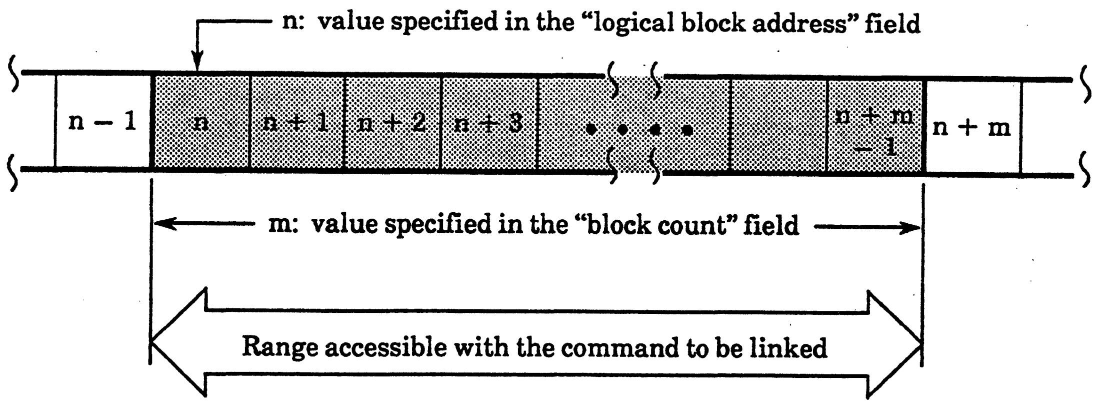


Figure 3.16 SET LIMITS command: accessible range setting

When 1 is specified for RdInh (Read Inhibit) of bit 1 or WrInh (Write Inhibit) of bit 0 of CDB byte 1, the read or write operation is disabled for commands linked following to this command as follows.

RdInh	WrInh	Operation restriction
0	0	Enable read/write operation in the specified range.
0	1	Enable read operation only in the specified range.
1	0	Enable write operation only in the specified range.
1	1	Disable read/write operation. Enable access with the SEEK or SEEK EXTENDED command in the specified range only.

A command linked following to this command specifies access to the logical data block not contained in the address range defined with this command or specifies the disabled type of access operation, that command terminates with the CHECK CONDITION status (DATA PROTECT [=7]/Write protected [=27-00]: when the WrInh flag is violated, or DATA PROTECT [=7]/No additional sense information [=00-00]: when the RdInh flag is violated). When this command is issued again among a series of linked commands, the second SET LIMITS command is rejected and the CHECK CONDITION status (DATA PROTECT [=7]/Command sequence error [=2C-00]) is reported.

Notes:

1. When 1 is specified for the RdInh or WrInh flag, the following commands are restricted in the read or write operation.

   "RdInh":
   - READ
   - READ EXTENDED (*1)
   - READ LONG
   - VERIFY (*1)
   - WRITE AND VERIFY (*1)

   "WrInh":
   - FORMAT UNIT
   - REASSIGN BLOCKS
   - WRITE
   - WRITE AND VERIFY (*1)
   - WRITE EXTENDED (*1)
   - WRITE LONG
   - WRITE SAME

   *1 The command is not executed when zero is specified as the number of processed blocks.

2. The WP (write protected) bit to be reported to the INIT with the MODE SENSE command indicates 0 even though the write operation is disabled with this command.

3. For the FORMAT UNIT or REASSIGN BLOCKS command linked and executed following to this command, execution is enabled if 0 is specified for the WrInh flag with this command. (Setting of the accessible range is not applied.)

3.3 Format Commands

3.3.1 FORMAT UNIT

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'04'							
	1	LUN			FmtData	CmpLst	Defect List Format		
	2	Initializing data pattern							
	3	Interleave factor (MSB)							
	4	Interleave factor (LSB)							
	5	0	0	0	0	0	0	Flag	Link

This command initializes (formats) all areas (user space and CE space) of the disk that are accessible from the INIT. At the same time, the IDD implements the defect management process such as alternate block assignment for defects on the disk according to the specification of this command.

To initialize the disk by changing the following format attributes, the INIT must issue the MODE SELECT command to specify the format attributes before issuing this command.

- Logical data block length
- Number of logical data blocks
- Number of cylinders in the user space
- Number of spare sectors for alternate block

(1) Defect lists

The following defect lists have been defined to register or specify the defect position on the disk for the defect management process that can be specified from the INIT with this command.

a. P list: primary defect list

The P list registers the defect position information (permanent defects) at disk drive shipment. This list has been recorded in an area on the disk that is not explicitly accessible from the INIT. The INIT can reference this list with the READ DEFECT DATA command only, but cannot change or erase it.

b. D list: data defect list

The defect information of this list is the defect position information to be transferred from the INIT upon execution of this command. The IDD registers this defect information on the disk as the G list.

c. C list: target certification list

This defect list is the position information of defective data blocks that are detected during the data block verification (certification) after initialization upon execution of the FORMAT UNIT command. The IDD generates this list internally upon execution of the FORMAT UNIT command and registers the data to the G list.

d. G list: growth defect list

The defect information of this list is the defect position information specified from the INIT and the position information of defective data blocks detected by the IDD itself. This defect list does not include the P list. The IDD records the G list in an area on the disk that is not explicitly accessible from the INIT. The INIT can reference this list with the READ DEFECT DATA command. The G list contains the following defect position information:

- Defect information transferred as the D list from the INIT with this command
- Defect information (C list) detected during verification upon execution of this command
- Defect information specified by the INIT with the REASSIGN BLOCKS command
- Defect information of defective data blocks that were detected by the IDD and assigned to alternate blocks when the automatic alternate block assignment process was enabled

(2) Specifying initialing procedure

With the FmtData (format data) bit, CmpLst (complete list) bit, in CDB byte 1 or the "defect list format" field, INIT can specify the defect processing method to be executed with this command.

When 1 is specified for the FmtData bit, format parameters (header and defect list) explained later are transferred from the INIT upon execution of this command. When 0 is specified for this bit, format parameters are not transferred.

When 1 is specified for the CmpLst bit, the current G list is replaced with the defect list (D list) to be transferred from the INIT upon execution of this command. When 0 is specified for this bit, the contents of the D list are added to the current G list.

The "defect list format" field specifies the format of the defect list (D list) to be transferred from the INIT when the FmtData bit is '1'. For the defect list, one of the following formats can be specified:

Defect	List	Format	D list format
0	0	0	Block address format
1	0	0	Format of byte distance from index
1	0	1	Physical sector address format

The "initializing data pattern" field (Fijitsu unique parameter) of CDB byte 2 specifies the data pattern used at initialization of the disk by this command. The IDD initializes the disk by writing the data specified by this field onto all bytes in the data field of all logical blocks which is accessible from the INIT. When X'00' is specified in this field, X'5B' is used as the default initializing data pattern.

The "interleave factor" field of CDB specifies the method of logical data block allocation for physical sectors on the disk. The IDD continuously allocates logical data blocks in the physically continuous sectors without applying sector interleave even though any value is specified in this field.

(3) Format parameters

Figure 3.17 shows the data formats of the format parameters to be transferred from the INIT when 1 is specified for the FmtData bit of CDB.

Header

	Bit 7	6	5	4	3	2	1	0
Byte 0	X'00'							
1	FOV	DPRY	DCRT	STPF	0	0	0	0
	0	0	0	0				
	1	0/1	0/1	×				
2	Defect list length (MSB)							
3	Defect list length (LSB)							

Defect list (D list)

	Bit 7	6	5	4	3	2	1	0
Byte 0	Defect descriptor 0							
1								
≈								
x								
≈								
xx	Defect descriptor n							
xx+x								
≈								
xx+1								

Figure 3.17 FORMAT UNIT command parameter list configuration

a. Header

The leading of the format parameter to be transferred from the INIT is a 4-byte header. With the control flag in the header, INIT can specify the defect processing method to be executed with this command.

(a) FOV (format option valid)

0: INIT does not specify the functions (explained in (b) through (d) below) to be set with the control flags of bits 6 to 4 in byte 1 in particular. According to the default value of each control flag, the IDD executes the format process. When 0 is specified in this bit, the INIT must specify 0 for the control flags in bits 6 to 4 in byte 1.

1: The INIT explicitly specifies the functions (explained in (b) through (d) below) to be set with control flags in bits 6 to 4 of byte 1. The IDD executes the format process according to the specified value for each control flag.

(b) DPRY (disable primary): Default value is 0.

0: Execution of the format process by using the P list is specified. Logical data blocks are not allocated onto the sector positions, where defects are present, registered in the P list. Instead, the alternate blocks are assigned.

1: It is specified that the P list is not used for the defect process. The P list itself is not erased but stored in spite of this setting.

Note:

To implement disk initialization for ordinary operations, the P list must be used. Specify 0 for this bit accordingly.

(c) DCRT (disable certification): Default value is 0.

0: Data block verification after disk initialization is specified. The IDD confirms that all logical data blocks can be read out successfully after initialization. Defective data blocks detected during this verification are registered as the C list. Alternate blocks are assigned for such data blocks.

1: Data block verification after disk initialization is disabled.

(d) STPF (stop format): Default is 1.

This bit specifies whether the command process is to be continued (0) or terminated (1) when the defect list (P list or G list) specified with this command for executing the defect process cannot be read from the disk. For the IDD, setting of this bit is invalid and the specified value is ignored. When the necessary defect list cannot be read, this command terminates with the CHECK CONDITION status. When the P list cannot be read, the sense data indicates "MEDIUM ERROR [=3]/Primary defect list not found [=1C-01]" or "MEDIUM ERROR [=3]/Defect list error in primary list [=19-02]". When the G list cannot be read, the sense data indicates "MEDIUM ERROR [=3]/Grown defect list not found [=1C-02]" or "MEDIUM ERROR [=3]/Defect list error in grown list [=19-03]".

(e) Defect list length

This field specifies the total number of bytes in the defect list to be transferred from the INIT after the header. The byte length of the defect descriptor that constructs the defect list is dependent upon the format. The value to be specified in this field must be a multiple of 4 for the defect descriptor in the block address format, or a multiple of 8 for the defect descriptor in the format of byte distance from the index or in the physical sector address format. When zero is specified in this field, the defect list is not transferred. The maximum byte length of the defect list that can be specified for the IDD is 16 KB (16,384: X'4000').

Note:

The disk defect processing method to be executed upon execution of the FORMAT UNIT command is specified with CDB and the header of the format parameter transferred from the INIT. The INIT may specify the control flag for the formatting process without transfer of the defect list (D list) by setting zero in the "defect list length" field in the header of the format parameter.

b. Defect list (D list)

The defect list (D list) is the defect position information on the disk specified by the INIT and consists of one or more defect descriptors. The defect descriptor must have been described in the format specified in the "defect list format" field of CDB.

Configurations and description formats of the defect descriptor that can be specified with the defect list (D list) are explained below. Only the address information of the user space and CE space on the disk can be specified for the defect list (D list).

(a) Defect descriptor in the block address format

Figure 3.18 shows the description format of this defect descriptor. The defect descriptor in this format specifies the logical block address of the data block containing the disk defect in a 4-byte length. When two or more defect descriptors are specified, they must be specified in ascending order of block addresses.

	Bit							
	7	6	5	4	3	2	1	0
Byte 0	Logical data block address of defective block (MSB)							
1	Logical data block address of defective block							
2	Logical data block address of defective block							
3	Logical data block address of defective block (LSB)							

Figure 3.18 Defect descriptor: block address format

(b) Defect descriptor in the format of byte distance from index

Figure 3.19 shows the description format of this defect descriptor. The defect descriptor in this format specifies the byte distance from the index to the first byte position where the defective bit of the disk is contained, along with the cylinder number and head (track) number. The length of a defect is treated as 8 bytes (64 bits). For the length of a defect that is more than 8 bytes, two or more defect descriptors must be specified. When two or more defect descriptors are specified, they must be specified in ascending order of defect positions with the cylinder number as the the most significant value and the byte distance from the index as the least significant value.

Note:

X'FFFFFFFF' must not be specified as the byte distance between the index and defect position to make an entire track defective.

	Bit							
	7	6	5	4	3	2	1	0
Byte 0	Cylinder number (MSB)							
1	Cylinder number							
2	Cylinder number (LSB)							
3	Head number							
4	Byte distance between index and defect position (MSB)							
5	Byte distance between index and defect position							
6	Byte distance between index and defect position							
7	Byte distance between index and defect position (LSB)							

Figure 3.19 Defect descriptor: format of byte distance from index

(c) Defect descriptor in physical sector address format

Figure 3.20 shows the description format of this defect descriptor. The defect descriptor in this format specifies the physical sector number of the data block containing the disk defect, along with the cylinder number and head (track) number. When two or more defect descriptors are specified, they must be specified in ascending order of defect positions with the cylinder number as the most significant value and the physical sector number as the least significant value.

Note:

The sector number to be described in this format is a physical sector number to which the track skew factor and cylinder skew factor are not applicable. X'FFFFFFFF' must not be specified as the physical sector number of the defective block to make an entire track defective.

	Bit							
	7	6	5	4	3	2	1	0
Byte 0	Cylinder number (MSB)							
1	Cylinder number							
2	Cylinder number (LSB)							
3	Head number							
4	Defective block physical sector number (MSB)							
5	Defective block physical sector number							
6	Defective block physical sector number							
7	Defective block physical sector number (LSB)							

Figure 3.20 Defect descriptor: physical sector address format

Notes: Precautions for specifying D list

The defect position information at the factory shipment is registered in the IDD as the P list. The defect position information as the result of the alternate block assignment after installation is registered as the G list. The function that the D list is specified as the defect position information at the FORMAT UNIT command execution is used for specifying the initial defect position information for the disk medium on which the P list is not registered. Since the known defect position can be specified in the IDD at the installation by specifying the use of the P and G lists, it is necessary to use the D list in the IDD. When the D list is used in the IDD, following precautions must be cave.

1. The maximum number ($p = n + m$) of the defect sectors (logical data blocks) that require the alternate block process (sector slip or alternate sector) are as follows:

   $n + 2m \leqq 2,657$

   "n" is the number of defect sectors to which the sector slip process is applied, and "m" is the number of defect sectors to which the alternate sector process is applied. For details of the alternate block process, refer to Chapter 3 in OEM Manual Specifications & Installation. When the defect process that exceeds this limited number is specified with the FORMAT UNIT command, the command terminates with the CHECK CONDITION status (HARDWARE ERROR [=4]/No defect spare location available [=32-00]).

2. The defect descriptor to be specified as the D list is successfully accepted and registered as the G list when the defect position information that is no more than the physical limit (user space and CE space) of the disk drive has been specified. However, the formatting process is only executed for the user space and CE space specified with the block descriptor of the MODE SELECT parameter, format parameter (page 3), and drive parameter (page 4).

3. When the defect descriptor in the format of byte distance from index is specified in the D list, two sectors may be processed as a defect sector by one defect descriptor or one sector may be processed as a defect sector by two or more defect descriptors depending on the specified defective byte position. Only physical neutral point of the sector processed as a defect sector is registered in the G list. Further, when the specified defective byte position does not affect the read/write operation to the data block, the defect position information is ignored and not treated in the defect sector process nor registered in the G list. Therefore, the defect position information specified with this command does not always match the defect position information read with the READ DEFECT DATA command after this command terminates.

4. When the initialization is performed with changing the number of cylinder or number of logical data block in the user space, the D list of the block address format cannot be used.

5. The logical data block is not always aligned on the same physical sector on the disk medium because of the alternate block assignment, etc. The storage area that does not have a logical address also exists. Therefore, the defect descriptor of the block address format cannot describe all physical defect position information on the disk medium universally.

6. The defect descriptor in the block address format specified for the D list has different meanings of the address to be specified depending on the CmpLst bit value of CDB in this command, so care must be taken.

   - CmpLst = 0: The defect position is treated as it has been specified with the logical block address. When this command is issued, the sector to which the logical data block has been assigned is processed as the defect sector.

   - CmpLst = 1: The defect position is treated as it has been specified with the physical block address. When no defect is present on the disk, the sector to which the data block is assigned is treated as the defect sector.

   In any case, the defect position information to be registered in the G list is the physical block address. Therefore, the defect position information specified with this command with the CmpLst bit set to 0 does not always match the defect position information read with the READ DEFECT DATA command after this command terminates.

(4) Defect process at initialization

Table 3.3 lists combinations of specified values of control flags and the contents of the defect process to be executed by the IDD. For details of the alternate block assignment process, refer to Chapter 3 in OEM Manual Specifications & Installation.

Table 3.3 FORMAT UNIT command defect process (1/2)

CDB byte 1			Header			Defect processing method
FmtData	CmpLst	Defect list format	FOV	DPRY	Defect list length	
0	0 (*4)	0 0 0	(Format parameter not transferred)			① The alternate block is assigned to the defect registered in the P list. ② The existing G list is erased.
1	0 (*1, *2, *3)	d d d	0 1	0 0	Zero	① The alternate block is assigned to the defect registered in the P list and existing G list. ② The existing G list is retained.
1	0 (*1, *2, *3)	d d d	1	1	Zero	① The alternate block is assigned to the defect registered in the existing G list. ② The P list is stored but not used for the defect process. ③ The existing G list is retained.
1	1 (*1, *2)	d d d	0 1	0 0	Zero	① The alternate block is assigned to the defect registered in the P list. ② The existing G list is erased and not used for the defect process.
1	1 (*1, *2)	d d d	1	1	Zero	① Both the P list and existing G list are not used for the defect process. (The alternate block is not assigned.) ② The P list is retained but the existing G list is erased.
1	0 (*1, *3)	d d d	0 1	0 0	> 0	① Alternate blocks are assigned to defects registered in the P list and existing G list, and to defects described in the D list transferred from the INIT. ② The D list is added to the existing G list.

Table 3.3 FORMAT UNIT command defect process (2/2)

CDB byte 1			Header			Defect processing method
FmtData	CmpLst	Defect list format	FOV	DPRY	Defect list length	
1	0	d d d (*1, *3)	1	1	> 0	① Alternate blocks are assigned to defects registered in the existing G list and to defects described in the D list transferred from INIT. ② The P list is retained but not used for the defect process. ③ The D list is added to the existing G list.
1	1	d d d (*1)	0 1	0 0	> 0	① Alternate blocks are assigned to defects registered in the P list and to defects described in the D list transferred from the INIT. ② The existing G list is erased and not used for the defect process. ③ The D list is registered as the new G list.
1	1	d d d (*1)	1	1	> 0	① Alternate blocks are assigned to the defects described in the D list transferred from the INIT. ② The P list is retained but not used for the defect process. ③ The existing G list is erased and not used for the defect process. ④ The D list is registered as the new G list.

*1 ddd: 0,0,0 = D list in the block address format
1,0,0 = D list in the format of byte distance from index
1,0,1 = D list in the physical sector address format

*2 The D list is not transferred from the INIT.

*3 When initialization is performed with changing the data block length, the INIT cannot specify the defect processing method with this combination.

*4 When the defect processing method is specified with this combination, the IDD verifies the data blocks after the initialization and generates the C list. In the defect processing method specified with other combination, the INIT can specify the inhibition or permission of the verification by the DCRT flag of the format parameter explicitly.

3.3.2 REASSIGN BLOCKS

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'07'							
	1	LUN			0	0	0	0	0
	2	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	Flag	Link

This command assigns the alternate data block to the defect data block specified in the defect data list transferred from the INIT. For the alternate block assignment method to be implemented with this command, refer to Chapter 3 in OEM Manual Specifications & Installation.

The INIT specifies the logical block addresses of one or more defect data blocks in the defect data list transferred to the IDD with this command. The IDD finds out the unused alternate block spare sector and assigns the alternate block to the specified logical data block. When a data block to which the alternate block has been assigned is specified, the IDD reassigns another usable alternate block spare sector to the data block.

The IDD tries to duplicate the contents of the logical data blocks specified by the defect data list into the data field of the assigned alternate block. When the data field of the logical data blocks specified by the defect data list can be corrected by the ECC, the corrected data is duplicated. When it cannot be corrected, uncorrected data (including error) is duplicated as it is. When other medium error occurs, X'00' is duplicated in all bytes. The contents of the logical data blocks other than blocks specified by the defect data list are not affected by the alternate assignment of this command.

Note:

The IDD tries to duplicate the contents of the data field in the logical data blocks specified by the defect data list into the alternate blocks assigned by this command, but sometimes fails. The INIT should confirm the contents of the assigned alternate blocks, saves data before issuing this command, and restores data after execution of this command.

Figure 3.21 shows the format of the defect data list to be transferred from the INIT by this command.

		Bit							
		7	6	5	4	3	2	1	0
Header	Byte 0	X'00'							
	1	X'00'							
	2	Defect list length (m) (MSB)							
	3	Defect list length (m) (LSB)							
Defect descriptor list	4	Logical block address of defect block (MSB)							
	5	Logical block address of defect block							
	6	Logical block address of defect block							
	7	Logical block address of defect block (LSB)							
	8 ≈ m+3	Logical block address of defect block							

Figure 3.21 REASSIGN BLOCK command: defect data list configuration

The defect data list consists of the 4-byte header and one or more following defect descriptors. One defect descriptor is 4 bytes long.

The "defect list length" field in the header indicates the total number of bytes (m) of the defect descriptor transferred after the header and it must be a multiple of 4. When zero is specified in this field, the defect descriptor list transfer and the alternate block assignment process are not performed and the command terminates.

Note:

The defect list length that can be specified for the IDD is 2044 (X'7FC') bytes or less. Therefore, a maximum of 511 blocks can be specified with one REASSIGN BLOCKS command.

The 4-byte logical block address of the defect block is described in the defect descriptor. When two or more defect descriptors are specified, the INIT had better to describe the defect descriptors in ascending order of logical data block addresses.

When a logical data block address is specified in a duplicated manner in the defect descriptor list, this command terminates with the CHECK CONDITION status (ILLEGAL REQUEST [=5]/Invalid field in parameter list [=26-00]) and the command does not execute the alternate block assignment process at all.

The IDD assigns alternate blocks to the specified data blocks from the leading of the defect descriptor list. When all of the usable spare sectors are used up and alternate block assignment becomes impossible, this command terminates at that point and the CHECK CONDITION status is reported. In this case, the sense data shows:

- Sense key: 4 = HARDWARE ERROR
- Sense code/subsense code: 32-00 = No defect spare location available
- "VALID" bit: 1
- Information field/Command-inherent information field:
  Logical block address specified for the defect descriptor at the point alternate block assignment is disabled

When this command abnormally terminates with the CHECK CONDITION status due to any error other than the above, the "command-inherent information" field in the sense data reports the logical block address that has been specified for the first defect descriptor for which the alternate block was not assigned. When the defect descriptor for which the alternate block was not assigned cannot be identified or when the alternate block assignment process for all specified defect descriptors has terminated, the "command-inherent information" field indicates X'FFFFFFFF'.

When the alternate block assignment process is successful, the defect position information on the disk related to the data block specified in the defect descriptor list is recorded on the disk as the defect list (G list). INIT can read the contents of the G list by using the READ DEFECT DATA command. This command does not affect the contents of the primary defect list (P list).

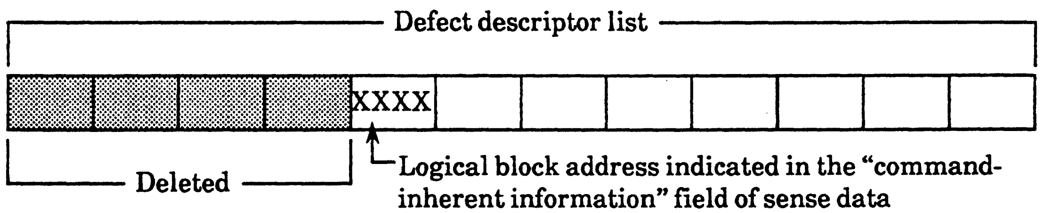
Note:

The defect position information to be recorded in the G list is a physical block address (address of the logical data block assigned when there is no defect on the disk). Therefore, the value (logical block address) specified in the defect descriptor list of this command does not always match the contents (physical block address) of the G list read with the READ DEFECT DATA command after this command terminates. For details of the READ DEFECT DATA command, see Subsection 3.3.3.

Note: Precautions for use of command

When this command terminates with the CHECK CONDITION status, the sense code/subsense code in the sense data is other than "No defect spare location available [=32-00]", and the "command-inherent information" field indicates a valid logical block address (other than X'FFFFFFFF'), the INIT should execute the recovery process (see Section 4.3) according to the contents of sense data and then reissue this command as explained below.

1. From the defect descriptor list specified with this command, defect descriptors before the defect descriptor that specifies the logical block address displayed in the command-inherent information field of sense data are deleted.



2. The defect list length of the header is revised, the new defect descriptor list updated in 1) is added, and the REASSIGN BLOCKS command is issued again.

3.3.3 READ DEFECT DATA

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'37'							
	1	LUN			0	0	0	0	0
	2	0	0	0	PList	GList	Defect List Format		
	3	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0	0
	7	Transfer byte-length (MSB)							
	8	Transfer byte-length (LSB)							
	9	0	0	0	0	0	0	Flag	Link

This command transfers the list (defect data) that describes the defect position information of the disk to INIT.

There are two types of defect data; P list (primary defect list) and G list (growth defect list). The P list contains the defect position information at disk drive shipment from the factory. The G list contains the position information of the defect data block for which the alternate block has been assigned by the REASSIGN BLOCKS command, the automatic alternate block assignment process, or with the defect position information specified from the INIT upon execution of the FORMAT UNIT command or with verification after initialization.

The INIT can specify the type of defect data to be transferred to the INIT with the PList (primary list) bit and GList (growth list) bit on CDB, and the format of defect data with the "defect list format" field.

PList	GList	Defect data type
1	1	P list and G list
1	0	P list only
0	1	G list only
0	0	4-byte header information only (explained later in this section)

Defect list format			Defect data format
0	0	0	Block address format
1	0	0	Format of byte distance from index
1	0	1	Physical sector address format

The "transfer byte-length" field of CDB specifies the length (bytes) of defect data that can be received by the INIT. When the IDD completes transfer of defect data corresponding to the length specified in the "transfer byte-length" field or completes transfer of all defect data of the specified type, the IDD terminates data transfer. When zero is specified in the "transfer byte-length" field, data transfer is not executed and this command terminates.

Figure 3.22 shows the format of defect data to be transferred to the INIT by this command.

		Bit							
		7	6	5	4	3	2	1	0
Header	Byte 0	X'00'							
	1	0	0	0	PList	GList	Defect List Format		
	2	Defect list length (MSB)							
	3	Defect list length (LSB)							
Defect descriptor list	4	Defect data							
	≈								
	n								

Figure 3.22 READ DEFECT DATA command: defect data list configuration

(1) Header

a. PList (primary list) bit

When this bit is 1, the defect descriptor list to be actually transferred to the INIT contains the defect data in the P list. When this bit is 0, the defect data in the P list is not contained. (See Note 3)

b. GList (growth list) bit

When this bit is 1, the defect descriptor list to be actually transferred to the INIT contains the defect data in the G list. When this bit is 0, the defect data in the G list is not contained.

c. Defect list format

This field indicates the description format of the defect descriptor list to be actually transferred to the INIT. The IDD can transfer defect data in three formats that can be specified with CDB. This field shows the same values as that specified in the "defect list format" field of CDB.

d. Defect list length

This field indicates the total number of bytes of the defect descriptor list that can be transferred next to the 4-byte header. The total number of bytes indicated is a multiple of 4 or 8. The value indicated in this field is the total number of bytes provided when description is made with the "defect list format" with certain defect data (P list and G list) specified, regardless of the value specified in the "transfer byte count" field of CDB. The INIT should check that the "value indicated in this field + 4" is equal to or less than the value specified in the "transfer byte count" field of CDB, to confirm that all defect data requested with this command has been transferred. The INIT can know the number of defects on the disk from the value (quotient) obtained by dividing the value in the field by the number of bytes (4 or 8) per defect descriptor.

(2) Defect descriptor list

Data to be transferred after the 4-byte header is the defect descriptor list (defect data) that describes the defect position information of the type and format specified with CDB. A defect descriptor is 4 bytes in the block address format or 8 bytes in the format of byte distance from index or in the physical sector address format. Defect descriptors are not always transferred in ascending order of defect position information.

For configuration and contents of the defect descriptor in each format, see Subsection 3.3.1.

Notes: Precautions for use of command

1. When 0 is specified for both the PList bit and GList bit of CDB and transfer of both the P list and G list is requested, the IDD transfers all P lists first then transfers G lists. (Defect information of both lists is not merged.)

2. When 0 is specified for both the PList and GList bits of CDB, the command transfers the header only. The following information is indicated in the header.

   - PList bit: 0
   - GList bit: 0
   - "Defect list length" field: Total number of bytes provided when description is made with "defect list format" for which defect data contained in the P list and G list has been specified

3. If there is no defect data (the defect data list is empty) in the defect list specified by CDB (P list or G list), 1 is indicated in the PList bit and the GList bit of the header transferring to INIT, according to the CDB specification.

4. The INIT can know the length (number) of defect data contained in the P list and the G list by issuing this command with 4 specified in the "transfer byte-length" field of CDB and checking the header information transferred from the IDD.

5. Defect data to be transferred is conditioned as explained below depending on the combination of the type and format of defect data.

defect list format	PList	GList
Block address format	②	②
Format of byte distance from index	①	①, ③
Physical sector address format	①	①

① Regardless of the size of the user space, all defect position information on disk medium other than system space are reported. The defect position information for the portion that cannot be explicitly accessed from the INIT such as the alternate block spare sector is also reported.

② When there is no defect on the disk, the address of the logical data block to be designed to the block (=physical block address) is reported. The defect position information for the portion that cannot be explicitly accessed from the INIT (the portion which does not have the address of the logical data block) is not reported.

6. The number of defects to be reported with this command is dependent upon the format of defect data.

   - In the block address format, the defect position information for the portion that cannot be explicitly accessed from the INIT is not reported.

   - In the block address format or the physical sector address format, even if defects exist in the sector, the defect position information is reported with one defect descriptor.

   - When the P list is reported in the format of byte distance from index, all registered defect position information are reported. However in the block address format or the physical sector address format, the defect position information for the defect that does not affect the read/write operation of the data block is not reported.

   - When the P list is reported in the format of byte distance from index, a defect sector may be reported with two or more defect position information and two defect sectors may be reported with one defect position information because a defect crossed over the boundary between sectors.

7. Defect data of the block address format cannot describe all physical defect position information on the disk media universally. For example, the defect position information for the portion that does not have block address (cylinder that cannot be used as spare sector, user space or CE space) cannot be described. The defect data of this format is equipped to keep the continuance with the conventional models, but it is recommended for INIT to avoid adopting the block address format.

3.4 Maintenance and Diagnostic Commands

3.4.1 SEND DIAGNOSTIC

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'1D'							
	1	LUN			PF	0	SelfTest	DevOfl	UnitOfl
	2	0	0	0	0	0	0	0	0
	3	Parameter list length (MSB)							
	4	Parameter list length (LSB)							
	5	0	0	0	0	0	0	Flag	Link

This command executes the self-diagnostic test provided in the IDD or the operation specified with the parameter list transferred from the INIT.

(1) Self-diagnostic test

When the SelfTest (Self Test) bit of CDB is 1, this command specifies execution of the self-diagnostic test provided in the IDD. In this case, the PF (page format) bit and parameter list length field of CDB are meaningless and the specified values are ignored. The DevOfl (device offline) bit specifies whether the operation in the self-diagnostic test that may affect the states of logical units other than that specified with this command is enabled. There is only one logical unit in the IDD, so setting of this bit is meaningless and the specified value is ignored.

The INIT can specify the type of self-diagnostic test to be executed with the UnitOfl (unit offline) bit of CDB. When 1 is specified for the SelfTest bit, the IDD executes a series of self-diagnostic tests according to the combination with the UnitOfl bit as follows:

Type of self-diagnostic test	UnitOfl = "0"	UnitOfl = "1"
① Controller function test	×	×
② Disk drive seek test		×
③ Disk (CE space) write/read/data compare test		×

×: executed

The IDD reports the GOOD status when all of the specified self-diagnostic tests terminate successfully. When an error is detected in any specified self-diagnostic test, the CHECK CONDITION status is reported and the sense data indicates the information related to the detected error. For instance, when an error is detected in the controller function test, the sense key indicates "HARDWARE ERROR [=4]" and the sense code/subsense code indicates "Diagnostic failure on component 'nn' [=40-nn]". (nn is a code in the range of X'80' to X'FF' and indicates the type of the error. This value is an unique code for Fujitsu, and the meaning is not announced. The user should notify this code to the field engineer as a information for the repair.)

Upon execution of the diagnostic tests (seek test and write/read/data compare test) related to the disk drive, the error recovery process complies with the mode that has been set with the parameter (page code 1: read/write error recovery parameter, page code 21: additional error recovery parameter) of the MODE SELECT command except in the following case:

Particular case of MODE SELECT parameter in self-diagnostic test:

- AWRE, ARRE, and TB flags are not applicable.
- For the PER and DTE flags, see Table 3.4.

Table 3.4 Error recovery control flags in self-diagnostic test

PER	DTE	Diagnostic test operation
0	0	The diagnostic test is continued when error recovery is successful. The contents of the recovered error are not reported. When an unrecoverable error is detected, the diagnostic test terminates with an error at that point.
0	1	----- Setting impossible ------
1	0	The diagnostic test is continued when error recovery is successful. When an unrecoverable error is detected, the diagnostic test terminates with an error at that point. Even if recovery of all of the detected errors is successful, the CHECK CONDITION status (RECOVERED ERROR [=1]) is reported after a series of diagnostic tests terminate and the sense data indicates the contents of the last recovered error.
1	1	The diagnostic test terminates with an error when the enable error recovery process terminates whether error recovery is successful or impossible, and the CHECK CONDITION status is reported. Sense data indicates the contents of the detected error.

Note:

When 1 is specified in the SelfTest bit with this command, the command execution result is reported with the status byte and sense data. Therefore, the self-diagnostic test execution result is not reported as response data even though the RECEIVE DIAGNOSTIC RESULTS command is executed after that command.

(2) Parameter setting

When 0 is specified in the SelfTest bit of CDB, the IDD executes the operation specified with the parameter list transferred from the INIT by this command. In this case, the IDD reports the GOOD status when the specified operation terminates and the response data is prepared, and then terminates this command. The INIT can read the execution result (response data) with the RECEIVE DIAGNOSTIC RESULTS command.

When the PF (page format) bit of CDB is 1, the parameter list to be transferred from the INIT by this command is in the page format explained later. The IDD ignores the specified value of this bit. When the parameter list is transferred by this command, the IDD always assumes that the parameter list is in the page format. When the SelfTest bit is 0, settings of the DevOfl and UnitOfl bits are meaningless and the specified values are ignored.

The "parameter list length" field of CDB indicates the length (bytes) of the parameter list transferred from the INIT when the SelfTest bit is 0. When zero is specified in the "parameter list length" field, no operation is executed and this command terminates. When all bytes of the parameter list cannot be received since the specified value in the "parameter list length" field is less than the specified length of the parameter list explained below, the command terminates with the CHECK CONDITION status (ILLEGAL REQUEST [=5] / Invalid field in CDB [=24-00]).

Figure 3.23 shows the format of the parameter list (called the parameter page) transferred from the INIT to the IDD by this command. The parameter page consists of the 4-byte page header and the page parameter that follows. The INIT can specify only a single parameter page with this command. Even if the INIT specifies two or more parameter pages, the IDD executes only the operation specified by the first parameter page.

Note: Precautions for use of command

1. For this command that has specified 0 in the SelfTest bit, the INIT can specify only a single parameter page.

2. To avoid that the execution result (response data) of this command is not ensured with a command issued by another INIT, the INIT should execute the RECEIVE DIAGNOSTIC RESULTS command linked to this command when 0 is specified in the SelfTest bit. Otherwise, the INIT should issue this command after the IDD is reserved, and reset the reservation after the RECEIVE DIAGNOSTIC RESULTS command is executed.

3. When a command other than the RECEIVE DIAGNOSTIC RESULTS command is linked to this command, the command execution result (response data) may not be ensured.

	Byte	Bit 7	6	5	4	3	2	1	0
Header	0	Page code							
	1	0	0	0	0	0	0	0	0
	2	Page parameter length (MSB)							
	3	Page parameter length (LSB)							
Page parameter	4	Parameter							
	5								
	≈								
	n								

Figure 3.23 SEND DIAGNOSTIC command: parameter list configuration

- Page code

  This field specifies the code that identifies the type of the parameter page to be transferred from the INIT and the operation to be executed at the IDD. Parameter lists and functions that can be specified by the INIT are as follows:

Page code (Hex)	Function
00	Reporting the list of the supported page code
40	Logical/physical address conversion
80	Reporting the mode setting (setting terminal) state
81	Reporting the drive-inherent information

- Page parameter length

  This field specifies the byte-length of the page parameter that follows byte 4. The INIT must specify the same value as the specified length of each parameter page explained below.

- Page parameter

  This field specifies the parameter inherent to each page code. This field may not be required depending on the page code (page parameter length = zero).

a. Page code list

   This parameter page specifies that the list of page codes of the parameter page supported by the IDD using the SEND DIAGNOSTIC or RECEIVE DIAGNOSTIC RESULTS command is transferred to the INIT. Figure 3.24 shows the format of this parameter page. The list of page codes supported by the IDD is transferred to the INIT by the RECEIVE DIAGNOSTIC RESULTS command that is issued after the SEND DIAGNOSTIC command that has specified this parameter page (see item (1) in Subsection 3.4.2).

	Bit							
	7	6	5	4	3	2	1	0
Byte 0	X'00' (page code)							
1	0	0	0	0	0	0	0	0
2	X'00' (page parameter length)							
3	X'00' (page parameter length)							

Figure 3.24 SEND DIAGNOSTIC parameter: page code list

b. Logical/physical address conversion

   This parameter page specifies that the address information described in the format of the logical block address, physical sector address, or byte distance from index is converted to another description format. Figure 3.25 shows the format of this parameter page. The INIT specifies the address information to be converted with this parameter page of the SEND DIAGNOSTIC command and can read the conversion result with the RECEIVE DIAGNOSTIC RESULTS command (see item (2) in Subsection 3.4.2).

	Bit 7	6	5	4	3	2	1	0
Byte 0	X'40' (page code)							
1	0	0	0	0	0	0	0	0
2	X'00' (page parameter length)							
3	X'0A' (page parameter length)							
4	0	0	0	0	0	Address format before conversion		
5	0	0	0	0	0	Address format after conversion		
6	Logical or physical address							
7								
≈								
13								

Figure 3.25 SEND DIAGNOSTIC parameter: logical/physical address conversion

The "address format before conversion" field in the parameter page indicates the format of the address information specified in bytes 6 to 13. The IDD converts the address information to the description format specified in the "address format after conversion" field. As the address format, one of the following codes can be specified.

Code	Address format
0 0 0	Logical block address format
1 0 0	Format of byte distance from index
1 0 1	Physical sector address format

The description format of the address information specified in bytes 6 to 13 is the same as the description rule in the D list to be transferred from the INIT by the FORMAT UNIT command. For details, refer to the explanation of the FORMAT UNIT command (Subsection 3.3.1). When the logical block address format is specified, it must be described in bytes 6 to 9 and zero must be specified in the other byte positions.

When the address of a nonexistent logical data block (out of range of the MODE SELECT parameter) is specified in the logical block address format or when an unallocatable area as the user space or CE space (physically nonexistent cylinder) in the disk drive in the format of byte distance from index or in the physical sector address format, the command terminates with the CHECK CONDITION status (ILLEGAL REQUEST [=5] / Invalid field in parameter list [=26-00]) and address conversion is not executed.

Details of the algorithm of address conversion to be executed when this parameter is specified and the data format of the conversion result to be reported to the INIT are explained in explanation of the RECEIVE DIAGNOSTIC RESULTS command (Subsection 3.4.2).

c. Mode setting

This parameter page specifies that setting states of various operation modes specified with setting terminal on the IDD are reported to the INIT with the RECEIVE DIAGNOSTIC RESULTS command to be issued after this command (see item (3) in Subsection 3.4.2). Figure 3.26 shows the format of this parameter page.

[Fujitsu unique parameter]

	Bit							
	7	6	5	4	3	2	1	0
Byte 0	X'80' (page code)							
1	0	0	0	0	0	0	0	0
2	X'00' (page parameter length)							
3	X'00' (page parameter length)							

Figure 3.26 SEND DIAGNOSTIC parameter: mode setting

d. Drive-inherent information

This parameter page specifies that the drive-inherent information, such as the device serial number of the IDD and the microcode revision, is reported to the INIT with the RECEIVE DIAGNOSTIC RESULTS command to be issued after this command (see item (4) in Subsection 3.4.2). Figure 3.27 shows the format of this parameter page.

[Fujitsu unique parameter]

	Bit 7	6	5	4	3	2	1	0
Byte 0	X'81' (page code)							
1	0	0	0	0	0	0	0	0
2	X'00' (page parameter length)							
3	X'00' (page parameter length)							

Figure 3.27 SEND DIAGNOSTIC parameter: drive-inherent information

3.4.2 RECEIVE DIAGNOSTIC RESULTS

Byte	Bit 7	6	5	4	3	2	1	0
0	X'1C'							
1	LUN			0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	Transfer byte-length (MSB)							
4	Transfer byte-length (LSB)							
5	0	0	0	0	0	0	Flag	Link

This command transfers the data (response data) indicating the execution result of the SEND DIAGNOSTIC command from IDD to INIT. The type and contents of the response data are determined by the parameter list (page code) specified by the INIT using the SEND DIAGNOSTIC command.

The "transfer byte-length" field of CDB indicates the maximum number of bytes of the response data that can be received by the INIT with this command. The IDD transfers the number of bytes that is specified in this field or all bytes of effective response data, whichever is smaller, to the INIT. When zero is specified in this field, this command transfers no data and terminates.

Notes: Precautions for use of command

1. To avoid that the execution result (response data) of the SEND DIAGNOSTIC command is not ensured by commands issued by another INIT before this command is issued, this command should be linked with the SEND DIAGNOSTIC command and issued or the SEND DIAGNOSTIC command and this command should be executed after the IDD is reserved.

2. The response data is valid when 0 is specified in the SelfTest bit and after the SEND DIAGNOSTIC command with a particular operation specified with the parameter list is executed. This command transfers the response data indicating the result of the SEND DIAGNOSTIC command executed last by the IDD to the INIT. The response data is not cleared when this command is executed but effectively retained until next SEND DIAGNOSTIC command is executed, the RESET condition occurs, or the BUS DEVICE RESET message is issued from any INIT.

3. When this command is issued while effective response data is not present, the IDD transfers maximum 4-byte X'00' data to the INIT.

Figure 3.28 shows the format of the response data to be transferred from the IDD to the INIT by this command. The response data consists of the 4-byte page header and succeeding page parameter.

	Byte	Bit 7	6	5	4	3	2	1	0
Header	0	Page code							
	1	0	0	0	0	0	0	0	0
	2	Page parameter length (MSB)							
	3	Page parameter length (LSB)							
Page parameter	4	Parameter							
	5								
	≈								
	n								

Figure 3.28 RECEIVE DIAGNOSTIC RESULTS command: response data configuration

- **Page code**

  This field is the same value as the page code specified with the parameter list transferred from the INIT by last executed SEND DIAGNOSTIC command, and indicates the code that identifies the type of response data to be reported with this command.

- **Page parameter length**

  This field indicates the byte length of the page parameter that follows byte 4.

- **Page parameter**

  Data indicating the execution result of the specified operation of the SEND DIAGNOSTIC command is reported to this field.

The type and contents of the response data to be transferred to the INIT by this command are explained below.

(1) Page code list

This response data reports the list of page code of the parameter page supported by the IDD with the SEND DIAGNOSTIC and RECEIVE DIAGNOSTIC RESULTS commands to byte 4 and succeeding bytes. Figure 3.29 shows the type and contents of this response data.

	Bit							
	7	6	5	4	3	2	1	0
Byte 0	X'00' (page code)							
1	0	0	0	0	0	0	0	0
2	X'00' (page parameter length)							
3	X'04' (page parameter length)							
4	X'00' (page code list)							
5	X'40' (Logical/physical address conversion)							
6	X'80' (Mode setting)							
7	X'81' (Drive-inherent information)							

Figure 3.29 RECEIVE DIAGNOSTIC RESULTS response data: page code list

(2) Logical/physical address conversion

This response data reports the execution result of address conversion specified with the "logical/physical address conversion" parameter of the SEND DIAGNOSTIC command to byte 4 and succeeding bytes. Figure 3.30 shows the type and contents of this response data.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	X'40' (page code)							
1	0	0	0	0	0	0	0	0
2	X'00' (page parameter length)							
3	X'0A' (page parameter length)							
4	0	0	0	0	0	Address format before conversion		
5	RAREA	ALTSEC	0	0	0	Address format after conversion		
6	Logical or physical address							
7								
≈								
13								

Figure 3.30 RECEIVE DIAGNOSTIC RESULTS response data: logical/physical address conversion

Note:

The "page parameter length" field value of this response data is variable in the range of (2 + 8n) on SCSI specifications. For instance, when two or more logical data blocks are placed in one physical sector or one logical data block is placed in two or more physical sectors, "n" pieces of address information are reported as the address conversion result.

According to the present IDD specifications, two or more address information are not reported with this response data and the page parameter length always indicates X'000A'. For future expansion of specifications, the INIT should be able to accommodate the variable"page parameter length".

The "address format before conversion" field in byte 4 and "address format after conversion" field in byte 5 are the same values as the code indicating the description format of address information specified with the parameter of the SEND DIAGNOSTIC command. The "address format after conversion" field indicates the description format of address information to be reported to bytes 6 to 13 of this response data. Codes of the address format are as follows:

Code	Address format
0 0 0	Logical block address format
1 0 0	Format of byte distance from index
1 0 1	Physical sector address format

The description format of address information indicated in bytes 6 to 13 is equal to the description rule in the D list to be transferred from the INIT with the FORMAT UNIT command. For details, see the explanation of the FORMAT UNIT command, (Subsection 3.3.1). In the logical block address format, the address is indicated in bytes 6 to 9 and zero is reported to the other byte positions. When address information specified by the SEND DIAGNOSTIC command indicates an unused position on the disk (RAREA = 1) as the logical data block, X'FFFFFFFF 00000000' is reported as the logical block address after conversion.

The RAREA (reserved area) and ALTSCT (alternate sector) bits indicate the area and its state on the disk where the data block indicated with address information of bytes 6 to 13 is present, according to definitions in Table 3.5. For area classification, spare sector position, and details of the alternate block process, refer to Chapter 3 of OEM Manual Specifications & Installation.

Table 3.5 Logical/physical address conversion: data block state indication

RAREA	ALTSCT	Data block state
0	0	Data block is present in the primary area and is in either block explained below. • Normal block • Block slipped to the position other than the spare sector (within primary area)
0	1	Data block is present on the assigned spare sector and is in either block below. • Block slipped to the spare sector • Block for which the alternate sector has been assigned to the spare sector on the same cylinder or alternate cylinder
1	0	Data block is present in the primary area, but is a defect block for which an alternate block has been assigned and is unused as a logical data block. Otherwise, a logical data block has not been assigned to this data block (i.e. outside the range of the MODE SELECT parameter).
1	1	The data block is an unused (unassigned or defect) spare sector.

Note:

Primary area is the area other than the spare sector and the alternate cylinder that configure the user space or the CE space on a cylinder.

Algorithm of address conversion to be executed with the SEND DIAGNOSTIC command and address information after conversion to be reported to the INIT with this command are as follows:

- Logical block address ⇒ Logical block address

  The logical block address specified with the SEND DIAGNOSTIC command is reported as it is. The area on the disk to which the logical data block has been assigned is indicated in the ALTSCT bit.

- Logical block address ⇒ Byte distance from index

  The starting position of the physical sector to which the logical data block specified with the SEND DIAGNOSTIC command has been assigned is reported. The area on the disk where the logical data block is present is indicated in the ALTSCT bit.

- Logical block address ⇒ Physical sector address

  The physical sector address to which the logical data block specified with the SEND DIAGNOSTIC command has been assigned is reported. The area on the disk where the logical data block is present is indicated in the ALTSCT bit.

- Byte distance from index ⇒ Logical block address

  The logical block address of the data block which has been assigned to the physical sector containing the byte position specified with the SEND DIAGNOSTIC command is reported. The area and its state on the disk where the data block is present are indicated in the RAREA and ALTSCT bits. (*1)

- Byte distance from index ⇒ Byte distance from index

  The starting position of the physical sector containing the byte position specified with the SEND DIAGNOSTIC command is reported. The area and its state on the disk where the sector is present are indicated in the RAREA and ALTSCT bits. (*1)

- Byte distance from index ⇒ Physical sector address

  The physical sector address containing the byte position specified with the SEND DIAGNOSTIC command is reported. The area and its state on the disk where the sector is present are indicated in the RAREA and ALTSCT bits. (*1)

- Physical sector address ⇒ Logical block address

  The logical block address of the data block assigned to the physical sector specified with the SEND DIAGNOSTIC command is reported. The area and its state on the disk where the sector is present are indicated in the RAREA and ALTSCT bits.

- Physical sector address ⇒ Byte distance from index

  The starting position of the physical sector specified with the SEND DIAGNOSTIC command is reported. The area and its state on the disk where the sector is present are indicated in the RAREA and ALTSCT bits.

- Physical sector address → Physical sector address

  The physical sector address specified with the SEND DIAGNOSTIC command is reported as it is. The area and its state on the disk where the sector is present are indicated in the RAREA and ALTSCT bits.

*1 When the position at the rear of the track that is not used as the data block (sector) is specified in the format of byte distance from index, the address conversion is executed assuming that the sector physically positioned last in the track has been specified.

(3) Mode setting

This response data reports the values actually set in the setting terminal on the IDD to byte 4 and succeeding bytes. Figure 3.31 shows the format and contents of this response data. For details of each setting terminal, refer to Chapter 5 of OEM Manual Specifications & Installation.

	Bit							
	7	6	5	4	3	2	1	0
Byte 0	X'80' (page code)							
1	0	0	0	0	0	0	0	0
2–3	X'0008' (page parameter length)							
4	CNH1				CNH7			
	13 – 14	15 – 16	17 – 18	19 – 20	7 – 8	5 – 6	3 – 4	1 – 2
	SCSI bus parity	Synchro-nous transfer request	LED indica-tion	Motor start	Write protect	SCSI ID		
						2^2	2^1	2^0
5	CNH2		CNH1					
	3 – 4	1 – 2	11 – 12	9 – 10	7 – 8	5 – 6	3 – 4	1 – 2
	"0"	"0"	"1"	Reselect retry number	Unit attention report	Offline diagnos-tics	SCSI level	PER default
6	X'00' (Reserved)							
7	X'00' (Reserved)							
8	X'00' (Reserved)							
9	X'00' (Reserved)							
10	X'00' (Reserved)							
11	X'00' (Reserved)							

Figure 3.31 RECEIVE DIAGNOSTIC RESULTS response data: mode setting

Note:

The reported value in bytes 4 to 5 of this response data corresponds to the state of the setting terminal as follows.

0: Open state between pins
1: Short state between pins

(4) Drive-inherent information

This response data reports the drive-inherent information, such as the product ID, device serial number, and microcode revision, to byte 4 and succeeding bytes. Figure 3.32 shows the format and contents of this response data.

	Bit							
	7	6	5	4	3	2	1	0
Byte 0	X'81' (page code)							
1	0	0	1	1	1	0	1	1
2–3	X'002C' (page parameter length)							
4 5 ≈ 19	M262※S–□□□□ (Product ID: ASCII) [※: 2, 3, 4 □: Logical block length]							
20 21 ≈ 31	Device serial number (ASCII)							
32 33 34 35	Microcode revision (ASCII)							
36 37 ≈ 47	X'2020 · · · 20' (Reserved: ASCII)							

Figure 3.32 RECEIVE DIAGNOSTIC RESULTS response data: drive-inherent information

a. Product ID

This field contains a left-justified ASCII code representing the product model name. This value is the same as the value in the "product ID" field of the standard INQUIRY data reported by the INQUIRY command. For details, refer to the explanation of INQUIRY command (Subsection 3.1.2).

b. Device serial number

This field contains a right-justified decimal integer (ASCII code) that indicates the device serial number of the IDD. This value is the same as the value in the "device serial number" field of the VPD information reported by the INQUIRY command. For details, refer to the explanation of INQUIRY command (Subsection 3.1.2).

c. Microcode revision

This field indicates the microcode revision of the IDD with the ASCII code. This value is the same as the value in the "product revision" field of the standard INQUIRY data reported by the INQUIRY command. For details, refer to the explanation of INQUIRY command (Subsection 3.1.2).

3.4.3 WRITE BUFFER

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'3B'							
	1	LUN			0	0	Mode		
	2	Buffer ID							
	3	Buffer address (MSB)							
	4	Buffer address							
	5	Buffer address (LSB)							
	6	Transfer byte-length (MSB)							
	7	Transfer byte-length							
	8	Transfer byte-length (LSB)							
	9	0	0	0	0	0	0	Flag	Link

This command is used to diagnose the validity of the IDD data buffer memory and SCSI bus in combination with the READ BUFFER command. This command does not change data on the disk.

According to the setting of CDB in this command, the IDD stores data transferred from the INIT into the data buffer on the IDD. The IDD has four segmented 60-KB (61,440 bytes) data buffers. This command can specify the data storage position in units of bytes by using the buffer address in the range of X'000000' to X'00EFFF'. Zero, One, Two or Three must always be specified in the "Buffer ID" field of CDB. The INIT can know the IDD buffer configuration and the address-settable unit by using the READ BUFFER command.

The function of this command and the format of the data to be transferred from the INIT are specified in the "Mode" field of CDB byte 1. Either transfer mode explained below can be selected.

"Mode" Bit 2	1	0	Transfer mode
0	0	0	Header + data, address not specified
0	0	1	Header + data, address specified
0	1	0	Data only, address specified

(1) Mode = 0,0,0: Header + data, address not specified

In this mode, the 4-byte header (all zeroes specified) must precede the data to be transferred from the INIT. Zero must be specified in the "buffer address" fields of CDB.

The "transfer byte-length" field of CDB specifies the total number of bytes to be transferred from the INIT. This setting includes 4 bytes of the header. The IDD stores the data corresponding to the length of data transferred from the INIT, excluding the header [transfer byte-length − 4 bytes], to the starting address X'000000' of the data buffer.

For the "transfer byte-length" field of CDB, a value not greater than [IDD buffer size + 4 bytes] must be specified. When a larger value is specified, data transfer from the INIT is not executed. When zero is specified in the "transfer byte-length" field, data transfer is not executed and this command terminates.

Figure 3.33 shows the format of data to be transferred from the INIT when this mode is specified.

		Bit							
		7	6	5	4	3	2	1	0
Header	Byte 0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0
Data	4	Buffer data (byte 0)							
	5	Buffer data (byte 1)							
	≈	≈							
	n	Buffer data (byte n-4)							

Figure 3.33 WRITE BUFFER command: buffer data (Mode = 000, 001)

(2) Mode = 0,0,1: Header + data, address specified

In this mode, the format of data to be transferred from the INIT is the same as that in Mode = 0,0,0. A 4-byte header (all zeroes specified) must precede the data to be transferred.

In this mode, the starting address of the data buffer to store data transferred from the INIT can be specified in the "buffer address" field of CDB.

The "transfer byte-length" field of CDB specifies the total number of bytes of data to be transferred from the INIT. This setting includes 4 bytes of the header. The IDD stores data corresponding to the length of [transfer byte-length − 4 bytes]; bytes of data subtracting data transferred by the INIT from data of the header, starting from the byte position on the data buffer specified in the "buffer address" field of CDB.

A value less than [IDD buffer size − buffer address field specified value + 4 bytes] must be specified in the "transfer byte-length" field of CDB. When a larger value is specified, data transfer from the INIT is not executed. When zero is specified in the "transfer byte-length" field, data transfer is not executed and this command terminates.

(3) Mode = 0,1,0: Data only, address specified

Data to be transferred from the INIT in this mode is the buffer data only and the 4-byte header is not added.

The starting address of the data buffer to store data transferred from the INIT can be specified in the "buffer address" field of CDB.

The "transfer byte-length" field of CDB specifies the total number of bytes of buffer data to be transferred from the INIT. The IDD stores data transferred from the INIT starting from the byte position on the data buffer specified in the "buffer address" field of CDB.

A value less than [IDD buffer size − buffer address field specified value] must be specified in the "transfer byte-length" field of CDB. When a larger value is specified, data transfer from the INIT is not executed. Also when zero is specified in the "transfer byte-length" field, data transfer is not executed and this command terminates.

3.4.4 READ BUFFER

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'3C'							
	1	LUN			0	0	Mode		
	2	Buffer ID							
	3	Buffer address (MSB)							
	4	Buffer address							
	5	Buffer address (LSB)							
	6	Transfer byte-length (MSB)							
	7	Transfer byte-length							
	8	Transfer byte-length (LSB)							
	9	0	0	0	0	0	0	Flag	Link

This command is used to diagnose the validity of the IDD data buffer memory and SCSI bus in combination with the WRITE BUFFER command.

The IDD has four segmented 60-KB (61,440 bytes) data buffers. This command can specify the data byte position on the data buffer in units of bytes by using the buffer address in the range of X'000000' to X'00EFFF'.

The function of this command and the data to be transferred to the INIT by this command are specified in the "mode" field of CDB. Either transfer mode listed below can be selected.

"Mode" Bit 2	1	0	Transfer mode
0	0	0	Header + data, address not specified
0	0	1	Header + data, address specified
0	1	0	Data only, address specified
0	1	1	Buffer descriptor

(1) Mode = 0,0,0: Header + data, address not specified

When this mode is specified, data stored in the IDD data buffer is transferred to the INIT after the 4-byte header. Zero must be specified in the "buffer address" fields of CDB. The value of the "buffer ID" is allowed zero, one, two or three.

The "transfer byte-length" field of CDB specifies the total number of bytes of the header and buffer data that can be received by the INIT. The IDD adds the 4-byte header to the data read from the starting address X'000000' of the data buffer and transfers it to the INIT. Data transfer terminates when the header and buffer data that correspond to the number of bytes specified in the "transfer byte-length" field are transferred or when buffer data up to the last byte position (address X'00EFFF') of the header and IDD data buffer is transferred. When zero is specified in the "transfer byte-length" field, data transfer is not executed and this command terminates.

Figure 3.34 shows the format of data to be transferred to the INIT when this mode is specified.

		Bit							
		7	6	5	4	3	2	1	0
Header	Byte 0	0	0	0	0	0	0	0	0
	1	Effective buffer data length (MSB)							
	2	Effective buffer data length							
	3	Effective buffer data length (LSB)							
Data	4	Buffer data (byte 0)							
	5	Buffer data (byte 1)							
	≈	≈							
	n	Buffer data (byte n-4)							

Figure 3.34 READ BUFFER command: buffer data (Mode = 000, 001)

The "effective buffer data length" field of the header indicates the IDD data buffer size (byte-length). This value indicates the size of the IDD data buffer that can be used with the WRITE BUFFER or READ BUFFER command regardless of the length specified in the "transfer byte-length" field of CDB or the data length actually stored in the data buffer with the WRITE BUFFER command. When this mode is specified, the "effective buffer data length" field indicates the size of entire data buffer of the IDD. The length of buffer data to be transferred to the INIT by this command is the smaller of either [the value in the "transfer byte-length" field of CDB – 4 bytes] or the value indicated in the "effective buffer data length" field of the header.

(2) Mode = 0,0,1: Header + data, address specified

When this mode is specified, the format of the data to be transferred to the INIT is the same as that in Mode = 0,0,0. Data stored in the IDD data buffer is transferred to the INIT after the 4-byte header. Zero, one, two or three must be specified in the "buffer ID" field of CDB.

The address on the data buffer can be specified in the "buffer address" field of CDB in this mode.

The "transfer byte-length" field of CDB specifies the total number of bytes of the header and buffer data that can be received by the INIT. The IDD transfers data read from the byte position on the data buffer specified in the "buffer address" field of CDB to the INIT after the 4-byte header. Data transfer terminates when the header and buffer data corresponding to the number of bytes specified in the "transfer byte-length" field are transferred or when the header and buffer data up to the last byte position (address X'00EFFF') in the IDD data buffer are transferred. When zero is specified in the "transfer byte-length" field, data transfer is not executed and this command terminates.

The format and contents of the 4-byte header to be transferred in this mode are the same as in Mode = 0,0,0. The "effective buffer data length" field of the header indicates the size (byte-length) from the byte position on the data buffer specified in the "buffer address" field of CDB to the last byte position of the data buffer. The length of buffer data to be transferred to the INIT by this command is the smaller of either [the value in the "transfer byte-length" field of CDB − 4 bytes] or the value indicated in the "effective buffer data length" field of the header.

(3) Mode = 0,1,0: Data only, address specified

When this mode is specified, data to be transferred to the INIT is only the data read from the IDD data buffer, but the header such as in Mode = 0,0,0 or Mode = 0,0,1 is not transferred. Zero, one, two or three must be specified in the "buffer ID" field of CDB.

In this mode, the address on the data buffer can be specified in the "buffer address" field of CDB.

The "transfer byte-length" field of CDB specifies the total number of bytes that can be received by the INIT. The IDD transfers data read from the byte position on the data buffer specified in the "buffer address" field to the INIT. Data transfer terminates when buffer data corresponding to the number of bytes specified in the "transfer byte-length" field is transferred or when buffer data up to the IDD data buffer last byte position (address X'00EFFF') is transferred. When zero is specified in the "transfer byte-length" field, data transfer is not executed and this command terminates.

(4) Mode = 0,1,1: Buffer descriptor

When this mode is specified, the IDD transfers the 4-byte buffer descriptor only to the INIT. The attribute of the IDD data buffer is indicated in the 4-byte buffer descriptor. When this mode is specified, zero must be specified in the "buffer address" field of CDB. The IDD transfers data (buffer descriptor) that is the length specified in the "transfer byte-length" field of CDB or 4 bytes, whichever is smaller, to the INIT. When zero is specified in the "transfer byte-length" field, data transfer is not executed and this command terminates.

Figure 3.35 shows the format of the buffer descriptor to be transferred to the INIT when zero is specified in the "buffer ID" field of CDB. When a buffer ID more than three is specified in CDB, the IDD transfers the buffer descriptor with all bytes set to X'00' to the INIT.

	Bit								
	7	6	5	4	3	2	1	0	
Byte 0	X'00' Addressing boundary								
1	X'00' Buffer capacity (MSB)								(= 60 KB)
2	X'F0' Buffer capacity								
3	X'00' Buffer capacity (LSB)								

Figure 3.35 READ BUFFER command: buffer descriptor

The "addressing boundary" field of the buffer descriptor indicates the "exponent" when the addressing boundary on the data buffer that can be specified with the WRITE BUFFER or READ BUFFER command with a "power of 2". The IDD reports X'00' (= 2^0) to indicate that the address can be specified on the byte boundary. The "buffer capacity" field indicates the size in bytes of the data buffer that can be operated with the WRITE BUFFER or READ BUFFER command.

Note: Precautions for use of command

When the WRITE BUFFER and READ BUFFER commands are used in the multiple initiator or multiple task environment, commands issued by another initiator or task may change the contents of the data buffer before the READ BUFFER command is executed after execution of the WRITE BUFFER command terminates. To avoid such an occurrence, the READ BUFFER command should be linked to the WRITE BUFFER command and issued or the IDD should be reserved with the RESERVE UNIT command in the multiple initiator environment.

3.4.5 READ LONG

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'3E'							
	1	LUN			0	0	0	CORRCT	0
	2	Logical block address (MSB)							
	3	Logical block address							
	4	Logical block address							
	5	Logical block address (LSB)							
	6	0	0	0	0	0	0	0	0
	7	Transfer byte-length (MSB)							
	8	Transfer byte-length (LSB)							
	9	0	0	0	0	0	0	Flag	Link

This command reads the data field and ECC byte of the logical data block on the disk specified in the "logical block address" field of CDB and transfers them to the INIT. Normally, this command in combination with the WRITE LONG command is used to check the ECC function.

Only one data block is to be operated by this command. Figure 3.36 shows the format of the data to be transferred to the INIT by this command. The 8-byte ECC follows data that is equal to bytes of the logical data block length.

Data (n bytes)	ECC (8 bytes)

Figure 3.36 READ LONG command: transfer data format

When 0 is specified in the CORRCT (corrected) bit of CDB byte 1, bit 1, the IDD does not correct data read from the disk. When the CORRCT bit is 1, correctable data errors are corrected on the IDD data buffer, then transferred to the INIT.

The "transfer byte-length" field of CDB indicates the number of bytes of data to be transferred to the INIT by this command and a length of [logical data block length + 8 bytes] should be specified. When the "transfer byte-length" is specified as zero, this command only executes the seek operation to the cylinder/track where the logical data block specified in the "logical block address" field of CDB is present, but does not execute data transfer to the INIT, and then terminates.

When a length (other than 0) that does not match the data format on the disk is specified in "the transfer byte-length" field of CDB, this command does not execute data transfer to the INIT and terminates with the CHECK CONDITION status. In this case, sense data indicates the following contents, and the INIT can obtain the correct transfer byte-length from the contents.

- Sense key: 5 = ILLEGAL REQUEST
- Sense code/subsense code: 24-00 = Invalid field in CDB
- VALID bit: 1
- ILI bit: 1
- Information field:
  ("transfer byte-length" of CDB) − (logical data block length + 8) (*1)

  *1 A negative number is represented with a complement of 2.

The error recovery process during execution of this command complies with the setting of the MODE SELECT parameter (page code 1: read/write error recovery parameter, page code 21: additional error recovery parameter) except for the following cases:

- The ARRE and DTE flags are not applicable.
- The TB flag is treated as 1.
- The ERR and DCR flags are not applicable. Data check for correctable data complies with setting in the CORRCT bit of CDB.

3.4.6 WRITE LONG

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'3F'							
	1	LUN			0	0	0	0	0
	2	Logical block address (MSB)							
	3	Logical block address							
	4	Logical block address							
	5	Logical block address (LSB)							
	6	0	0	0	0	0	0	0	0
	7	Transfer byte-length (MSB)							
	8	Transfer byte-length (LSB)							
	9	0	0	0	0	0	0	Flag	Link

This command writes data transferred from the INIT in the logical data block on the disk specified in the "logical block address" field of CDB as the data field and ECC byte for that block. Normally, this command in a combination with the READ LONG command is used to check the ECC function.

Only one data block is to be operated by this command. Data to be transferred by the INIT by this command must have the same order and same length as the data to be transferred from the IDD to the INIT by the READ LONG command. In other words, the 8-byte ECC must follow the data corresponding the bytes of the logical data block length.

The "transfer byte-length" field of CDB indicates the number of bytes of data to be transferred from the INIT by this command. A length of [logical data block length + 8 bytes] should be specified. When zero is specified for the "transfer byte-length", this command only executes the seek operation to the cylinder/track where the logical data block specified in the "logical block address" field of CDB is present, but does not executes data transfer from the INIT, and then terminates.

When a length (other than 0) that does not match the data format on the disk is specified in the "transfer byte-length" field of CDB, the command does not execute data transfer from the INIT but terminates with the CHECK CONDITION status. In this case, the sense data indicates the following contents, and the INIT can obtain the correct transfer byte-length for the contents.

- Sense key: 05 = ILLEGAL REQUEST
- Sense code/subsense code: 24/00 = Invalid field in CDB
- VALID bit: 1
- ILI bit: 1

- Information field:

  (transfer byte-length of CDB) – (logical data block length + 8) (*1)

  *1 A negative number is represented with a complement of 2.

The error recovery process during execution of this command complies with setting of the MODE SELECT parameter (page code 1: read/write error recovery parameter, page code 21: additional error recovery parameter), but the AWRE and DTE flags are not applicable.

3.4.7 WRITE SAME

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'41'							
	1	LUN			0	0	0	LBdata	0
	2	Logical block address (MSB)							
	3	Logical block address							
	4	Logical block address							
	5	Logical block address (LSB)							
	6	0	0	0	0	0	0	0	0
	7	Block count (MSB)							
	8	Block count (LSB)							
	9	0	0	0	0	0	0	Flag	Link

This command repeatedly writes data transferred from the INIT in consecutive logical data blocks on the disk.

Data to be transferred by the INIT by this command is always a single logical data block.

When the LBdata (logical block data) bit of CDB is 0, data of a single logical data block transferred from the INIT is repeatedly written in consecutive logical data blocks in the specified range. When the LBdata bit is 1, the first 4 bytes of data transferred from the INIT replace the logical block address of each data block, and is written repeatedly in consecutive logical data blocks in the specified range.

The "block count" field of CDB specifies the number of logical data blocks to be written by this command. The IDD repeatedly writes the same data pattern (i.e., the first 4 bytes are the logical block address of each data block when the LBdata bit is 1) in consecutive logical data blocks in ascending order of addresses starting with the data block specified in the "logical block address" field and corresponding to the number specified in the "block count" field. When zero is specified in the "block count" field, the specified data pattern is repeatedly written in the area from the data block specified in the "logical block address" field to the last logical data block in the space to be accessed. (*1)

*1 Depending on the address value specified in the "logical block address" field of CDB, the space to be accessed by this command is either user space or CE space.

3.4.8 RECOVER ID

[Fujitsu unique command]

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	X'DA'							
	1	LUN			Cylinder address (MSB)				
	2	Cylinder address (LSB)							
	3	Head number							
	4	Physical sector number							
	5	Logical data block length (MSB)							
	6	Logical data block length (LSB)							
	7	Transfer sector count (MSB)							
	8	Transfer sector count (LSB)							
	9	0	0	0	0	0	0	Flag	Link

This command is not supported currently.

3.4.9 RECOVER DATA

[Fujitsu unique command]

		Bit 7	6	5	4	3	2	1	0
Byte	0	X'D8'							
	1	LUN			Cylinder address (MSB)				
	2	Cylinder address (LSB)							
	3	Head number							
	4	Physical sector number							
	5	Logical data block length (MSB)							
	6	Logical data block length (LSB)							
	7	Transfer block count (MSB)							
	8	Transfer block count (LSB)							
	9	0	0	0	0	0	0	Flag	Link

This command is not supported currently.

This page is intentionally left blank.

CHAPTER 4 SENSE DATA AND ERROR RECOVERY

4.1 Sense Data

4.2 Initiator Error Recovery (Recommended)

4.3 Disk Drive Error Recovery

This chapter describes the configuration and contents of the sense data reported to the INIT at error occurrence, the recommended error recovery procedure to be executed by the software inside the INIT, and the error recovery process executed inside the IDD. For the error recovery process for the error detected on the SCSI bus operation, refer to Chapter 3 in OEM Manual SCSI Physical Specifications.

4.1 Sense Data

When the IDD reports the CHECK CONDITION status or clears the command being executed or stacked because a fatal error related to the SCSI bus has been detected, the IDD generates sense data for the INIT which issued the command. The INIT can read the sense data by issuing the REQUEST SENSE command.

4.1.1 Sense data format and basic information

The IDD sense data is in the extended sense data format. Figure 4.1 shows the extended sense data format provided for the IDD.

Notes:

1. SCSI specifications permit byte 18 and succeeding bytes of the extended sense data to be arbitrarily defined for each drive. The length and format are dependent upon each drive. The length of the extended sense data is indicated in the sense data, so the INIT can know the effective length by analyzing the received sense data.

2. The REQUEST SENSE command terminates successfully even though a transfer byte-length shorter than that of the sense data provided with the object drive is specified. In that case, a part of the sense data is received but other information is lost. The INIT should read all sense data provided for the drive by checking the specifications of the connected drive.

	Byte	Bit 7	6	5	4	3	2	1	0
Basic information	0	Valid	X'70' or X'71' (error code)						
	1	X'00'							
	2	0	0	ILI	0	Sense key			
	3	MSB Information							
	4								
	5								
	6								LSB
	7	X'28' (additional sense data length)							
	8	MSB Command-inherent information							
	9								
	10								
	11								LSB
	12	Sense code							
	13	Subsense code							
	14	X'00'							
	15	SKSV	Sense key inherent information						
	16								
	17								
	18	0	SCSI ID			L U N			
	19	CDB operation code							
Additional information	20 ~ 47	Detailed information							

Figure 4.1 Extended sense data format

4.1.2 Sense data basic information

(1) Valid

When this bit is 1, the value indicated in the information field (bytes 3 to 6) is valid. When this bit is 0, the information field does not have valid information.

(2) Error code

This field indicates the format and type of sense data. The IDD always indicates X'70' (current errors) or X'71' (deferred errors) which means the extended sense data format in this field.

(3) ILI (incorrect length indicator)

When this bit is 1, the transfer byte-length requested by the command did not match the data block length on the disk. Commands that report 1 to this bit for the IDD are the READ LONG and WRITE LONG commands only. For details of these commands, see Subsections 3.4.5 and 3.4.6.

(4) Sense key

This field indicates the cause of sense data generation. Codes indicating more detailed reasons are indicated in the sense code field and subsense code field. Table 4.1 lists sense keys and their meanings.

(5) Information

This field indicates information related to the detected error. This field is valid when the Valid bit is 1. Depending on the command that generated the error, additional information may be indicated in the command-inherent information field. Information indicated in this field has the following meaning depending on the ILI bit value.

a) When the ILI bit is 0, the logical block address of the data block where the error occurred is indicated. If the error occurs with the RECOVER ID and RECOVER DATA commands, this field indicates the address of the data block where the error occurred in the format of cylinder address (bytes 3 to 4)/physical head number (byte 5)/sector number (byte 6).

b) When the ILI bit is 1, the difference between the transfer byte-length requested by the command and the actual data block length on the disk is indicated. When the difference is a negative value (i.e. the requested number of transferred bytes is smaller), it is represented with a complement of 2. Commands that indicate this information for the IDD are the READ LONG and WRITE LONG commands only. For details of these commands, see Subsections 3.4.5 and 3.4.6.

(6) Additional sense data length

This field indicates the length (bytes) of sense data for byte 8 and succeeding bytes. The value indicated in this field indicates the length of sense data equipped by the IDD regardless of the value specified in the "transfer byte-length" field of the CDB in the REQUEST SENSE command. The IDD sense data length is fixed at 48 bytes, so this field always indicates X'28' (40 bytes).

(7) Command-inherent information

This field indicates information inherent to the command that generated the error. The command that validates this field value for the IDD is the REASSIGN BLOCKS command only. For details of the information indicated in this field, see Subsection 3.3.2.

(8) Sense code and subsense code

Codes indicating the detailed reason for the error indicated by the sense key are reported to these fields. The INIT can determine the error recovery procedure (see Section 4.2) according to the sense key and values indicated in these fields. Table 4.2 lists definitions of sense codes and subsense codes.

(9) SKSV (sense key specific valid) and sense key inherent information

When the sense key indicates "RECOVERED ERROR [=1]", "MEDIUM ERROR [=3]", or "HARDWARE ERROR [=4]" and the SKSV bit indicates 1, this field indicates the number of internal retries of the IDD for recovery from the detected error as shown in Figure 4.2.

	Bit							
	7	6	5	4	3	2	1	0
Byte 15	SKSV	0	0	0	0	0	0	0
16	X'00'							
17	Executed retry count							

Figure 4.2 Sense key inherent information

Table 4.1 Sense keys

Sense key	Name	Explanation
0	NO SENSE	No particular sense key is present.
1	RECOVERED ERROR	① The last executed command completed successfully with some recovery operation performed by the IDD. When two or more errors occur and are recovered during processing of a command, the last error is reported. ② The MODE SELECT parameter value was rounded.
2	NOT READY	The disk drive is not accessible.
3	MEDIUM ERROR	An unrecoverable error was detected due to a defect of the medium or an error in the recorded data.
4	HARDWARE ERROR	The IDD detected the hardware error to which the recovery process cannot be applied during command execution or self-diagnostic test.
5	ILLEGAL REQUEST	An illegal value was detected in the parameter in the CDB or the parameter transferred by setting of a command. Or, setting of LUN is incorrect. When the IDD detects an illegal parameter in the CDB, the IDD terminates commands without rewriting the disk. When is detected in the parameter transferred from the INIT in the DATA OUT phase, contents of the disk may have been rewritten by the command.
6	UNIT ATTENTION	The UNIT ATTENTION condition occurred. (For details of the UNIT ATTENTION condition, see Section 1.5.)
7	DATA PROTECT	① In the area where read or write operation is prohibited, an attempt was made to execute an prohibited operation. In this case, the command is not executed. ② The SET LIMITS command was issued twice in a series of linked commands.
8	BLANK CHECK	Unused
9	(Reserved)	Unused
A	COPY ABORTED	Unused
B	ABORTED COMMAND	The IDD abnormally terminated the command being executed. Normally, the INIT can try recovery by re-issuing the command.
C	EQUAL	Unused
D	VOLUME OVERFLOW	Unused
E	MISCOMPARE	The source data from INIT did not match the data read from the medium.
F	(Reserved)	Unused

Table 4.2 Sense and subsense codes (1/5)

C	Q	Name	Explanation	Sense key
00	00	No additional sense information	No particular sense code is present.	0
			An attempt was made to read the read-prohibited area.	7
01	00	No index/sector signal	The Index or Sector signal was not detected in the specified period.	4
02	00	No seek complete	The seek or rezero seek operation did not complete in the specified period.	4
03	00	Peripheral device write fault	Write operation to the disk abnormally terminated.	1, 4
04	00	Logical unit not ready, cause not reportable	The disk drive is not accessible.	2
04	04	Logical unit not ready, format in progress	The drive is not accessible because it is being formatted.	2
09	00	Track following error	The track crossing pulse was detected during the track following state.	1, 4
0C	01	Write error recovered with auto reallocation	The error at write operation was recovered by the automatic alternate block allocation process.	1
0C	02	Write error, auto reallocation failed	The automatic alternate block allocation process was failed at the write operation.	3, 4
10	00	ID CRC or ECC error	A CRC error was detected in the ID field.	1, 3
11	00	Unrecovered read error	An unrecoverable error was detected when data was read.	3
11	04	Unrecovered read error, auto reallocation failed	The automatic alternate block allocation process was failed at the read operation.	3
12	00	Sync byte not found for ID field	Sync byte of the ID field cannot be detected.	1, 3
13	00	Sync byte not found for data field	Sync byte of the data field cannot be detected.	1, 3
14	01	Record not found	The desired data block (sector) could not be found.	3

C: sense code, Q: subsense code

Table 4.2 Sense and subsense codes (2/5)

C	Q	Name	Explanation	Sense key
15	01	Mechanical positioning error	A seek error occurred on the drive.	1, 4
15	02	Positioning error detected by read of medium	The cylinder address of the ID field did not match.	1, 4
17	01	Recovered data with retries	The data error was recovered by read retry.	1
17	02	Recovered read data with positive head offset	The data error was recovered by read retry accompanied by the head offset operation in the + direction. [+ direction: outer direction on the disk]	1
17	03	Recovered read data with negative head offset	The data error was recovered by read retry accompanied by the head offset operation in the − direction. [− direction: inner direction on the disk]	1
18	00	Recovered read data with error correction applied	The data error was immediately recovered by ECC correction.	1
18	01	Recovered read data with error correction and retries applied	The data error was recovered by ECC correction after read retry.	1
18	02	Recovered data with ECC and/or retries, data auto-reallocation	The data error was recovered by ECC correction and the automatic alternate block allocation process was applied.	1
18	80	Recovered data with ECC and/or retries, re-write applied	The data error was recovered by ECC correction and the data was recovered with re-writing to the same block.	1
19	00	Defect list error	An error was detected when the defect control information was read.	3
19	02	Defect list error in primary list	The error was detected while reading the defect list. (P list)	3
19	03	Defect list error in grown list	The error was detected while reading the defect list. (G list)	3

C: sense code, Q: subsense code

Table 4.2 Sense and subsense codes (3/5)

C	Q	Name	Explanation	Sense key
1C	01	Primary defect list not found	Defect list (P list) was not detected.	3
1C	02	Grown defect list not found	The defect list (G list) was not detected.	3
1D	00	Miscompare during verify operation	The data transferred form the INIT did not match the data read on the drive during verify byte check operation.	E
1D	80	Miscompare during self diagnostic operation	The data written by IDD did not match the data read on the drive during self diagnostic operation.	E
20	00	Invalid command operation code	CDB byte 0 (operation code) is invalid.	5
21	00	Logical block address out of range	A logical block address exceeding the maximum value of the drive was specified.	5
24	00	Invalid field in CDB	Setting in the CDB is incorrect.	5
25	00	Logical unit not supported	Invalid LUN was specified.	5
26	00	Invalid field in parameter list	Setting of the parameter list transferred from the INIT during command execution is invalid.	5
27	00	Write protected	An attempt was made to write in the write-prohibited area. Or, the write operation in inhibited by the external operator panel.	7
29	00	Power-on, RESET, or BUS DEVICE RESET occurred	State immediately after power-on. Or, state immediately after resetting by the RESET condition or BUS DEVICE RESET message.	6
2A	01	Mode parameters changed	Another INIT changed the MODE SELECT parameter value related.	6
2C	00	Command sequence error	The command issue sequence is incorrect.	5, 7
2F	80	Commands cleared by PRIORITY RESERVE command	Another INIT issued the PRIORITY RESERVE command and forcedly terminated the command being executed or stacked.	6
31	00	Medium format corrupted	The medium format is different from the original one. (Formatting was not performed after the data format setting was changed with the MODE SELECT command.)	3

C: sense code, Q: subsense code

Table 4.2 Sense and subsense codes (4/5)

C	Q	Name	Explanation	Sense key
31	01	FORMAT command failed	Formatting is not complete due to some reason, and re-formatting is needed.	3
32	00	No defect spare location available	No usable alternate block area is present. Or, the alternate block process cannot be performed due to the overflow of the control table.	4
32	01	Defect list update failure	Updating of the defect list (G list) was failed.	4
37	00	Rounded parameter	The MODE SELECT parameter specified by the command was rounded.	1
3D	00	Invalid bits in IDENTIFY message	1 was specified for the reserve bit of the IDENTIFY message.	5
3E	00	Logical unit has not self-configured yet	The IDD initial setup operation is not complete.	2
40	nn	Diagnostic failure on component "nn"	An error was detected in self-diagnostic test. ("nn" is a Fujitsu unique code (80 to FF).)	4
43	00	Message error	The message sent from the IDD was unreasonably rejected.	B
44	nn	Internal target failure	A hardware error was detected in the IDD. ("nn" is a Fujitsu unique code (80 to FF).)	4
45	80	Select/reselect failure	Response waiting timeout for the INIT was detected in RESELECTION phase.	1, B
47	nn	SCSI parity error	A parity error was detected in the SCSI data bus. nn = 00 : COMMAND phase 98 : MESSAGE OUT phase A1 : DATA OUT phase A3 : DATA IN phase 00 : Other phase	1, B
48	00	INITIATOR DETECTED ERROR message received	The INITIATOR DETECTED ERROR message was received from the INIT.	1, B
49	00	Invalid message error	Unsupported or illegal message was received.	1, B
4C	00	Logical unit failed self-configuration	The IDD initial setup failed. (System space information could not be read.)	4
4C	81	Calibration failed	Calibration failed before SA read.	4

C: sense code, Q: subsense code

Table 4.2 Sense and subsense codes (5/5)

C	Q	Name	Explanation	Sense key
4E	00	Overlapped commands attempted	A new command was issued from the same INIT to the same logical unit before execution of a command was completed.	B
5C	01	Spindle synchronized	The spindle synchronization was completed.	6
5C	02	Spindle not synchronized	The spindle synchronization was failed.	6
80	00	Command execution delay required	The command cannot be immediately executed because an internal process such as seek control automatic calibration is being executed.	B
90	00	Initiator's SCSI ID not identified	SCSI ID of the INIT was not sent during SELECTION phase, so the RESERVE UNIT command, RELEASE UNIT command, or PRIORITY RESERVE command cannot be executed.	5
C4	01	Motor speed failure	Abnormal rotational speed of the spindle motor was detected.	4
C4	02	VCM heat	Overcurrent of the voice coil motor (VCM) was detected.	4
C4	03	Spindle on time out	The spindle start operation failed.	4
C4	04	Spindle motor current failure	Overcurrent of the spindle motor was detected.	4
C4	05	Power failure	Power source voltages (+5, +12 VDC) was under the standard value.	4
C4	09	Drive hard error	Drive hard error was detected.	4
C4	0A	Calibration error	Calibration error was detected after the SA read.	4
C4	0B	Illegal drive command	Illegal drive command was issued to DSP.	4
C4	0C	Unexpected drove error	Unexpected drove error was detected.	4
E0	nn	Microprogram detected error	Microcode of the IDD detects the error such as logical conflict. ("nn" is a Fujitsu unique code (80 to FF).)	B

C: sense code, Q: subsense code

4.1.3 Sense data additional information

Bytes 18 to 47 of the sense data are the field defined by Fujitsu peculiarly, and indicate following additional information.

(1) SCSI ID

This field indicates the SCSI ID of the IDD that generates the sense data.

(2) LUN

This field indicates the logical unit number (LUN) for which the sense data is generated.

"0, X, X, X": indicates the LUN is not defined at the point that the sense data is generate.

"0, L, L, L": lower 3 bits (L, L, L) indicate the LUN.

(3) CDB operation code

This field indicates the operation code (value of byte 1 in CDB) of the command in which the error was occurred. When the sense data is generated regardless of the command issued by the INIT, this field indicates '0'.

(4) Details information

Information indicated in bytes 20 to 47 of the sense data is a Fujitsu unique information for analyzing the trouble and the contents of this information is not announced. However, the user should store all bytes of the sense data including this field into the error logging (see Subsection 4.2.3) and inform the field engineer of them for trouble analysis and repair.

4.2 Initiator Error Recovery (Recommended)

When a single command or a series of linked commands terminates without report of the GOOD status or COMMAND COMPLETE message, the INIT should execute error recovery processing according to the END state. This section explains the analysis procedure of a command execution END state and the recommended procedure of error recovery processing which the INIT executes according to the results.

4.2.1 END status analysis and error recovery

Figure 4.3 shows the model of an INIT processing flow from command issue to reception of the END status. BUS FREE error in the figure indicates that the SCSI bus goes to the BUS FREE phase even if the INIT has no intention of doing so. This error is usually detected by hardware of firmware of the SCSI control unit in the host system (INIT).

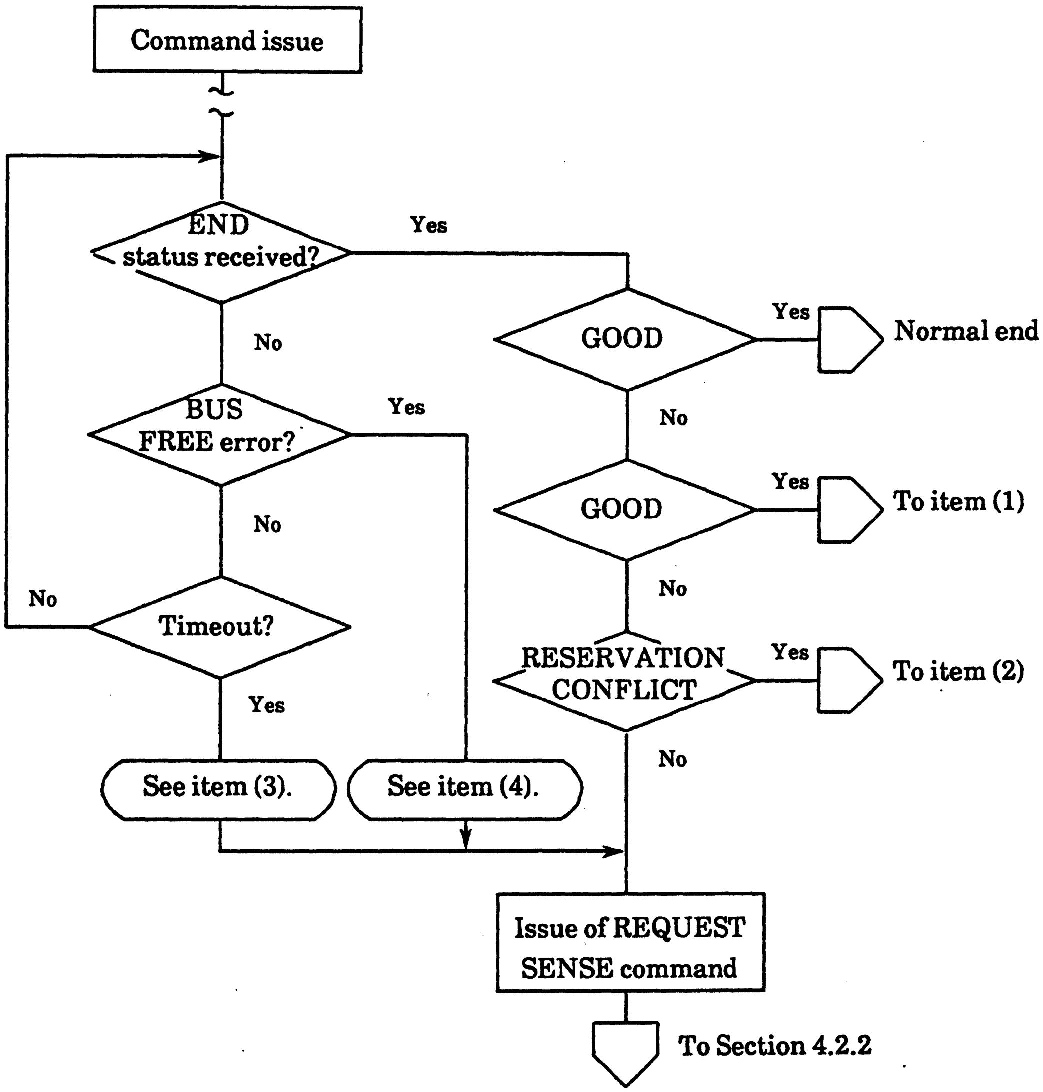


Figure 4.3 END status analysis

(1) BUSY status

This status indicates that the IDD is executing another command and a new command cannot be accepted. The INIT that received this status should issue the original command again.

When an INIT receives this status under a multiple initiator environment, it cannot estimate the time till the IDD can accept a next command, because the time depends on another INIT operation state. In this case, the INIT repeats issuing the command to which this status is reported.

Note:

The IDD has a command stack function. Therefore, this status is not generally sent when the INIT permits disconnect operation. (See Sections 1.2 and 1.4).

(2) RESERVATION CONFLICT status

This status indicates that the IDD is reserved by another INIT under a multiple initiator environment and that the IDD cannot be used until it is released from the reservation.

Although the system should manage the duration of the reservation status, the INIT that received this status issues the original command again after waiting for an appropriate time. Wait time until re-issue of the command is specific to the system. Another INIT operation status in the system must be considered to decide the waiting time.

If the INIT that reserves the IDD falls in an unrecoverable condition before releasing the reservation, another INIT cannot access the IDD because the RESERVATION status continues. In this case, the RESERVATION status must be forcibly released to recover the access right by one of the following:

- PRIORITY RESERVE command
- BUS DEVICE RESET message
- RESET condition

(3) Command complete wait timeout

This error indicates that the END status for the issued command is reported to the INIT at an unexpected time.

Following cases are the cause of this error: 1) the IDD operation is abnormal; 2) A command that an INIT issued is cleared by the PRIORITY RESERVE command or BUS DEVICE RESET message issued by another INIT, when the method of clearing the command cannot be explicitly detected by the INIT; 3) The command was cleared because an unrecoverable error was detected during reconnection processing executed by the IDD. If this error occurs, the INIT should get sense data by issuing the REQUEST SENSE command, and then decide how to recover from the error (see Subsection 4.2.2) depending an its contents.

The command execution time depends on not only the command type and the specification contents with the command but also the factors below. Caution and consideration should be taken when setting a timeout value with an INIT.

- Even if the IDD accepts a command successfully when other commands previously issued by another INIT are being executed or stacked, the command stack feature (see Section 1.4) lets the command wait for execution until the completion of other command. This waiting time depends on the type and specification contents of the commands issued by the other INIT.

- When the IDD executes error recovery processing (retry) for an SCSI bus error or disk drive error, the command complete time is longer than usual.

(4) BUS FREE error

This error indicates that the bus status enters the BUS FREE phase during execution of a command on the SCSI bus even if the INIT has no intention of doing so. This also occurs when another SCSI device on the bus generates the RESET condition. Generation of an unintended RESET condition should be processed as a system abnormal state.

Unrecoverable SCSI bus error, Serious protocol error, etc. are the cause of this error. For details of cause of error, refer to Chapter 3 in OEM Manual SCSI Physical Specifications. If this error occurs, INIT should get sense data by issuing the REQUEST SENSE command, and decide how to recover from the error (see Subsection 4.2.2) according to its contents.

4.2.2 Sense data analysis and error recovery

The INIT can decide the procedure for error recovery by analyzing the contents of sense data obtained by issuing the REQUEST SENSE command. Table 4.3 lists the classification of error information indicated in sense data. Table 4.4 lists the recommended procedure for error recovery processing executed by the INIT. The error recovery processing that can be executed by the INIT depends on system conditions. See the descriptions below, and select the procedure appropriate for the user system.

Table 4.3 Sense data error classification (1/5)

K	C	Q	Error contents	L	Recovery method (See Table 4.4.)
0	×	×	Sense data indicating the error contents is not pending.	None	4
1	0C 10 12 13 15 17 18	01 00 00 00 xx xx xx	Write error recovered with auto reallocation ID CRC or ECC error Sync byte not found for ID field Sync byte not found for data field Positioning error Recovered data without ECC Recovered data with ECC These errors detected at access to the disk are successfully recovered by the IDD retry processing. But when "1" is specified in DTE flag of MODE SELECT command, it is not always that the command is completed.	Need	8
1	45 47 48	xx nn 00	Select/reselect failure SCSI parity error INITIATOR DETECTED ERROR message received These errors detected during SCSI bus operation are successfully recovered by the IDD retry processing.	Need	0
1	37	00	Rounded parameter The rounding processing of the IDD was applied to the MODE SELECT para-meter specified by the INIT.	None	9
2	04	00	Logical unit not ready The logical unit specified by the command is unaccessible.	Need *1	10 *1
2	04	04	Logical unit not ready, format in progress The logical unit specified by the command is inaccessible because of formatting processing.	None	11

K: Sense key C: Sense code
Q: Subsense key L: Logging necessity (See Subsection 4.2.3.)

*1: Excluding the necessary time (approx. 20 seconds) from power-on or start instruction issurrance by the START/STOP command to the disk drive in in the ready state.

Table 4.3 Sense data error classification (2/5)

K	C	Q	Error contents	L	Recovery method (See Table 4.4.)
2	3E	00	Logical unit has not self-configured yet The initial setup (system information read) of the IDD is not completed yet.	None	5
3	0C 10 11 12 13 14	02 00 xx 00 00 01	Write error, auto reallocation failed ID CRC or ECC error Unrecovered read error Sync byte not found for ID field Sync byte not found for data field No record found These errors were detected while accessing the disk, but could not be recovered by the IDD retry processing.	Need	12
3	19 1C	xx xx	Defect list error Defect list not found The defect list on the disk cannot be read successfully.	Need	7
3	31 31	00 01	Medium format corrupted Format command failed The disk data format is incorrect.	Need	18
4	01 02 03 09 0C 32	00 00 00 00 02 01	No index/sector signal No seek complete Peripheral device write fault Track following error Write error, auto allocation failed Defect list update failure These errors were detected on the disk drive operation.	Need	7
4	15	xx	Positioning error These error are disk drive seek errors.	Need	12
4	32	00	No defect spare location available There is no usable alternate block in the disk drive.	None	19

K: Sense key C: Sense code
Q: Subsense key L: Logging necessity (See Subsection 4.2.3.)

Table 4.3 Sense data error classification (3/5)

K	C	Q	Error contents	L	Recovery method (See Table 4.4.)
4	40	nn	Diagnostic failure on component "nn" Error was detected in the IDD self-diagnostics.	Need	13
4	44 E0	nn nn	Internal target failure Microprogram detected error A hardware error or microprogram detected error was detected inside the IDD.	Need	20
4	4C	00	Logical unit failed self-configuration The initial setup (system information reading) of the IDD was unsuccessful.	Need	8
4	4C C4	81 xx	Calibration failed Drive failure or Power failure Initial seek was terminate abnormally. Or, severe error was detected in the drive controls.	Need	10
5	20 21 24 25 26 2C	00 00 00 00 00 00	Invalid command operation code Logical block address out of range Invalid field in CDB Logical unit not supported Invalid field in parameter list Command sequence error The command specification is invalid.	None	2
5	3D 90	00 00	Invalid bits in IDENTIFY message Initiator SCSI ID not identified A SCSI protocol error was detected in the command execution sequence.	None	3
6	29	00	Power on, reset, or BUS DEVICE RESET occurred The IDD was initialized by power-on, RESET condition, or BUS DEVICE RESET message.	None	14
6	2A	01	Mode parameters changed The MODE SELECT parameter was changed by another INIT.	None	21

K: Sense key   C: Sense code
Q: Subsense key   L: Logging necessity (See Subsection 4.2.3.)

Table 4.3 Sense data error classification (4/5)

K	C	Q	Error contents	L	Recovery method (See Table 4.4.)
6	2F	80	Command cleared by PRIORITY RESERVE A command being executed or stacked was forcibly terminated by the PRIORITY RESERVE command issued by another INIT.	None	17
6	5C	01	Spindle synchronized The spindle synchronization was completed.	None	22
6	5C	02	Spindle not synchronized The spindle synchronization was failed. The master index signal had faulty.	None	23
7	00 27	00 00	Read protected Write protected The read-/write-prohibited area was accessed.	None	15
7	2C	00	Command sequence error The SET LIMITS command was issued twice in a set of linked commands.	None	2
B	1B 43 45 47 48 49	nn 00 xx nn 00 00	Synchronous data transfer error Message error Select/reselect failure SCSI parity error INITIATOR DETECTED ERROR message received Invalid message error An unrecoverable SCSI error was detected in the command execution sequence.	Need	6
B	4E	00	Overlapped commands attempted Before the command issued by an INIT was completely executed, the INIT issued another command.	None	2
B	80	00	Command execution delay required A command cannot be immediately executed because internal processing (e.g., automatic setting of seek control) is being executed.	None	16

K: Sense key C: Sense code
Q: Subsense key L: Logging necessity (See Subsection 4.2.3.)

Table 4.3 Sense data error classification (5/5)

K	C	Q	Error contents	L	Recovery method (See Table 4.4.)
E	1D	00	Miscompare during verify operation The data comparison was unsuccessful during verify byte check operation.	None	24

K: Sense key C: Sense code
Q: Subsense key L: Logging necessity (See Subsection 4.2.3.)

Table 4.4 Error recovery procedure (1/4)

Recovery method	Recovery processing procedure
0	Error recovery processing is not necessary. Processing is continued.
1	Error recovery processing is impossible. Processing is terminated.
2	Error recovery processing is impossible due to a programming error. Terminate the processing, and correct the system (INIT) programming error.
3	Error recovery processing is impossible due to an SCSI protocol error. Terminate the processing, and check the system SCSI bus operation.
4	① Re-issue the original command. (Retry) ② If retry terminates with the same error, follow recovery method 3 or 1. ③ If retry terminates with another error, execute the recovery processing procedure for that error.
5	① After waiting for approx. 1 second, re-issue the original command. (Retry) ② If 10 retries are unsuccessful, follow recovery method 1.
6	① Re-issue the original command. (Retry) ② If 10 retries are unsuccessful, follow recovery method 3.
7	① Re-issue the original command. (Retry) ② If 10 retries are unsuccessful, follow recovery method 1.
8	① When the DTE flag of the MODE SELECT parameter is 0, follow recovery method 0. ② When the DTE flag of the MODE SELECT parameter is 1, execute either of the following: • Follow recovery method 0 when Valid bit is 1 and the address of the last data block in the data blocks specified by the command is indicated in the information field. • re-issue the original command regardless of whether the Valid bit is 0 or 1 when other than the address of the last data block in the data blocks specified by the command is not indicated in the information field. ③ If this error (except positioning error [15-xx]) occurs on the same data block, execute alternate block allocation processing (see Section 5.4).

Table 4.4 Error recovery procedure (2/4)

<table>
<tr><th>Recovery method</th><th>Recovery processing procedure</th></tr>
<tr><td>9</td><td>① Issue the MODE SENSE command, and read the current value of the MODE SELECT parameter changed when RECOVERED ERROR was reported.
② Follow recovery method 0 when the value set in the parameter page is in the allowable range specified by the INIT.
③ When the value set in the parameter page is not in the allowable range specified by the INIT, correct the specified value of the parameter, and re-issue the MODE SELECT command.</td></tr>
<tr><td>10</td><td>① Issue the START/STOP UNIT command (Immed = 0) for start indication.
② If the START/STOP UNIT command terminates abnormally, follow recovery method 1.
③ When the START/STOP UNIT command terminates successfully, re-issue the original command. (Retry)</td></tr>
<tr><td>11</td><td>① Wait until formatting (FORMAT UNIT command) terminates.
② Re-issue the original command. (Retry)</td></tr>
<tr><td>12</td><td>① Set the retry count of the MODE SELECT parameter to the default value.
② Re-issue the original command. (Retry) (*1)
③ If the first retry is unsuccessful, issue the REZERO UNIT command.
④ Re-issue the original command. (Retry) (*1)
⑤ If recovery is unsuccessful, perform either of the followings:
• Follow recovery method 1 when the sense key is HARDWARE ERROR [=4].
• Execute alternate block allocation processing (Section 5.4) when the sense key is MEDIUM ERROR [=3].
*1 When the original command is the REASSIGN BLOCKS command, the defect list may need to be reconfigured. See Subsection 3.3.2 for details of this command.</td></tr>
<tr><td>13</td><td>① Generate RESET condition, or issue the BUS DEVICE RESET message to the IDD that reported this error.
② After waiting for more than 2 seconds, re-issue the original command (Retry). If the starting mode of spindle motor is specified as "the starting by the command", the START/STOP command ("Immed" = 0) for indicating start is issued before the retry.
③ Follow recovery method 1 when the retry procedure (②) does not terminate successfully.</td></tr>
</table>

Table 4.4 Error recovery procedure (3/4)

Recovery method	Recovery processing procedure
14	① Since the MODE SELECT parameter is initialized to the saved value (default value when there is no saved value), issue the MODE SELECT command when the setting of a specific parameter is necessary. ② Re-issue the original command. (Retry)
15	① When the SET LIMITS command is not linked, the error cause is that the write operation is inhibited by the external operator panel. Release the protection, and re-issue the original command. If the write protection cannot be released, follow recovery method 2. ② When the SET LIMITS command is linked, follow recovery method 2.
16	① Set the DCED flag in the additional error recovery parameter (page 21) of the MODE SELECT parameter to 0. ② Re-issue the original command. (Retry)
17	When the PRIORITY RESERVE command issued by another INIT is valid with the system, perform recovery processing necessary for the system. If not, follow recovery method 2.
18	① Issue the MODE SENSE command, and check the values of parameters (block descriptor, page 3: format parameter, page 4: drive parameter) which are related to the disk data format. ② When the parameter values are correct, issue the FORMAT UNIT command, and initialize the disk. (*1) ③ When the parameter values are incorrect, issue the MODE SELECT command, re-specify necessary parameters, then initialize the disk using the FORMAT UNIT command. (*1) *1: Since the error is reported by the MODE SELECT or FORMAT UNIT command issued from another INIT in the system that several INITs are connected, confirm another INIT operation before applying this recovery method.
19	Initialize the whole disk. At this time, it is desirable to increase the number of spare sectors to as many as possible. If this error recurs, check the alternate block allocation processing procedure in the system or the installation environment of the disk drive.
20	① Re-issue the original command. (Retry) ② If recovery is unsuccessful after 10 retries, follow the recovery method 13.

Table 4.4 Error recovery procedure (4/4)

Recovery method	Recovery processing procedure
21	① Issue the MODE SENSE command, and read out the current value of the MODE SELECT parameter. ② If the parameter value is adequate, re-issue the original command. (Retry) **Note:** The UNIT ATTENTION condition may be generated by the error on the other INIT program. At this case, follow the recovery method 2 or confirm the other INIT operation when the parameter value read out at ① is inadequate.
22	Re-issue the original command (retry).
23	① Re-issue the original command (retry). ② Issue the MODE SELECT command if the spindle synchronization is needed.
24	① Issue the READ command, and read out data in the block in which the error occurs. ② Confirm the cause of data unmatch, and execute the data recovery process from the INIT.

4.2.3 Error logging

To collect useful maintenance information, it is desirable for the INIT to accumulate (logging) information of SCSI bus errors (e.g., BUS FREE error or command complete waiting timeout) detected by itself and error information reported by the IDD.

When 1 is specified for the PER bits of read/write error recovery parameter (page 1) and verify error recovery parameter (page 7) with the MODE SELECT command and the PSER bits of additional error recovery parameter (page 21), the INIT can obtain information on errors which are successfully recovered with the IDD internal error recovery processing. The INIT can obtain data to analyze the operation status of the disk drive by logging the error information.

For error logging contents, it is recommended to provide a time stamp to edit error-detected times in time series, and include the following information:

- Command issuer, issue destination SCSI ID of the SCSI device, and command issue destination LUN
- Issued CDB
- Acceptance status, or the code and contents of an error type detected by INIT when the status cannot be accepted
- All bytes of sense data reported by the IDD

4.3 Disk Drive Error Recovery

This section explains the error recovery processing method and procedure which the IDD executes for errors related to the disk drive. The INIT can control the error recovery processing method of the IDD by using the MODE SELECT parameters.

4.3.1 Error status and retry process

(1) Seek error

This error includes the mechanical operation abnormality of the disk drive during seek operation and the inconsistency of cylinder addresses detected at verification for the ID field.

If a seek error occurs, the IDD executes re-max seek, repositions the head at the objective cylinder, then restarts the original processing. If the error is not recovered, the retry processing (rezero seek and repositioning) is repeated according to the retry count specified by the "retry count for seek error" of the MODE SELECT parameter. If the error is not recovered with the first retry, the IDD performs the automatic readjustment of the positioning control feature, then continues the second and subsequent retry processing.

If the error is not recovered even after the retry processing has been executed by the specified number of times, the IDD terminates the currently executed command with the CHECK CONDITION status. At this time, the sense key of the sense data indicates "HARDWARE ERROR [=4]", and the sense code denotes "Mechanical positioning error [=15-01]" or "Positioning error detected by read of medium [=15-02]".

(2) ID field read error

This error is either of Sync Byte pattern not detected when reading ID field or the CRC error. If this error is detected, the IDD performs retry processing with rereading after waiting one disk revolution. When the error is recovered, the IDD restarts processing.

The retry processing with re-reading is repeated according to the retry count specified by the MODE SELECT parameter ("retry count for reading", "retry count for writing", or "retry count verification"). If the error is not recovered even after the retry processing has been executed by the specified number of times, the IDD terminates the currently executed command with the CHECK CONDITION status except the case that the automatic alternate block allocation processing (see Subsection 4.4.2) is applied. At this time, the sense key of the sense data indicates "MEDIUM ERROR [=3]", and the sense code denotes "Sync byte not found for ID field [=12-00]" or "ID CRC or ECC error [=10-00]".

In the retry processing for this error, the re-read operation with repositioning after re-max seek or with head offset processing or changing slice level is executed. Table 4.5 shows the processing method for each retry count

Table 4.5 Read error retry procedure

a. For read command

Retry count		1 st	2 nd	3 rd	4 th	5 th	6 th	7 th	8 th	9 th	10 th	11 th	12 th	13 th	14 th	15 th	16 th	17 th	18 th
Retry method (*1)	R	–	–	–	–	–	X	–	–	–	–	–	X	–	–	–	–	–	X
	O	0	+1	–1	+2	–2	0	0	+1	–1	+2	–2	0	0	+1	–1	+2	–2	0
	S	0	0	0	0	0	0	–	–	–	–	–	–	+	+	+	+	+	0

b. For write command

Retry count		1 st	2 nd	3 rd	4 th	5 th	6 th	7 th	8 th	9 th	10 th	11 th	12 th	13 th	14 th	15 th	16 th	17 th	18 th
Retry method (*1)	R	–	–	–	–	–	X	–	–	–	–	–	X	–	–	–	–	–	X
	O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S	0	0	0	0	0	0	–	–	–	–	–	–	+	+	+	+	+	0

Note:

When the retry count is specified as 19 or more, the retry method shown between the 1st and 18th are repeated in this order. The last retry is executed with no offset and normal slice level.

*1 R: Re-max seek
X: The head is repositioned after re-max cylinder seek, then retry is executed.
–: Retry is executed without seek operation.

O: Offset retry
0 indicates that there is no offset; other digits indicate that retry is performed by executing head offset processing. A plus sign indicates an offset in the outer direction, a mimus sign indicates that in the inner direction.
The offset value differ depending on the digit.

S: Slice level
0 indicates that retry is executed with normal slice level (45%).
A plus sign indicates that retry is executed with higher slice level (50%).
A minus sign indicates that retry is executed with lower slice level.

(3) Uncorrectable read error in data field

This error is either of Sync Byte pattern not detected when reading data field or uncorrectable ECC error. If this error is detected, the IDD performs the retry processing with re-reading after waiting one disk revolution. If the error is not recovered, the retry processing with re-reading is repeated according to the retry count specified by the MODE SELECT parameter ("retry count for reading" or "retry count for verification").

When the error becomes correctable during execution of retry with re-reading, procedures explained in item (4) below is performed. If the error is not recovered even after the retry processing has been executed the specified number of times, the IDD terminates the currently executed command with the CHECK CONDITION status. At this time, the sense key of the sense data indicates "MEDIUM ERROR [=3]", and the sense code denotes "Sync byte not found for data field [=13-00]" or "Unrecovered read error [=11-00]".

In the retry processing for this error, re-read operation with repositioning after rezero seek or with head offset processing is executed. (See Table 4.5)

(4) Correctable read error in data field

This error is a correctable ECC error that occurs when reading the data field. The processing performed when this error is detected depends on the values specified for the EER and DCR flags of the MODE SELECT parameter.

When the immediate error correction is permitted (DCR=0, EER=1), the IDD corrects the error at once on the data buffer, and sends the corrected data to the INIT. For verification processing, verification is regarded as successful when it is determined that correction is possible, and processing for subsequent data blocks is continued.

When the error correction is permitted but the immediate correction is not allowed (DCR=0, EER=0), retry processing with rereading is repeated according to the retry count specified by another MODE SELECT parameter ("retry count for reading" or "retry count for verification") in the same way as in item (3). When the result of the last re-reading indicates that correction is possible, the error is corrected. In this case, the last re-reading retry is executed using the same offset value as that when the correctable error was detected.

When the error correction is inhibited (DCR=1, EER=0), retry processing with re-reading is repeated according to the retry count specified by "retry count for reading" or "retry count for verification" in the same way as in item (3).

If the error is not recovered even after the retry processing has been executed, the IDD terminates the currently executed command with the CHECK CONDITION status. At this time, the sense key of the sense data indicates "MEDIUM ERROR [=3]", and the sense code indicates "Unrecovered read error [=11-00]".

(5) Target data block not detected

This error indicates that an inconsistency of data block numbers was detected at verification for the ID field. If this error is detected, the IDD performs the retry processing with re-reading after waiting one disk revolution. The re-reading retry is executed only once.

If the target data block cannot be detected even after the re-reading retry has been executed, the IDD terminates the currently executed command with the CHECK CONDITION status. At this time, the sense key of the sense data indicates "MEDIUM ERROR [=3]", and the sense code indicates "Record not found [=14-01]".

(6) Other errors in IDD

If an unrecoverable error other than (1) to (7) above is detected in the IDD, the IDD terminates the currently executed command with the CHECK CONDITION status. At this time, the sense key of the sense data indicates "HARDWARE ERROR [=4]".

4.3.2 Automatic alternate block allocation processing

The IDD have a function that the alternate block allocation process is automatically applied to the defective data blocks detected during the execution of the WRITE, WRITE EXTENDED, WRITE AND VERIFY, READ, or READ EXTENDED command.

(1) Automatic alternate block allocation process during write operation

This function is enabled by setting 1 to the AWRE flag in the MODE SELECT parameter. This function is applied only to the WRITE command, WRITE EXTENDED command, and the write operation of the WRITE AND VERIFY command (write operation).

If this function is enabled, when an unrecoverable error is detected in the ID field verification at writing to the data field even though the re-reading retry (see item (2) in Subsection 4.3.1) is performed according to the retry count specified by the MODE SELECT parameter ("retry count for writing"), the IDD allocates the alternate block to that data block and executes the write operation to the data field of the alternate block. The alternate block allocation processing is the same as those of the REASSIGN BLOCKS command.

This function is applied only once during the execution of one command. The alternate block allocation processing and write operation is executed to the first defective block detected during the command execution but, when the second defective block is detected, the currently executed command terminates with error at that time. However, the INIT can recover the error by re-issue the same command to apply the automatic alternate block allocation processing.

(2) Automatic alternate block allocation process during read operation

This function is enabled by setting 1 to the ARRE flag in the MODE SELECT parameter. This function is applied only to the following commands.

- READ command
- READ EXTENDED command

If this function is enabled, when an unrecoverable error is detected in the ID field verification at reading to the data field even though the re-reading retry (see item (2) in Subsection 4.3.1) is performed according to the retry count specified by the MODE SELECT parameter ("retry count for reading") or when the data correction by ECC (see item (4) in Subsection 4.3.1) is applied to the data field, the IDD allocates the alternate block to that data block and duplicates the corrected data of the original data block to the data field of the alternate block. The alternate block allocation processing is the same as those of the REASSIGN BLOCKS command. The corrected data duplicated to the alternate block is a data read with skipping the ID field or a data corrected by ECC.

This function is applied only once during the execution of one command. The alternate block allocation processing and duplication of data is executed to the first defective block detected during the command execution but, when the unrecoverable error is detected at the second defective block, the currently executed command terminates with error at that time. However, the INIT can recover the error by re-issue the same command to apply the automatic alternate block allocation processing.

Notes:

1. When this function is specified (ARRE = 1), the EER flag setup in the MODE SELECT parameter is invalid, and the re-reading retry is performed according to the retry count specified by the "retry count for reading". If 1 is specified in the DCR flag, data correction for the data field is performed next to the above process (see item (4) in Subsection 4.3.1). When 1 is specified in the DCR flag, the data correction by ECC is inhibited. Therefore, even if 1 is specified in ARRE flag, automatic alternate block allocation processing is not applied.

2. When the error in the data field is recovered by the ECC correction, the re-writing the corrected data and verification check (re-reading) is performed to the data block before applying the alternate block allocation. When the error is corrected by re-writing, the alternate block allocation is not applied to the data block.

3. Even if the write operation is inhibited by the SET LIMITS command, the automatic alternate block allocation process is executed. However, when the write operation is inhibited by the external operator panel, specifying the automatic alternate block allocation processing is invalid.

4.3.3 Error recovery process control

The INIT can control the error recovery processing (retry) which the IDD executes for errors related to the disk drive, by specifying MODE SELECT parameters.

Table 4.6 lists the error types and the retry count specification for each error type. When retry count can be specified with the INIT, it is recommended that the default value or a greater value be specified. If error recovery processing fails and the command terminates abnormally when specifying a value less than the default value, the command should be reexecuted by setting the default value for the corresponding MODE SELECT parameter in the INIT.

Followings shows the type of MODE SELECT parameters that the INIT can specify for the error recovery process of the disk drive. For details of the function of each parameter, refer to the explanation of the MODE SELECT parameter (Subsection 3.1.4). The INIT can select the error recovery flag and the retry count by issuing the MODE SELECT command, if necessary.

a. Read/write error recovery parameters (page code 1)

- AWRE (automatic write reallocation enabled)
- ARRE (automatic read reallocation enabled)
- TB (transfer block)
- EER (enable early recovery)
- PER (post error)
- DTE (disable transfer on error)
- DCR (disable correction)
- Retry count for reading (see Table 4.6)
- Retry count for writing (see Table 4.6)

b. Verification error recovery parameters (page code 7)

- EER (enable early recovery)
- PER (post error)
- DTE (disable transfer on error)
- DCR (disable correction)
- Retry count for verification (see Table 4.6)

c. Additional error recovery parameters (page code 21)

- PSER (Post SCSI Error)
- Retry count for seek error (see Table 4.6)

Table 4.6 Disk drive errors and retry counts

Item	Error type	Retry count specification (default value)
1	Seek error	Retry count for seek error (2)
2	ID field read error	Retry count for reading, writing, or verification (18)
3	Data field read error	Retry count for reading or verification (18)
4	Target data block not detected	Not specifiable (retry count: 1)

Note:

Except for the following cases, a retry count is counted for each processing of one physical sector

- Retry count is counted separately for the ID field and data field. However, if an ID field read error is detected when retry processing for the data field is being executed during read operation, the retry processing is continued counting the retry count for the data field.

- When processing a data block to which an alternate sector has been assigned, the retry count is counted separately for the defect sector and the alternate sector.

This page is intentionally left blank.

CHAPTER 5 DISK MEDIUM MANAGEMENT

5.1	Defect Management
5.2	Disk Initialization
5.3	Data Block Verification Method (Recommended)
5.4	Alternate Block Allocation

This chapter describes the defect management method of the disk, countermeasures for defective disk medium occurring during operations, and disk management methods such as reinitialization procedures. For the details on the data recording format on the disk, refer to Chapter 3 of OEM Manual Specifications & Installation.

5.1 Defect Management

(1) Defect lists

The lists explained below contain the information related to defect position on the disk. For details of the defect lists, see the explanation of the FORMAT UNIT command (Subsection 3.3.1).

- P list: This list contains the information such as the positions of the defects which are found when the disk drive is delivered from the factory. The defective portions listed in the P list are the permanent defects and the contents of this list must be reflected to the alternate block allocation.
- D list: This list shows the positions of the defects specified by the INIT using the FORMAT UNIT command during disk initialization.
- C list: This list shows the positions of the defects detected during the initialized data block verification (certification) of the FORMAT UNIT command. This information is internally generated by the IDD during execution of the FORMAT UNIT command.
- G list: This list contains the information such as the positions of the defective logical data blocks specified by the INIT with the REASSIGN BLOCKS command, positions of the defective logical data blocks for which alternate blocks are allocated by the IDD automatic alternate block allocation, information specified for the D list, and information generated for the C list.

The information of the P and G lists is stored on the system area of the disk. This information can be read by the INIT with the READ DEFECT DATA command.

(2) Alternate block allocation

An alternate data block is assigned as a defect section unit to each defective data block (sector) on the disk by the IDD internal defect management method.

a. Spare sectors

The spare sectors to be used for alternate block allocation are reserved in either one or both of the areas described below. The location and size of each spare sector can be specified by the INIT at the disk initialization. For details, refer to Subsection 3.1.2 of OEM Manual Specifications & Installation.

- Spare sectors on cylinder: Spare sectors are reserved on the last track of each cylinder and are used as alternate blocks for the defective sector on the same cylinder.

- Alternate cylinder: This is a cylinder used exclusively for alternate block allocation. As many alternate cylinders as specified are reserved from the highest cylinder in the user space. The spare sectors on the alternate cylinders are used only when all the spare sectors on the same cylinder are used and no more space is available for alternate block allocation.

b. Alternate block allocation method

One of the following methods is used for alternate block allocation.

- Sector slip processing: Skips the defective sector and assigns the logical data block corresponding to the defective sector to the physically adjacent sector. This processing is performed until all the spare sectors on the cylinder of the defective sector are used up.

- Alternate sector processing: Assigns the data block corresponding to the defective sector to an unused spare sector on the same cylinder or on the alternate cylinder.

c. Alternate block allocation processing

The alternate block allocation is implemented by the FORMAT UNIT, REASSIGN BLOCKS command, or automatic alternate block allocation processing. For the sector slip processing, the logical data block except for the defective sector is assigned to the physically adjacent sector. However, for the alternate sector processing, the logical data block is assigned to a spare sector. In this case, the spare sector is not physically contiguous to or before the logical data blocks. For details, see Subsection 3.3.2 of OEM Manual Specifications & Installation.

(a) Alternate block allocation during FORMAT UNIT command execution

For the FORMAT UNIT command, alternate blocks are assigned to the defective sectors based on the specified defect list (P, G, or D list) until all spare sectors on the same cylinder are used up by the sector slip processing. The spare sectors on the alternate cylinders are assigned to the defective sectors detected after all the spare sectors on the same cylinder are used up by the alternate sector processing.

When the data block verification (certification) is not inhibited, the IDD reads all the initialized data blocks and verifies them after the disk is initialized as described above. If a defective data block is found during this processing, the IDD generates the C list as defect position information and assigns an alternate block to the defective data block by the alternate sector processing. In this case, when there is a spare sector on the same cylinder where the defective sector is found, the alternate block is assigned within the same cylinder. However, if all the spare sectors on the same cylinder are already used, the alternate block is assigned to the spare sector on the alternate cylinder.

(b) Alternate block allocation by REASSIGN BLOCKS command

For the REASSIGN BLOCKS command, alternate blocks are assigned to the defective logical data blocks specified by the INIT by the alternate sector processing. In this case, when there is a spare sector on the same cylinder where the defective logical block is found, the alternate block is assigned within the same cylinder. However, if all the spare sectors on the same cylinder are already used up, the alternate block is assigned to the spare sector on the alternate cylinder.

(c) Automatic alternate block allocation

When the automatic alternate block allocation is allowed by the AWRE and ARRE flags of the MODE SELECT parameter, the IDD automatically assigns alternate blocks to the defective data blocks detected during execution of the WRITE, WRITE EXTENDED, WRITE AND VERIFY, READ, or READ EXTENDED command. The method of alternate block allocation is the same as the REASSIGN BLOCKS command. For details on the automatic alternate block allocation, see Subsection 4.3.2.

5.2 Disk Initialization

5.2.1 Initialization at installation

When a disk drive is delivered from the factory, it is initialized with the default data format (different for each model (type/standard)). Therefore, disk initialization (formatting) is not required at installation. However, if the desired data attributes are different from the default, the entire disk can be initialized (formatted) as described below. At this time, the following attributes can be changed by the INIT:

- Logical data block length
- Number of logical data blocks or number of cylinders in the user space
- Size of spare areas for alternate blocks

(1) MODE SELECT command issuance

Specify the format attributes of the disk by the MODE SELECT command. The following parameters must be specified in the MODE SELECT command.

a. Block descriptor

Specify the logical data block size (bytes) in the "data block length" field. In order to explicitly specify the number of the logical data blocks, specify it in the "data block count" field. When zero is specified in the "data block count" field, the number of logical data blocks after initialization is determined based on the values of the format parameter (page 3) and the drive parameter (page 4).

b. Format parameter (page 3)

Specify the number of the sectors to be reserved for each cylinder in the "number of alternate sectors / drive" field and also specify the number of tracks for the alternate cylinders (= number of alternate cylinders × number of disk drive heads) in the "number of alternate tracks/zone" field. The values specified in these fields should not be smaller than the default values of the IDD.

c. Drive parameter (page 4)

In order to explicitly specify the number of the cylinders in the user space, specify it in the "cylinder count" field. Note that the number of alternate cylinders specified by the format parameter (page 3) for the number of cylinders in the user space will be included in this value. When not specifying the number of cylinders, specify zero or the default value in the "cylinder count" field. The smaller of the following values is assigned in the user space: the number of cylinders required to reserve the logical data blocks specified in the "data block count" field in the block descriptor, or the maximum number of cylinders which can be used for user space in the disk drive. Note that if both the "cylinder count" field and "data block count" field in the block descriptor have zero specified, the maximum number of cylinders which can be used for user space in the disk drive is assumed.

(2) FORMAT UNIT command issuance

Issue the FORMAT UNIT command to implement initialization of the entire disk. For the FORMAT UNIT command, if a defective block is detected during initialization of the entire disk based on the P list, during data block verification after initialization, or during verification operations, the alternate block is assigned for the defective data block. The value set in the "initializing data pattern" field of CDB is written into all bytes of all logical data blocks by initialization. Only defect position information of the defective blocks detected in the verification is recorded into the G list. Specify the following parameters:

a. CDB setting

Specify 0 for the FmtData and CmpLst bits, 000 in the "defect list format" field and data pattern to be written into the data block at initialization in the "initializing data pattern" field.

b. Format parameter

When CDB is set as described above, this parameter is not required above.

5.2.2 Re-initialization

The disk drive after use can be re-initialized by the INIT with the FORMAT UNIT command.

Note:

When the data on the disk must be restored after re-initialization, such operations, including data saving before re-initialization and data restoring after re-initialization, must be done by the system software.

The size and allocation of the spare sector area and also the number of logical blocks can be changed by the INIT at re-initialization. The recommended re-initialization procedures are described below. Note that the initialization procedures at installation (Subsection 5.2.1) are applied to change the logical data block length.

(1) MODE SELECT command

When changing the number of logical data blocks (user space size) or size and location of the spare sector area, the INIT issues the MODE SELECT command to change the format attributes on the disk. The way of specifying the parameters is the same as the initialization procedures at installation (Subsection 5.2.1). When the current format attributes are continuously used after re-initialization, this command is not required.

(2) FORMAT UNIT command

Issue the FORMAT UNIT command with one of the following formats and initialize the entire disk.

a) Specify 1 for the FmtData bit, 0 for the CmpLst bit, 000 in the "defect list format" field, and the value of the data pattern for initialization in the "initializing data pattern" field. Also, prepare only a 4-byte header as the format parameter and specify 0 for the FOV, DPRY, DCRT, and STPF bits and specify zero for the "defect list length" field.

- For this FORMAT UNIT command, the alternate block is assigned for the defective block detected during the disk initialization based on the P list and existing G list, data block verification after initialization, or verification operations. The existing G list is stored with the defect position information of the defective blocks detected during the verification operations. Because the sector slip processing is applied to the alternate block allocation for the previously-detected defects during initialization, the optimization of the logical data block allocation on the disk is also achieved.

b) Specify 0 for the FmtData and CmpLst bits on CDB, 000 in the "defect list format" field, and the value of the data pattern for initialization in the "initializing data pattern" field. In this case, the format parameter can be omitted.

- For this FORMAT UNIT command, the alternate block is assigned for the defective block detected during the disk initialization based on only the P list, data block verification after initialization, or verification operations. The existing G list is erased and the defect position information of the defective blocks detected during the verification operations is stored in the G list.

5.3 Data Block Verification Method (Recommended)

This section describes the recommended verification procedures for the logical data blocks allocated on the disk. The data block verification should be implemented by the INIT after initialization of the disk or during alternate block allocation with the REASSIGN BLOCKS command.

Note:

The defective data blocks included in the P list cannot be assumed to be the normal blocks regardless of the results of the verification.

The INIT writes the special data pattern in the logical data blocks subjected to the verification, then it reads or verifies the data in the data blocks. The verification procedures are described below.

(1) Parameter setting by MODE SELECT command

Issue the MODE SELECT command to inhibit the retry processing, data correction by ECC, and operations of the Read-Ahead cache feature. Specify the parameters as follows:

a. Read/write error recovery parameter (page 1)

- AWRE = 0, ARRE = 0
- TB = 0 or 1
- EER = 0, PER = 1, DTE = 1, and DCR = 1
- Retry count for reading = 0
- Retry count for writing = 0

b. Verify error recovery parameter (page 7)

- EER = 0, PER = 1, DTE = 1, and DCR = 1
- Retry count for verification = 0

c. Caching parameter (page 8)

- RCE = 1

(2) Data pattern writing

Write the verification data pattern in the data blocks subjected to verification using the WRITE, WRITE EXTENDED, or WRITE SAME command. Data pattern X'BBBA645A27BB' is recommended for this purpose.

(3) Data read confirmation

Issue the READ, READ EXTENDED, or VERIFY command and confirm that the data written on the disk can be successfully read.

Data read confirmation should be implemented at least twice for each data block. The number of times it is implemented should be determined based on the system conditions. When all the data is successfully read, the data block can be assumed to be a normal block.

(4) Error verification

If a disk error (sense key = 3: MEDIUM ERROR) occurs during data pattern writing [(2)] or data read confirmation [(3)], perform the series of rewriting and rereading confirmation twice, for the data block where the error occurred. Repeat the series at least eight times.

If the same error occurs during retries even once, the data block should be treated as a defective block. When all the retries successfully terminate, the data block can be treated as a normal block.

5.4 Alternate Block Allocation

When an unrecoverable error is detected on the disk or a recoverable error repeatedly occurs in the same data block, an alternate block should be assigned for the data block in which the error occurred using the REASSIGN BLOCKS command. This section describes the procedures of alternate block allocation.

Note:

> The IDD tries to duplicate the contents of the data field in the logical data blocks specified by the defect data list into the alternate blocks assigned by this command, but sometimes fails. The INIT should confirm the contents of the assigned alternate blocks, saves data before issuing this command, and restores data after execution of this command.

① Verify the data block in which the error occurred according to the procedures described in Section 5.3. When it is determined that the data block is normal, alternate block allocation is not required. In this case, rewrite the original data into the data block as required. If the data block is judged to be defective, perform further operations as described below.

② Issue the REASSIGN BLOCKS command specifying the logical block address of the defective data block.

③ After the REASSIGN BLOCKS command successfully terminates, verify the logical data block based on the procedures described in Section 5.3. When the data block is determined to be a normal block, terminate the alternate block allocation processing. If the data block is judged to be a defective block, issue another REASSIGN BLOCKS command as in step ② and reassign another alternate block for the defective logical data block.

The history of alternate block allocation (defect position information) is recorded in the defect list (G list). Therefore, when the use of the G list is specified during re-initialization with the FORMAT UNIT command (Subsection 5.2.2), the history of the previous operations can be reflected to the defect processing. Note that because the alternate sector processing is applied to the alternate block allocation with the REASSIGN BLOCKS command, the physical sequence of the logical data block allocation on the disk is not assured. However, when the number of the defective sectors in the cylinder is not greater than the number of spare sectors in the same cylinder, as the sector slip processing is applied, the logical data blocks are reallocated so that they are in the physical sequence.

FUJITSU COMPUTER PRODUCTS OF AMERICA, INC.
2904 Orchard Parkway
San Jose, CA 95134

FUJITSU COMPUTER PRODUCTS OF AMERICA, INC.

Reader Comment Form

We would appreciate your comments and suggestions for improving this publication.

Publication No.	Rev. Letter	Title	Current Date

How did you use this publication?

- ☐ Learning
- ☐ Reference
- ☐ Installing
- ☐ Maintaining
- ☐ Sales
- ☐ Operating

Is the material presented effectively?

- ☐ Fully Covered
- ☐ Well Illustrated
- ☐ Well Organized
- ☐ Clean

What is your overall rating of this publication?

- ☐ Very Good
- ☐ Good
- ☐ Fair
- ☐ Poor
- ☐ Very Poor

What is your occupation?

Your other comments may be entered here. Please be specific and give page, paragraph and line number references where applicable.

Your Name & Return Address

Thank you for your interest. (Fold & fasten as shown on back, no postage needed if mailed in U.S.A.)

Fold

No Postage
Necessary
If Mailed
in the
United States

BUSINESS REPLY MAIL

First Class Permit # 7446 San Jose

Postage will be paid by Addressee

FUJITSU COMPUTER PRODUCTS OF AMERICA, INC.
Technical Support Dept.
2904 Orchard Parkway
San Jose, CA 95134

Fold

FUJITSU COMPUTER PRODUCTS OF AMERICA, INC.
2904 Orchard Parkway, San Jose, California 95134-2017, U.S.A.
TEL: (408) 432-633 FAX: (408) 894-1709