SCSI-II PROTOCOL CONTROLLER

MB86604A/MB86604L
DESIGN MANUAL

FUJITSU

MB86604A/MB86604L
SCSI-II PROTOCOL CONTROLLER
DESIGN MANUAL

1. The products described in this manual and the specifications thereof may be changed without prior notice. To obtain up-to-date information and/or specifications, contact your Fujitsu sales representative or Fujitsu authorized dealer.
2. Fujitsu will not be liable for infringement of copyright, industrial property right, or other rights of a third party caused by the use of information or drawings described in this manual.
3. The contents of this manual may not be transferred or copied without the express permission of Fujitsu.
4. The products contained in this document are not internded for use with equipments which require extermely high reliability such as aerospace equipmenta, undersea repeaters, nuclear control systems or medical equipments for life support.
5. Some of the products described in this manual may be strategic materials (or special technology) as defined by the Foreign Exchange and Foreign Trade Control Law. In such cases, the products or portions thereof must not be exported without permission as defined under the Law.

Contents

Figures

Tables

CHAPTER 1 OVERVIEW

This chapter describes SPC functional features and SPC block configuration.

1.1 SCSI PROTOCOL CONTROLLER FEATURES

The MB86604A/MB86604L are SCSI protocol controllers (below, SPC) based on the ANSI (SCSI-2) standard. By internalizing a sequence control function for the bus phases, the host MPU burden is diminished.

High-speed synchronous transfer is supported. Also, macro-commands which combine multiple commands can be defined in the SPC.

■ Features

SPC have the following features.

(1) SCSI Specifications

- Operation as an initiator or target.
- Transfers two data types.

  Synchronous transfer: Maximum 10 MB/second, offset value of 1 to 32 bytes can be set.

  Asynchronous transfer: Maximum 5 MB/second.
- Supports programmable commands (internal program memory size: 256 bytes).
- Contains 32-byte FIFO for data phases.
- Transfer block number can be set up to 64 K and transfer block length to 16 MB.
- Contains two 32-byte memory-type data buffers (send, receive) for message, command, and status phases.
- Supports two Automatic information receive modes.

  Initiator: Receives information about new target transition phase without issuing command.

  Target: Receives information for attention condition produced by initiator without issuing command.
- Supports two Automatic selection/reselection response modes.

  Initiator: Operation up to receiving message without issuing command.

  Target: Operation up to receiving command without issuing command.
- Specifies transfer parameter by ID (connected device).
- Contains 48 mA single-ended driver/receiver.

  Tri-state buffers for the REQ, ACK, DB0 to DB7, DBP, MSG, C/D, I/O, and ATN pins.
- On-chip SCAM registers for SCAM Level-1 protocol (only for MB86604L)

(2) System Specifications

a. Supports separate 16-bit MPU bus and DMA bus.

b. Direct connection with 68 series/80 series MPU.

c. Supports two types of transfers.

- Program transfers
- DMA transfers (Handshake/burst mode)

Figure 1.1a shows a drawing of the SPC packages. See Appendix A for detailed outer dimensions of SPC packages.

Plastic QFP, 100 pins

MB86604A

Figure 1.1a Package Drawing

Plastic SQFP, 100 pins
MB86604L

Figure 1.1b Package Drawing

1.2 BLOCK CONFIGURATION

The SPC consists of the following nine blocks.

- **Internal processor**
- **Timer**
- **Phase control**
- **Transfer control**
- **Registers**
- **Receive message, command, status buffer**
- **Send message, command, status buffer**
- **Command user program memory**
- **Data register**

■ Block Configuration

The SPC consists of the following nine blocks.

(1) Internal processor

Internal processor for sequence control among the bus phases. The max. operating clock frequency is 20 MHz.

(2) Timer

Manages SCSI time conditions.

Performs other time management as follows.

- Time until REQ/ACK signals are asserted for data in asynchronous transfers.
- Time until selection/reselection are retried.
- Selection/reselection timeout time.
- REQ/ACK timeout time during transfers.

  Asynchronous (target):After REQ signal is asserted, time until initiator asserts ACK signal.

  Asynchronous (initiator):After ACK signal is asserted, time until target negates REQ signal.

  Synchronous (target only):After REQ signal is sent, time until ACK signal which defines offset at 0 is received from initiator.
- SPC Timeout Timer: Indicates that SPC has been in busy for a time longer than time specified.

(3) Phase control

Controls SCSI arbitration, selection/reselection, data in/out, command, status, and message in/out phases.

(4) Transfer control

Controls SCSI information (data, command, status, message) transfer phases.

There are two types of transfer phase controls.

Asynchronous: Interlock control of REQ signal and ACK signal.

Synchronous: Control by offset value up to 32-byte maximum in data in/out phase.

There are also two data transfer modes.

Program transfer: Data register transfer via MPU interface.

DMA transfer: Transfer using DREQ and $\overline{\text{DACK}}$ signals via DMA interface.

Transfer parameters in synchronous transfers (minimum repeat cycle for sending REQ or ACK from SPC during transfer mode and synchronous transfer, maximum offset value) can be saved by ID and automatically set at the begiinning of data phase.

The number of transfer bytes is defined by the block length and number of blocks.

(5) Registers

The main registers are listed below.

Command register: Commands are defined by 8-bit code. For user program operation, specify the head address of the program allocated to user program memory.

Nexus status register: Show chip operation state, nexus destination ID, and data register state.

SCSI control signal status register: Shows SCSI control signal state.

Interruptstatus register: Describes interrupt state with 8-bit code.

Command step register: Describes command execution state with 8-bit code. Cause of errors can be analyzed by reference to the interrupt status register and this command step register.

Group 6/7 command length define register: Sets the command length for group 6/7 which are undefiend by the SCSI standard. Once this register is set, the command length for group 6/7 can be determined.

(6) Message, command, status receive buffer (Receive MCS buffer)

Receive information buffer. Stores message, command, and status information received by SCSI (32 bytes).

(7) Message, command, status send buffer (Send MCS buffer)

Send information bufffer. Stores message, command, and status information sent by SCSI (32 bytes).

(8) User prgram memory

Program memory for setting programmable commands (256 bytes).

(9) Data register

FIFO data register for storing data phase data (32 bytes).

1.3 BLOCK DIAGRAM

Block diagram for the MB86604A/MB86604L is shown in Figures 1.3.

■ Block Diagram

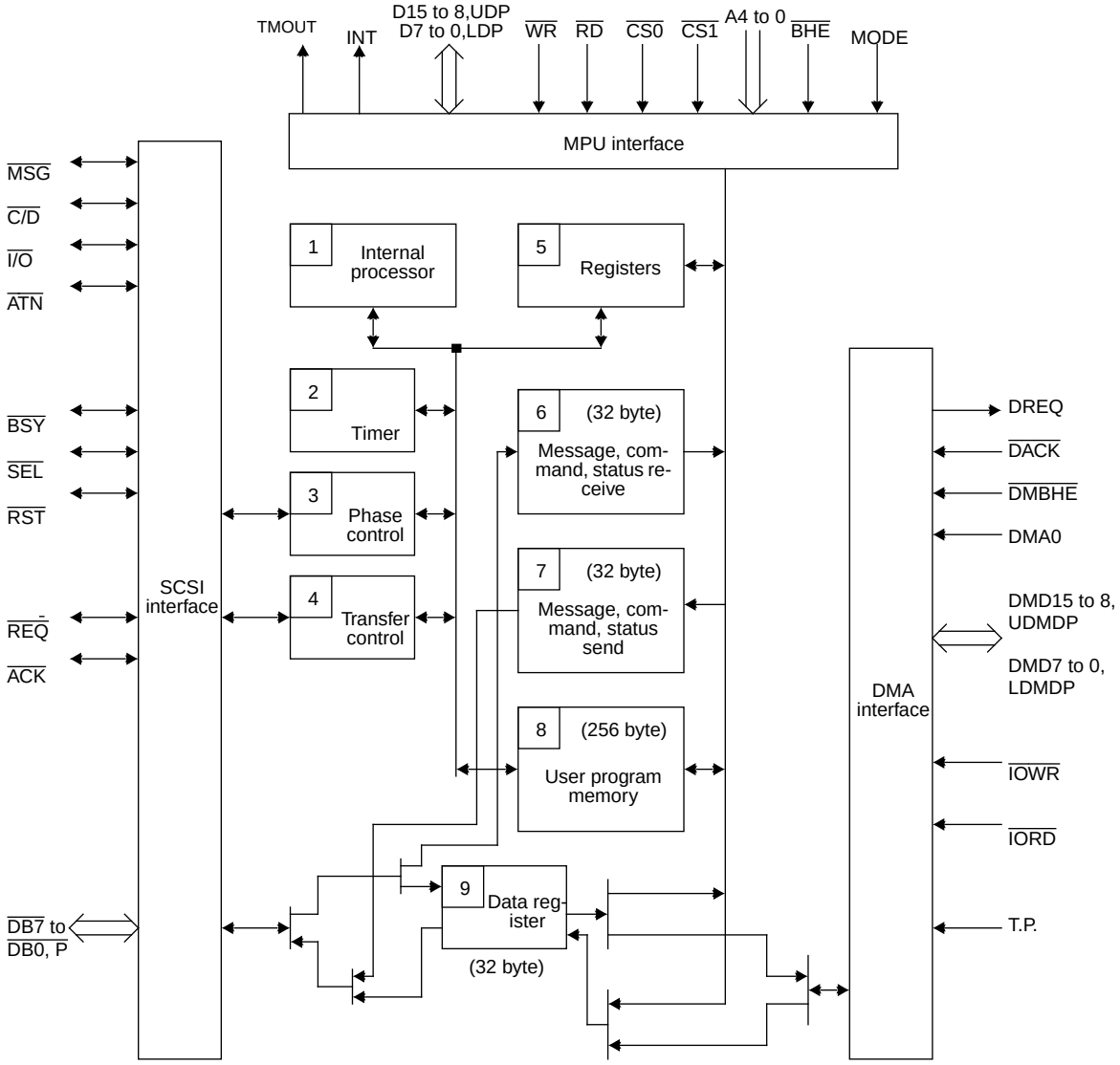


Figure 1.3 Block Diagram

CHAPTER 2 INPUT/OUTPUT PINS

This chapter describes the pin assignments for each SPC package and the functions of each pin.

2.1 MB86604A PIN ASSIGNMENT

Figure 2.1 shows the MB86604A pin assignment.

Also, Table 2.1 provides a list of MB86604A pins.

■ MB86604A Pin Assignment

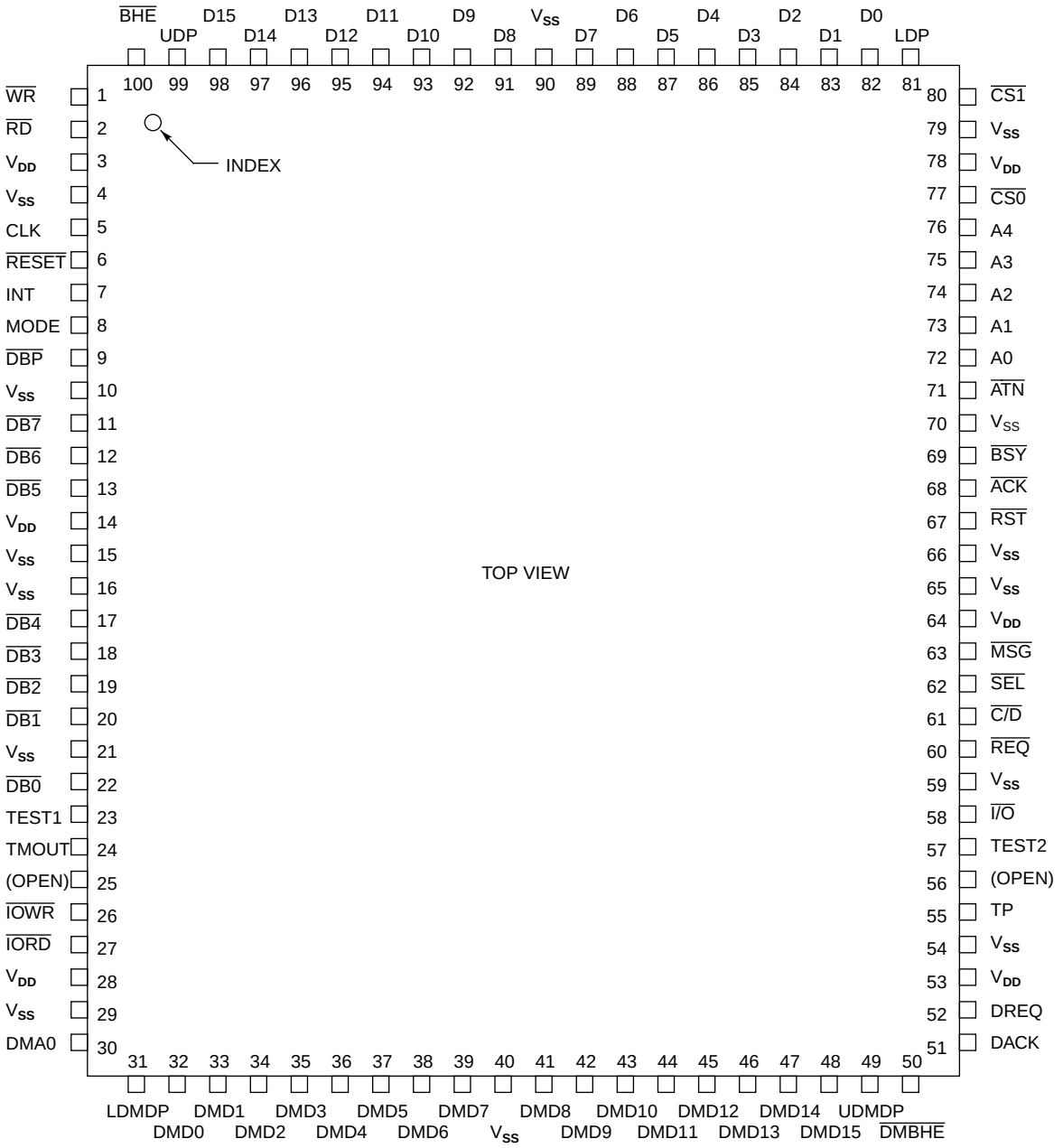


Figure 2.1 MB86604A Pin Assignment Diagram

Table 2.1 List of MB86604A Pins

No.	I/O	Pin name	No.	I/O	Pin name	No.	I/O	Pin name	No.	I/O	Pin name
1	I	$\overline{WR}$	26	I	$\overline{IOWR}$	51	I	$\overline{DACK}$	76	I	A4
2	I	$\overline{RD}$	27	I	$\overline{IORD}$	52	O	DREQ	77	I	$\overline{CS0}$
3	Power supply	V_{DD}	28	Power supply	V_{DD}	53	Power supply	V_{DD}	78	Power supply	V_{DD}
4	Ground	V_{SS}	29	Ground	V_{SS}	54	Ground	V_{SS}	79	Ground	V_{SS}
5	I	CLK	30	I	DMA0	55	I	TP	80	I	$\overline{CS1}$
6	I	$\overline{RESET}$	31	I/O	LDMDP	56		(OPEN)	81	I/O	LDP
7	O	INT	32	I/O	DMD0	57		TEST2	82	I/O	D0
8	I	MODE	33	I/O	DMD1	58	I/O	I/O	83	I/O	D1
9	I/O	$\overline{DBP}$	34	I/O	DMD2	59	Ground	V_{SS}	84	I/O	D2
10	Ground	V_{SS}	35	I/O	DMD3	60	I/O	$\overline{REQ}$	85	I/O	D3
11	I/O	$\overline{DB7}$	36	I/O	DMD4	61	I/O	$\overline{C/D}$	86	I/O	D4
12	I/O	$\overline{DB6}$	37	I/O	DMD5	62	I/O	$\overline{SEL}$	87	I/O	D5
13	I/O	$\overline{DB5}$	38	I/O	DMD6	63	I/O	$\overline{MSG}$	88	I/O	D6
14	Power supply	V_{DD}	39	I/O	DMD7	64	Power supply	V_{DD}	89	I/O	D7
15	Ground	V_{SS}	40	Groud	V_{SS}	65	Ground	V_{SS}	90	Groud	V_{SS}
16	Ground	V_{SS}	41	I/O	DMD8	66	Ground	V_{SS}	91	I/O	D8
17	I/O	$\overline{DB4}$	42	I/O	DMD9	67	I/O	$\overline{RST}$	92	I/O	D9
18	I/O	$\overline{DB3}$	43	I/O	DMD10	68	I/O	$\overline{ACK}$	93	I/O	D10
19	I/O	$\overline{DB2}$	44	I/O	DMD11	69	I/O	$\overline{BSY}$	94	I/O	D11
20	I/O	$\overline{DB1}$	45	I/O	DMD12	70	Ground	V_{SS}	95	I/O	D12
21	Power supply	V_{SS}	46	I/O	DMD13	71	I/O	$\overline{ATN}$	96	I/O	D13
22	I/O	$\overline{DB0}$	47	I/O	DMD14	72	I	A0	97	I/O	D14
23	I	TEST1	48	I/O	DMD15	73	I	A1	98	I/O	D15
24	O	TMOUT	49	I/O	UDMDP	74	I	A2	99	I/O	UDP
25	—	(OPEN)	50	I	$\overline{DMBHE}$	75	I	A3	100	I	$\overline{BHE}$

2.2 MB86604L PIN ASSIGNMENT

Figure 2.2 shows the MB86604L pin assignment.

Also, Table 2.2 provides a list of MB86604L pins.

■ MB86604L Pin Assignment

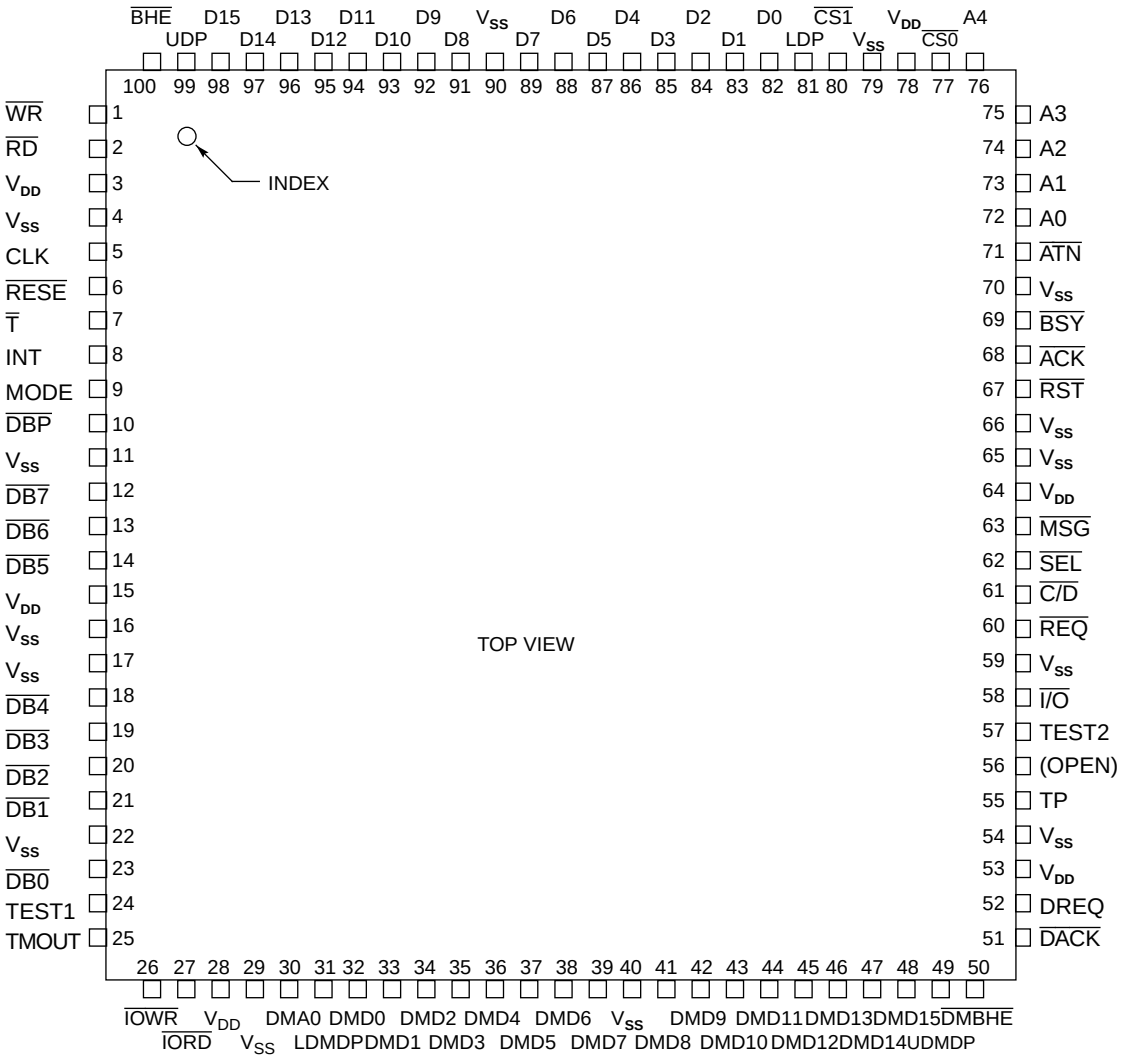


Figure 2.2 MB86604L Pin Assignment Diagram

Table 2.2 List of MB86604L Pins

No.	I/O	Pin name	No.	I/O	Pin name	No.	I/O	Pin name	No.	I/O	Pin name
1	I	$\overline{WR}$	26	I	$\overline{IOWR}$	51	I	$\overline{DACK}$	76	I	A4
2	I	$\overline{RD}$	27	I	$\overline{IORD}$	52	O	DREQ	77	I	$\overline{CS0}$
3	Power supply	V_{DD}	28	Power supply	V_{DD}	53	Power supply	V_{DD}	78	Power supply	V_{DD}
4	Power supply	V_{SS}	29	Power supply	V_{SS}	54	Power supply	V_{SS}	79	Power supply	V_{SS}
5	I	CLK	30	I	DMA0	55	I	TP	80	I	$\overline{CS1}$
6	I	$\overline{RESET}$	31	I/O	LDMDP	56	—	(OPEN)	81	I/O	LDP
7	O	INT	32	I/O	DMD0	57		TEST2	82	I/O	D0
8	I	MODE	33	I/O	DMD1	58	I/O	$\overline{I/O}$	83	I/O	D1
9	I/O	$\overline{DBP}$	34	I/O	DMD2	59	Power supply	V_{SS}	84	I/O	D2
10	Power supply	V_{SS}	35	I/O	DMD3	60	I/O	$\overline{REQ}$	85	I/O	D3
11	I/O	$\overline{DB7}$	36	I/O	DMD4	61	I/O	$\overline{C/D}$	86	I/O	D4
12	I/O	$\overline{DB6}$	37	I/O	DMD5	62	I/O	$\overline{SEL}$	87	I/O	D5
13	I/O	$\overline{DB5}$	38	I/O	DMD6	63	I/O	$\overline{MSG}$	88	I/O	D6
14	Power supply	V_{DD}	39	I/O	DMD7	64	Power supply	V_{DD}	89	I/O	D7
15	Power supply	V_{SS}	40	Power supply	V_{SS}	65	Power supply	V_{SS}	90	Power supply	V_{SS}
16	Power supply	V_{SS}	41	I/O	DMD8	66	Power supply	V_{SS}	91	I/O	D8
17	I/O	$\overline{DB4}$	42	I/O	DMD9	67	I/O	$\overline{RST}$	92	I/O	D9
18	I/O	$\overline{DB3}$	43	I/O	DMD10	68	I/O	$\overline{ACK}$	93	I/O	D10
19	I/O	$\overline{DB2}$	44	I/O	DMD11	69	I/O	$\overline{BSY}$	94	I/O	D11
20	I/O	$\overline{DB1}$	45	I/O	DMD12	70	Power supply	V_{SS}	95	I/O	D12
21	Power supply	V_{SS}	46	I/O	DMD13	71	I/O	$\overline{ATN}$	96	I/O	D13
22	I/O	$\overline{DB0}$	47	I/O	DMD14	72	I	A0	97	I/O	D14
23	I	TEST1	48	I/O	DMD15	73	I	A1	98	I/O	D15
24	O	TMOUT	49	I/O	UDMDP	74	I	A2	99	I/O	UDP
25	—	(OPEN)	50	I	$\overline{DMBHE}$	75	I	A3	100	I	$\overline{BHE}$

2.3 EXPLANATION OF PIN FUNCTIONS (SCSI INTERFACE)

Table 2.3 lists the SCSI interface pins and their functions.

■ SCSI Interface

Table 2.3 SCSI Interface Pins

Pin name	I/O	Function
$\overline{REQ}$, $\overline{ACK}$, $\overline{ATN}$, $\overline{MSG}$, $\overline{C/D}$, $\overline{I/O}$, $\overline{BSY}$, $\overline{SEL}$, $\overline{RST}$	I/O	SCSI control signal input/output Can be directly connected to single-ended SCSI connector. $\overline{REQ}$ and $\overline{ACK}$ Pins are the tri-state I/O port.
$\overline{DB7}$, $\overline{DB0}$, $\overline{DBP}$	I/O	SCSI data bus input/output Can be directly connected to single-ended SCSI connector.

Tri-state SCSI data bus pins:

You can select either tri-state or open-drain port for the $\overline{DB7}$ to $\overline{DB0}$, and $\overline{DBP}$ pins by a TEST1 pin input level.

Tri-state SCSI control pins:

You can select either tri-state or open-drain port for the $\overline{MSG}$, $\overline{C/D}$, $\overline{I/O}$, and $\overline{ATN}$ pins by a TEST2 pin input level.

2.4 EXPLANATION OF PIN FUNCTIONS (MPU INTERFACE)

Table 2.4 lists the MPU interface pins and their functions.
Depending on the mode setting, the MPU interface can be connected to the input/output signals of 80 series or 68 series MPU.

■ MPU Interface

Table 2.4 MPU Interface Pins

Pin name	I/O	Function
$\overline{CS0}$	I	Input signal for MPU to select SPC as I/O device.
$\overline{CS1}$	I	Input selection signal (external circuit selection signal) when MPU inputs/outputs DMA bus data via the SPC.
D15 to D8, UDP *1	I/O	Data bus upper byte and parity $\overline{CS0}$ input valid: SPC internal register input/output port $\overline{CS1}$ input valid: DMA bus data input/output port
D7 to D0, LDP	I/O	Data bus lower byte and parity $\overline{CS0}$ input valid: SPC internal register input/output port $\overline{CS1}$ input valid: DMA bus data input/output port
A4 to A0	I	Input address to select internal register.
$\overline{RD}$ (R/$\overline{W}$)	I	80 series mode: Input $\overline{IORD}$ signal or $\overline{RD}$ signal for MPU to read from SPC 68 series mode: Input control signal R/W for MPU to read from/write to SPC
$\overline{WR}$ ($\overline{LDS}$)	I	80 series mode: Input $\overline{IOWR}$ signal or $\overline{WR}$ signal for MPU to read from SPC 68 series mode: Input $\overline{LDS}$ signal output from MPU when data bus lower byte is valid
$\overline{BHE}$ ($\overline{UDS}$) *2	I	80 series mode: Input $\overline{BHE}$ signal output from MPU when data bus upper byte is valid 68 series mode: Input $\overline{UDS}$ signal output from MPU when data bus upper byte ils valid
INT	O	Interrupt request signal output. 80 series mode: Hactive 68 series mode: L active For this INT signal output, please see the INT signal hold mode in section 3.3.13
MODE	I	Input signal which specifies MPU (80 series) 80 series mode: Input 1 68 series mode: Input 0

*1: when 8-bit bus width, pull up the UDP and D15 to D8 pins with approx. 10 kΩ.
*2: when 8-bit bus width, fix the $\overline{BHE}$ pin to "1".

2.5 EXPLANATION OF PIN FUNCTIONS (DMA INTERFACE)

Table 2.5 lists DMA interface pins and their functions.

■ DMA Interface

Table 2.5 DMA Interface Pins

Pin name	I/O	Function
DREQ	O	Output DMA transfer request signal for DMAC. Request DMA data transfers between SPC and memory.
$\overline{\text{DACK}}$	I	Input DMA transfer permissible signal from DMAC. When DMA transfer permissible signal is active, DMA transfer (read/write) is executed.
D15 to D8, UDP *1	I/O	DMA data bus upper byte and parity CS1 input valid: MPU data bus is connected.
DMD7 to D0, LDMDP	I/O	DMA data bus lower byte and parity CS1 input valid: MPU data bus is connected.
$\overline{\text{IORD}}$ (DMR/$\overline{\text{W}}$)	I	80 series mode: Input $\overline{\text{IORD}}$ dignal or $\overline{\text{RD}}$ signal to output data from SPC to DMA bus 68 series mode: Input control segnal R/$\overline{\text{W}}$ for DMAC to input data to or output data from SPC
$\overline{\text{IOWR}}$ ($\overline{\text{DMLDS}}$)	I	80 series mode: Input $\overline{\text{IOWR}}$ signal or $\overline{\text{WR}}$ signal to input data on the DMA bus to SPC 68 series mode: Input control signal R/W for MPU to read from/ write to SPC
$\overline{\text{DMBHE}}$ ($\overline{\text{DMUDS}}$)	I	80 series mode: Input $\overline{\text{BHE}}$ signal output from DMAC when DMA data bus upper byte is valid 68 series mode: Input $\overline{\text{UDS}}$ signal output from DMAC when DMA data bus upper byte ils valid Connect to V_{DD} in 8-bit mode.
DMA0	I	Input address data A0 signal output from DMAC in 80 series mode. 68 series mode: Connect power source (V_{DD}) Connect to V_{SS} or V_{DD} in 8-bit mode.
TP (Transfer Permission)*2	I	Input DMA transfer permissible signal. When TP signal is active, SPC performs DMA transfer. If TP signal becomes inactive during a DMA transfer, the transfer is suspended at the block boundary.

*1: When DMA bus is used in 8-bit, pull up those pins with approx. 10 kΩ.
*2: See 6.1 for an explanation of TP signal control
Note: When DMA bus is used in 8-bit mode, use DMD7-0, P or $\overline{\text{IOWR}}$ ($\overline{\text{DMLDS}}$). $\overline{\text{DMUDS}}$ side cannot be used even in 68 series mode.

2.6 EXPLANATION OF PIN FUNCTIONS (CONTROL SIGNAL)

Table 2.6 lists control signal pins and their functions.

■ **Control Signals**

Table 2.6 Control Signal Pins

Pin name	I/O	Function
$\overline{\text{RESET}}$	I	System reset input. Set at least 4 CLK active.
CLK	I	Clock input. (20, 30, 40 MHz)
TEST1	I	1) Connect to Vss if $\overline{\text{DBP}}$, $\overline{\text{DB7}}$ to $\overline{\text{DB0}}$ are used as open-drain I/O port. 2) Connect to V_{DD} if $\overline{\text{DBP}}$, $\overline{\text{DB7}}$ to $\overline{\text{DB0}}$ are used as tri-state I/O port.
TEST2	I	1) Connect to Vss if $\overline{\text{MSG}}$, $\overline{\text{C/D}}$, $\overline{\text{I/O}}$, and $\overline{\text{ATN}}$ are used as open-drain I/O port. 2) Connect to V_{DD} if $\overline{\text{MSG}}$, $\overline{\text{C/D}}$, $\overline{\text{I/O}}$, and $\overline{\text{ATN}}$ are used as tri-state I/O port.
V_{DD}	—	Power supply
Vss	—	Ground
TMOUT	O	SPC timeout output. When SPC Busy flag=1 for longer time than one specified, this pin outputs "H".
(OPEN)	—	Leave as open pin with no connection.

CHAPTER 3 FUNCTIONS

This chapter describes the functions of SPC internal registers, memory, and buffers.

3.1 INTERNAL REGISTERS

Indicates the relationship between signal input to access SPC internal registers and accessed registers.
Depending on the type of information, the SPC register where the MPU reads/writes differs (data register or MCS buffer).

■ Internal Register Access

Tables 3.1a and 3.1b show the access state to internal registers for 80 series and 68 series modes.

Table 3.1a Internal Register Access (80 Series Mode)

$\overline{CS0}$	$\overline{BHE}$*1	A0	D15 to D8*2	D7 to D0
1	—	—	HI-Z	HI-Z
0	0	0	Odd address register	Even address register
0	0	1	Odd address register	HI-Z
0	1	0	HI-Z	Even address register
0	1	1	HI-Z	Odd address register

*1 Set $\overline{BHE}$ pin to 1 for 8-bit bus operation.
*2 Connect about 10-kΩ pull-up resistor to the UDP and D15 to D8 when using 8-bit bus..

Table 3.1b Internal Register Access (68 Series Mode)

$\overline{CS0}$	$\overline{UDS}$*2	LDS	A0*1	D15 to D8*3	D7 to D0
1	—	—	—	HI-Z	HI-Z
0	0	0	1	Even address register	Odd address register
0	0	1	1	Even address register	HI-Z
0	1	0	0	HI-Z	Even address register
0	1	0	1	HI-Z	Odd address register
0	1	1	—	HI-Z	HI-Z

*1 Set A0 pin to 1 for 16-bit bus operation.
*2 Set $\overline{UDS}$ pin to 1 for 8-bit bus operation.
*3 Connect about 10-kΩ pull-up resistor to the UDP and D15 to D8 when using 8-bit bus.

■ Data Flow During Transfers

Depending on the type of information, the SPC register where the MPU reads/writes differs (data register or MCS buffer). There are also program transfers and DMA transfers. For DMA transfers, data phases are supported, but messages, commands, and status phases are not.

Table 3.1c shows the relationship between information types and registers (program transfer/ data transfer). Figure 3.1 outlines the internal flow of information.

Table 3.1c Relationship between Information Types and Registers

Transfer / Information	Program transfer (via MPU bus)	DMA transfer (via DMA bus)
Message phase	MCS buffer (32 byte input/output)	Not possible
Command phase		
Status phase		
Data phase	Data register	Data register

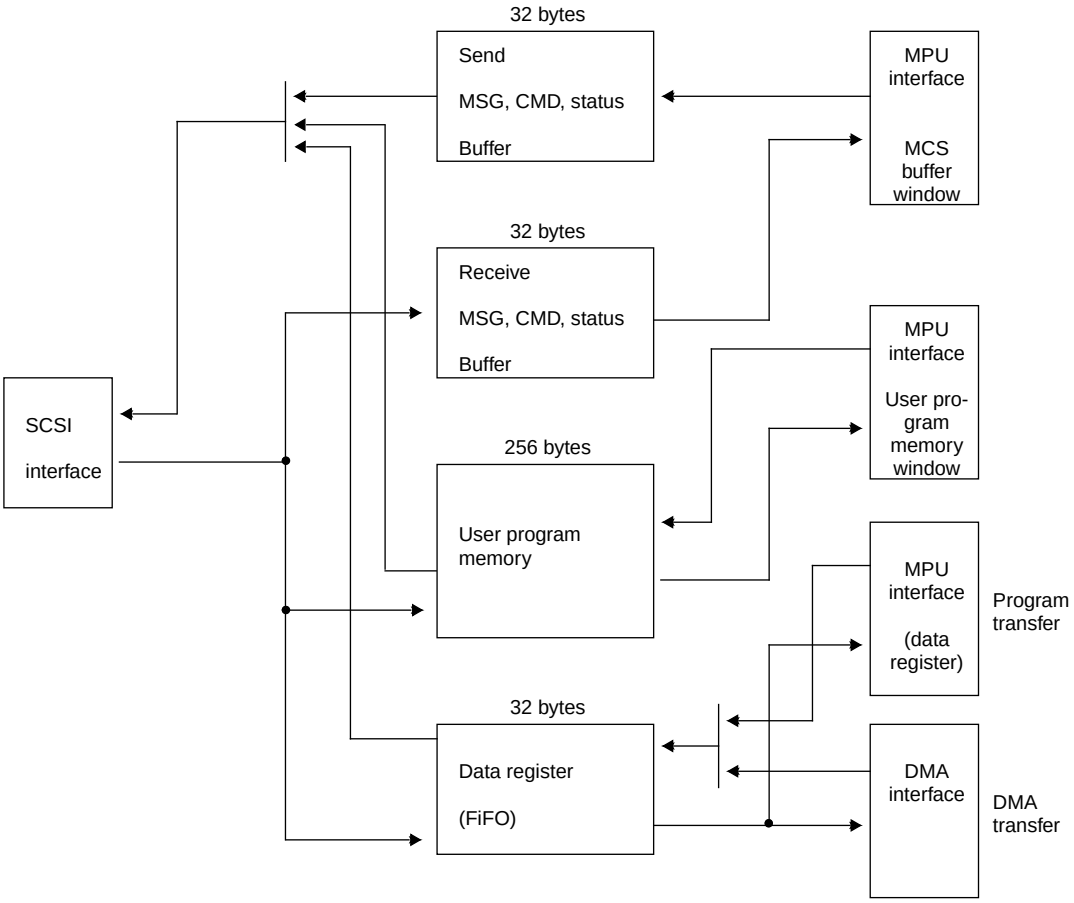
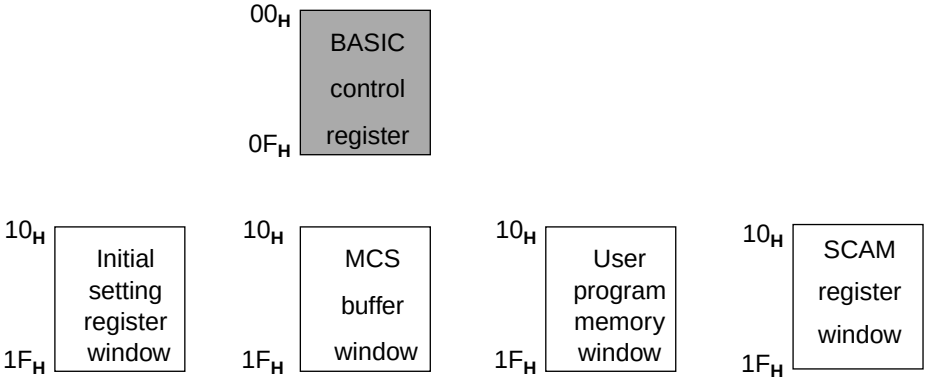


Figure 3.1 Internal Flow of Information

3.2 BASIC CONTROL REGISTER

This register controls the SPC and confirms SPC states.



Note: SCAM register window is available in MB86604L only.

■ BASIC Control Register

Table 3.2 lists the BASIC control register.

Table 3.2 BASIC Control Register List

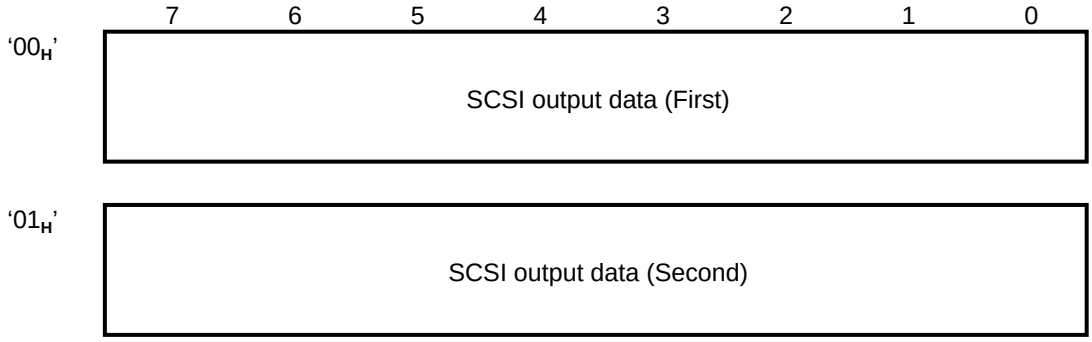
No	Address					Write	Read
	A4	A3	A2	A1	A0		
00	0	0	0	0	0	SCSI output data register (First)	SCSI input data register (First)
01	0	0	0	0	1	SCSI output data register (Second)	SCSI input data register (Second)
02	0	0	0	1	0	Direct control register	SPC status register
03	0	0	0	1	1	(RESERVED)	Nexus status register
04	0	0	1	0	0	SEL/RESEL-ID register	Interrupt status register
05	0	0	1	0	1	Command register	Command step register
06	0	0	1	1	0	Data block register (MSB)	←
07	0	0	1	1	1	Data block register (LSB)	←
08	0	1	0	0	0	Data byte register (MSB)	←
09	0	1	0	0	1	Data byte register	←
0A	0	1	0	1	0	Data byte register (LSB)/MC byte register	←
0B	0	1	0	1	1	Diagnostic control signal register	SCSI control signal status register
0C	0	1	1	0	0	Transfer mode register	←
0D	0	1	1	0	1	Transfer period register	←
0E	0	1	1	1	0	Transfer offset register	←
0F	0	1	1	1	1	Window address register	Modified byte register

3.2.1 Output Data Register (Write)

In the data phase program transfer mode, the output data register outputs data to the SCSI bus.

■ **Output Data Register (Write)**

The output data register configuration is shown below.

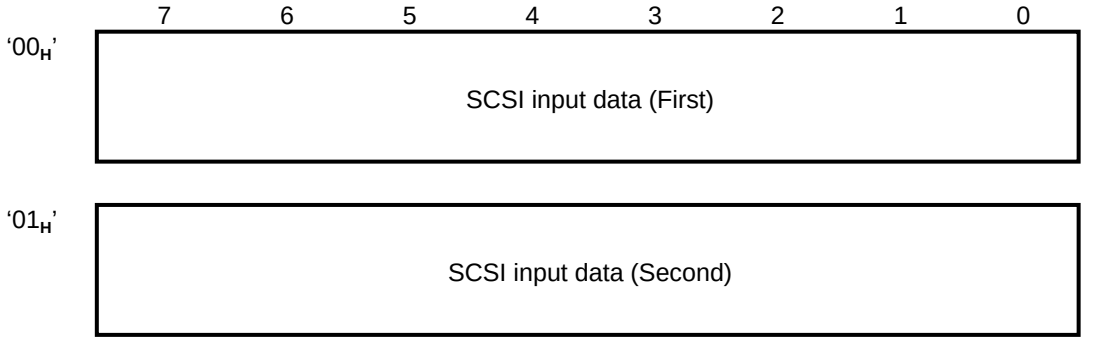


3.2.2 Input Data Register (Read)

In the data phase program transfer mode, the input data register inputs data to the SCSI bus.

■ **Input Data Register (Read)**

The input data register configuration is shown below.



MEMO

3.2.3 Direct Control Register (WRITE)

The direct control register (WRITE) provides direct control regardless of whether a command is issued to the SPC.

■ Direct Control Register (WRITE)

The direct control register bit configuration is shown below.

	7	6	5	4	3	2	1	0
'02_H'	DC 7	0	0	DC 4	0	0	0	0

Next, the functionality of each bit is described.

BIT 7: ATN signal control
Control bit for direct control of ATN signal assert.
When the SPC operates as an initiator, the ATN signal is asserted by writing 1 in BIT 7. Effective for interrupting a data phase (produces attention condition).

After writing 1 in BIT 7 and asserting the ATN signal, be sure to write 0 and return to the initial state before issuing the next command. The ATN signal is not negated by writing 0. In order to negate the ATN signal, issue a RESET ATN signal.

In Figure 3.2.3, the relationship between transfers and BIT 7 is diagrammed in a flowchart.

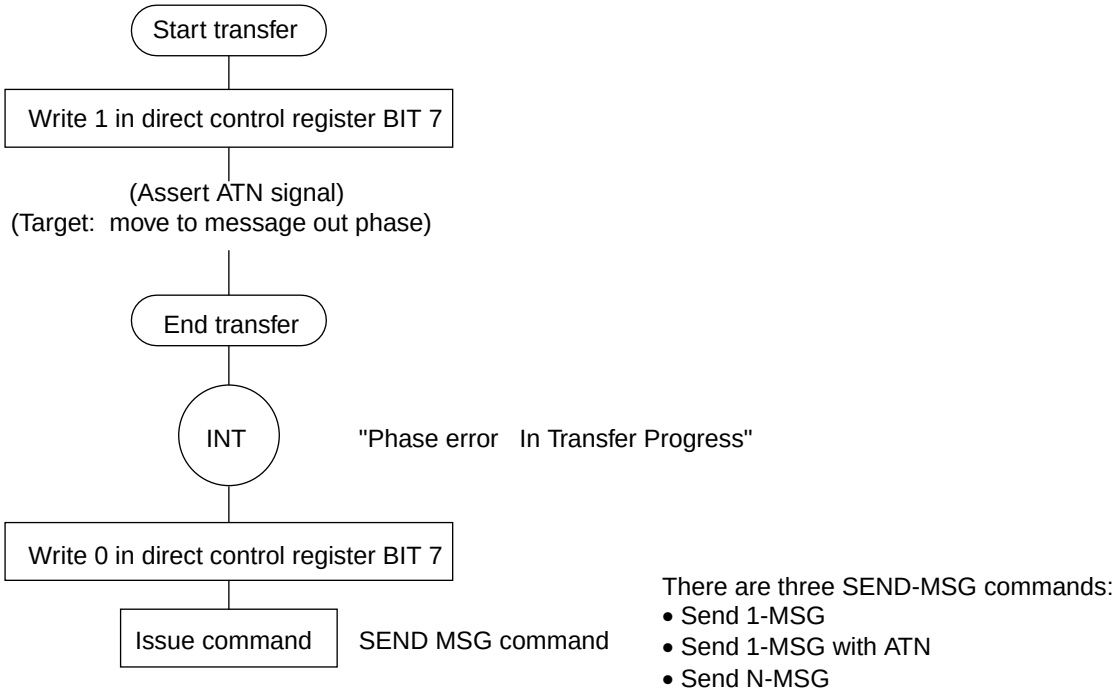


Figure 3.2.3 Transfer Flowchart (Direct Control Register: BIT 7)

BIT 4: TMOUT signal clear

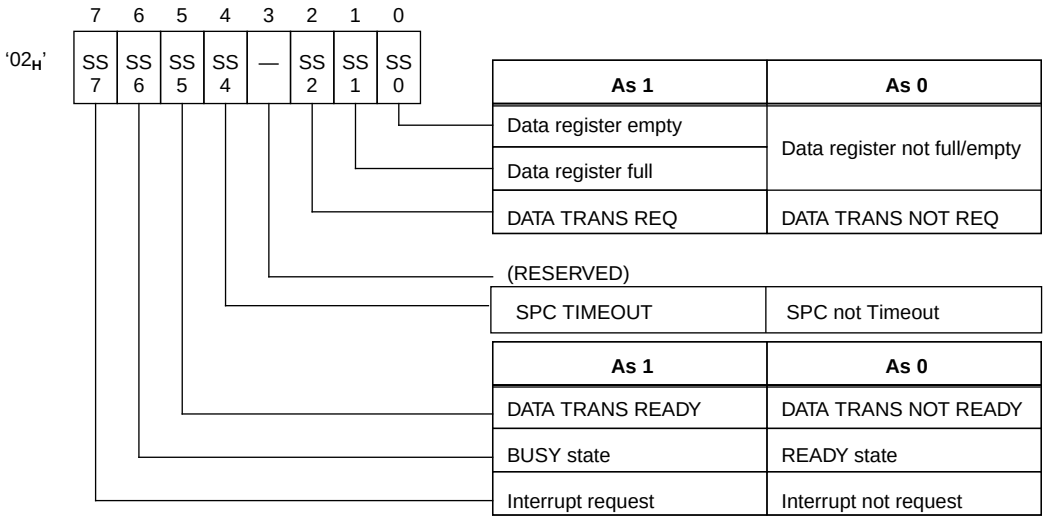
Control bit to clear the TMOUT pin signal.
The TMOUT pin is cleared by writing "1" to this bit while the SPC Busy bit (Bit 6) in the SPC Status Register (02h) = "0". When SPC Busy = "1", the TMOUT pin is not cleared.

3.2.4 SPC Status Register (READ)

The SPC status register shows the operating state of the SPC.

■ **SPC Status Register (READ)**

The SPC status register (READ) bit configuration is shown below.



Next, the functionality of each bit is described.

BIT 7: Interrupt request
When there is an interrupt request from the SPC to the host MPU, 1 is indicated. While linked to the INT signal (interrupt request signal) to the host MPU, this bit indicates 1 whenever there is an interrupt request regardless the interrupt is enabled or disabled (interrupt enable register).

BIT 6: BUSY state
Shows that the SPC is operating.
When the SPC receives a command or automatically begins operating (at Automatic selection/reselection response mode), 1 is indicated.
When the operation is successfully completed or ends unsuccessfully because of an error, 0 is displayed.
If a command is issued when BIT 6 is 1, except for SOFTWARE RESET, the command is ignored and a COMMAND REJECTED interrupt reported.

BIT 5: DATA TRANS READY state
Shows that a transfer is possible (during transfer).
When the SPC receives a transfer-related command and the SPC internal set-up finishes, 1 is displayed in this bit.

BIT 4: SPC Timeout State
Shows that SPC is operating over the time specified in the SPC timeout Setting register. When the SPC's operation terminated within the time specified, this bit value is "0". This bit can be cleared by writing "1" to the TMOUT pin clear bit (Bit 4) of Direct Control Register while SPC busy=0.

BIT 2: DATA TRANS REQ
Shows state of data phase transfer request.
During SCSI-INPUT, when two or more bytes can be read from the data register (the data register contains two or more bytes) or when the data register contains the last byte, 1 is indicated.
During SCSI-OUTPUT, when two or more bytes can be written in the data register (the data register contains 30 or less bytes) or the last byte can be written in the data register, 1 is indicated. Refer to this bit when conducting a program transfer.
Also, when DMA transfer mode, "1" will be indicated in this bit while the DREQ signal is asserted.

BIT 1: Data Register FULL
When the data register is FULL, 1 is indicated.

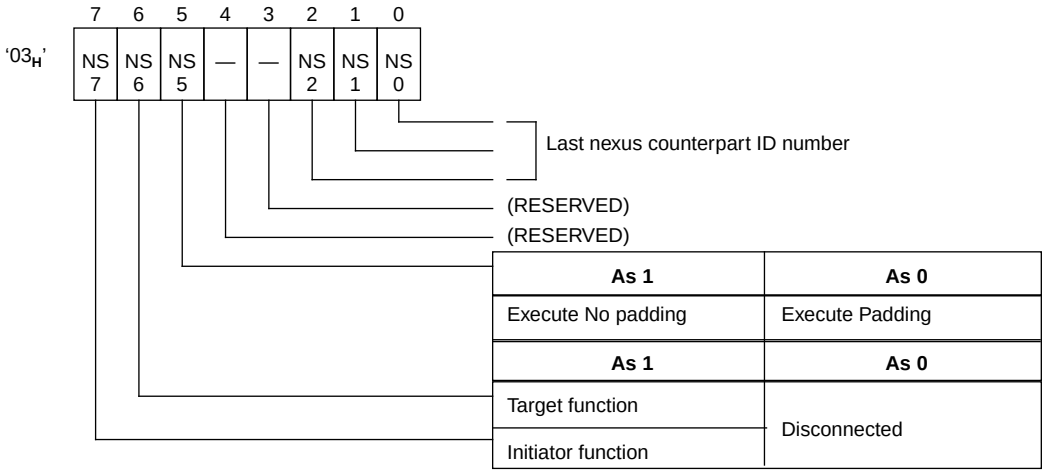
BIT 0: Data Register EMPTY
When the data register is EMPTY, 1 is indicated.

3.2.5 Nexus Status Register (READ)

The nexus status register (READ) shows the SPC internal state and the nexus counterpart ID.

■ Nexus Status Register (READ)

The nexus status register (READ) bit configuration is shown below.



Next, the functionality of each bit is described.

BIT 7: Initiator function

When the SPC starts as the initiator, 1 is indicated. At the following points, 1 is indicated.

- Reselection by the target (SEL and I/O signals are true when the self ID bit is true).
- After acquiring bus usage rights as the initiator and asserting the SEL signal, when the ID bit is sent to the data bus during the selection phase or when the ATN signal is asserted simultaneously.

When the SPC is disconnected by the target and I-T nexus is released, 0 is indicated.

BIT 6: Target function

When the SPC starts as the target, 1 is indicated. At the following points, 1 is indicated.

- Selection by the initiator (SEL signal is true and I/O signal is false when the self ID bit is true).
- After acquiring bus usage rights as the target and asserting the SEL signal, when the ID bit is sent to the data bus during the reselection phase or when the I/O signal is asserted simultaneously.

When the SPC is disconnected and I-T nexus is released, 0 is indicated.

BIT 5: Padding operation

1 is indicated when the SPC receives a data transfer command with padding and then it did not perform the padding transfer after transferring the specified byte data. When the padding transfer is performed, 0 is indicated. This bit does not change if a data transfer command without padding is performed.

Also, please read out the value for this bit until the disconnect is occurred, since this bit is cleared when the device is newly nexused.

Furthermore, this bit can not be referred by the user program.

BIT 2-0: Final nexus counterpart ID

Stores nexus counterpart ID. BIT 2-0 are set at the following points.

- During the initiator function in the selection phase, when a BSY signal is received from the counterpart target.
- During the target function in the reselection phase, when a BSY signal is received from the counterpart initiator.
- During the intiator function, when a BSY signal is asserted in response to a target reselect request.
- During the target function, when a BSY signal is asserted in response to an initiator select request.

Data is stored in those bits even when the SPC is disconnected. This data is rewritten when a new nexus is established.

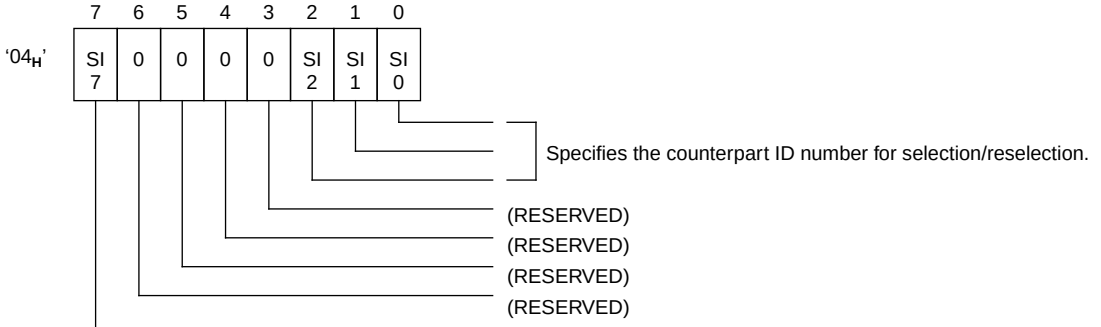
When the SPC is the target in the Automatic selection response mode, it operates automatically up to the command phase. At which time, the host MPU can reference this register to find out the initiator ID number. Also, when the SPC is the initiator in the Automatic reselection response mode, it operates automatically up to the message in phase. At which time, the host MPU can reference this register to find out the target ID number. (See 3.3.4 for more about the Automatic selection/reselection response mode).

3.2.6 SEL/RESEL ID Register (WRITE)

The SEL/RESEL ID register (WRITE) specifies DMA transfer misalignment processing and the selected counterpart bus device ID.

■ SEL/RESEL ID Register (WRITE)

The SEL/RESEL ID register (WRITE) bit configuration is shown below.



As 1	As 0
Perform DMA misalignment	Do not perform DMA misalignment

Next, the functionality of each bit is described.

BIT 7: DMA transfer misalignment

Please set when the DMA transfer width is 16 bits, the first MPU/DMA side read/write is done by byte access, and writing to the system memory or reading from the system memory is performed to/from an odd address.

1:Writing to the system memory or reading from the system memory can be performed to/from an odd address.

0:Writing to the system memory or reading from the system memory can be performed to/from an even address.
See 6.2 for more details.

BIT 2-0: Setting SEL-RESEL ID
Specify the counterpart bus device ID number for selection/reselection with a binary digit.

■ Setting Transfer Mode/Transfer Parameters

When setting the transfer mode/transfer parameters (PERIOD, OFFSET), after specifying the counterpart ID number in the SEL/RESEL ID register, set the following.

- Transfer mode register (address $0C_H$)
- Transfer period register (address $0D_H$)
- Transfer offset register (address $0E_H$)

For a target in the selection phase to which only the target ID bit is sent (single initiator), set the transfer parameters with the initiator ID number as 0.

3.2.7 Interrupt Status Register (READ)

The interrupt status register (READ) displays the reason for the interrupt using an 8-bit code.

■ Interrupt Status Register (READ)

The interrupt status register (READ) bit configuration is shown below.

	7	6	5	4	3	2	1	0
'04_H'	IS 7	IS 6	IS 5	IS 4	IS 3	IS 2	IS 1	IS 0

The interrupt status register, an FIFO-type register (8 bytes), stores interrupt codes which occur until the command is completed and can store two or more interrupt codes. Read this register when BIT 7 (interrupt request exists) of the SPC status register (address 02_H) is 1.

After each host MPU read operation, this register indicates the next interrupt code.

The interrupt status register value occurs at the step indicated by the command step register (address 05_H). Therefore, read this register and the command step register at the same time.

With an 8-bit MPU, read the command step register after reading the interrupt status register (unless this is done, the next interrupt code will not be shown).

See 5.7 for more on interrupt codes.

Notes on Reading Interrupt Status Register

1) If Bit 7 of automatic operation mode setting register (address 1Ch) is set to "1", all the interrupt code should be read out from this register. Otherwise, any commands issued later will be ignored.

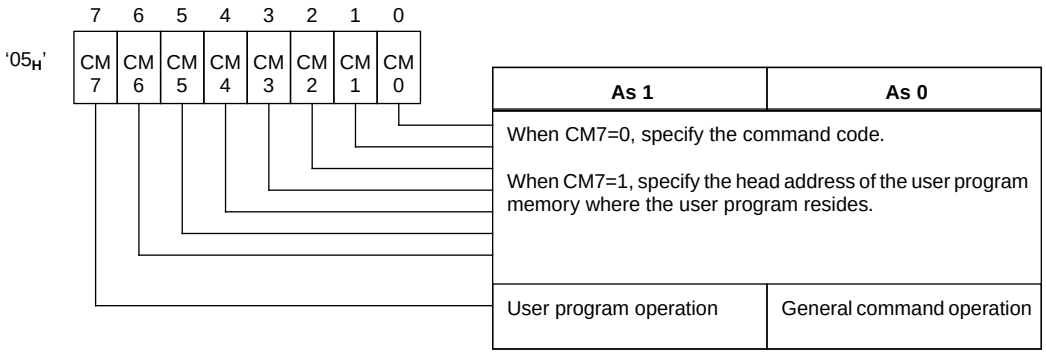
2) Do not access this register until SPC Busy bit (Bit 6 of SPC Status register) becomes "0" after the command was issued. Also, please read out the interrupt codes before issuing new command.

3.2.8 Command Register (WRITE)

The command register (WRITE) issues commands for the SPC.

■ Command Register (WRITE)

The command register (WRITE) bit configuration is shown below.



Next, the functionality of each bit is described.

BIT 7: User program operation

This bit indicates whether the SPC operation will be conducted by the user program or by command. The meaning of BIT 6-0 changes depending on this bit setting.

0:Write the command code in BIT 6-0 (see chapter 4 for more on command codes).

1:Write the head address of the user program in BIT 6-0 (Figure 3.2.8). The SPC starts executing from the command at the address specified in the user program memory written in this bit. The head address of the user program can be specified in 2-byte units (only even address).

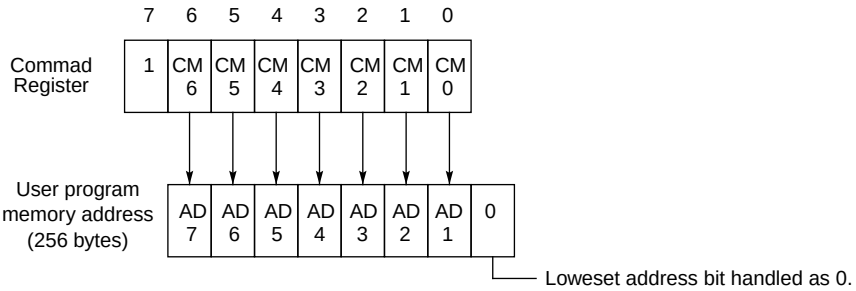


Figure 3.2.8 Relationship between Command Register (BIT 7 = 1) and User Program Memory

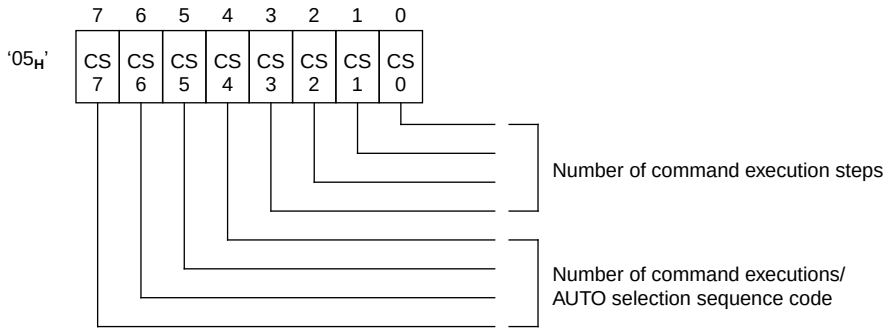
3.2.9 Command Step Register (READ)

The command step register (READ) indicates the number of command execution steps and program steps in the user program.

An FIFO-type register (8 bytes), it stores command steps corresponding to the interrupt status register.

■ **Command Step Register (READ)**

The command step register (READ) bit configuration is shown below.



Next, the functionality of each bit is described.

BIT 7-4: Number of user program command executions/Automatic selection response mode sequence code

During user program operation: Ring counter which indicates the number of command executions. Advanced with each command execution start.
Only counts discrete commands. Special commands are not counted (see 4.1 for more on discrete commands and special commands).

During Automatic selection response mode (target only) operation: Indicates the type of response operation sequence (see 3.3.4 for more on the Automatic selection response mode).

BIT 3-0:Number of command execution steps

Indicates the number of command execution steps. Step codes are defined for each command (see chapter 4 for more on command steps). Same when set to the Automatic selection/reselection response mode. In the Automatic receive mode, information is received but this bit is not affected.

(Example) When an ”Initial phase error & MSG-receive” interrupt occurs, the execution step at which the phase error occurred is indicated.

Notes on Reading Command Step Register

1) The contents of interrupt status register (address 04h) are the interrupt codes generated at the step indicated in this command step register. Therefore, it is necessary to read out this register along with the interrupt status register.

When 8-bit MPU is used, be sure always read out this step register after reading out the interrupt status register. If not, the next interrupt status will not be indicated. Also, if this register is read out before reading out the interrupt status register, the correct interrupt status will not be read out.

2) Please do not access this step register until Bit 6 (SPC Busy bit) of SPC Status register becomes "0" after issuing the command. If a new command is issued, please read out the existing step codes in advance.

3.2.10 Data Block Register (READ/WRITE)

The data block register (READ/WRITE) specifies the number of blocks for data phase transfers.

■ Data Block Register (READ/WRITE)

The data block register (READ/WRITE) bit configuration is shown below.

	7	6	5	4	3	2	1	0
'06_H'	BL 15	BL 14	BL 13	BL 12	BL 11	BL 10	BL 9	BL 8
'07_H'	BL 7	BL 6	BL 5	BL 4	BL 3	BL 2	BL 1	BL 0

Specify the number of blocks to be transferred in the data phase in the data block register bits.

Fixed length: Specify the number of transfer blocks.

Variable length: Enter 0001_H.

This register's read values become valid after the completion report (including abnormal completions),
when the data block register and byte register are both 00 following a normal completion or when the number of untransferred blocks and bytes are reported following an abnormal completion.

(Example)

Set values: Blocks at $000A_H$ and byte length at 000100_H.

Normal: Blocks at 0000_H and byte length at 000000_H.

Abnormal 1: Blocks at 0004_H and byte length at $000AC_H$.

In this case, the untransferred data is four blocks and AC_H bytes. Meanwhile, the transferred data is five blocks and 54_H bytes.

Abnormal 2: Blocks at 0004_H and bytes at 0000_H.

In this case, the untransferred data is four blocks, and the transferred data is six blocks.

Abnormal 3: Blocks at 0005_H and byte length at 0100_H.

In this case, the untransferred data is six blocks (five blocks and 100_H bytes), and the transferred data is four blocks.

■ Data Block Register Access

The data block register becomes valid after the completion report (including abnormal completions). Prior to the completion report, initial values are indicated.

Please specify (set) this register when SPC Busy bit (Bit 6 of SPC status register) = "0".

MEMO

3.2.11 Data Byte/MC Byte Register (READ/WRITE)

The data byte/MC byte register (READ/WRITE) specifies the number of transfer bytes for data phase transfers and message phase/ command phase transfers.

■ Data Byte/MC Byte Register (READ/WRITE)

The data byte/MC byte register (READ/WRITE) bit configuration is shown below.

	7	6	5	4	3	2	1	0
'08_H'	BY 23	BY 22	BY 21	BY 20	BY 19	BY 18	BY 17	BY 16
'09_H'	BY 15	BY 14	BY 13	BY 12	BY 11	BY 10	BY 9	BY 8
'$0A_H$'	BY 7	BY 6	BY 5	BY 4	BY 3	BY 2	BY 1	BY 0

- Data byte register (BY23-BY0)

  Specify the number of bytes for data phase transfers.

  Fixed length: Specify the length of a single block.
  Variable length: Specify the number of transfer bytes.

- MC byte register (BY7-BY0)

  Specify the number of bytes for message phase and command phase transfers.
  Only specify the number of transfer bytes in this register when the command being issued requires that the number of transfer bytes be set.

This register's read values become valid after the completion report (including abnormal completions),
when the data block register and byte register are both 00 following a normal completion or when the number of untransferred blocks and bytes are reported following an abnormal completion.

(Example)

Set values: Blocks at $000A_H$ and byte length at 000100_H.

Normal: Blocks at 0000_H and byte length at 000000_H.

Abnormal 1: Blocks at 0004_H and byte length at $000AC_H$.

In this case, the untransferred data is four blocks and AC_H bytes. Meanwhile, the transferred data is five blocks and 54_H bytes.

Abnormal 2: Blocks at 0004_H and byte at 0000_H.

In this case, the untransferred data is four blocks, and the transferred data is six blocks.

Abnormal 3: Blocks at 0005_H and byte length at 0100_H.

In this case, the untransferred data is six blocks (five blocks and 100_H bytes), and the transferred data is four blocks.

■ Data Byte/MC Byte Register Access

The data byte/MC byte register becomes valid after the completion report (including abnormal completions). Prior to the completion report, initial values are indicated.

However, when used as the MC byte register, initial values are always shown.

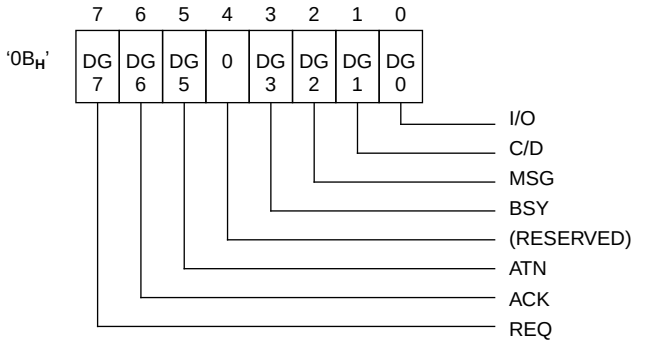
Please specify (set) this register when SPC Busy bit (Bit 6 of SPC status register) = "0".

3.2.12 DIAGNOSTIC Control Register (WRITE)

The DIAGNOSTIC control register (WRITE) emulates the SCSI signal sequence during the Diag mode.

■ DIAGNOSTIC Control Register (WRITE)

The DIAGNOSTIC control register (WRITE) bit configuration is shown below.



This register becomes valid from issuance of **INIT DIAG START** or **TARG DIAG START** command to the **DIAG END** command.

When 1 is written in the bits corresponding to the control signals, this signal is asserted.

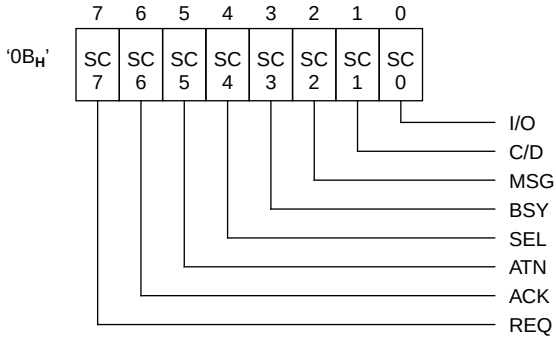
Note:Write "00_H" to this register prior to the issue of **INIT DIAG START** or **TARG DIAG START** command.

3.2.13 SCSI Control Signal Status Register (READ)

The SCSI control signal status register (READ) shows the state of the SCSI control signals.
Valid in Diag mode as well.

■ SCSI Control Signal Status Register (READ)

The SCSI control signal status register (READ) bit configuration is shown below.



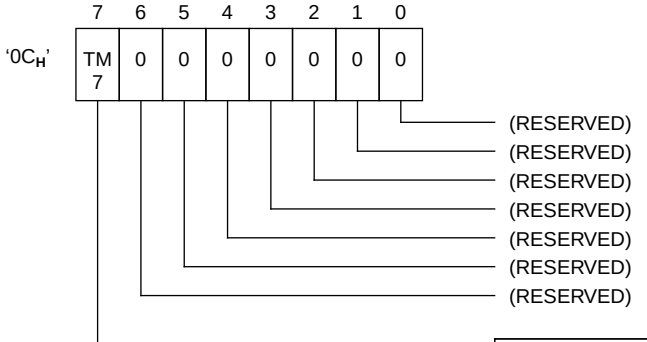
When the control signal corresponding to these bits is asserted, 1 is set.

3.2.14 Transfer Mode Register (WRITE/READ)

The transfer mode register (WRITE/READ) specifies the data phase transfer mode (synchronous transfer/asynchronous transfer).

■ Transfer Mode Register (WRITE/READ)

The transfer mode register (WRITE/READ) bit configuration is shown below.



Item	As 1	As 0
Transfer mode (data phase)	Synchronous	Asynchronous

Next, the functionality of each bit is described.

BIT 7: Transfer mode

Specifies the data phase transfer mode for the bus device ID listed in the SEL/RESEL ID register (address 04_H).

1: Data transfer in synchronous mode.
0: Data transfer in asynchronous mode.

■ Setting Transfer Mode Register

This register can store the transfer mode for each connected device. So, please set this register after the counterpart's ID to perform the data transfer is set to the SEL/RESEL ID register (address 04h). Once setting this register, the transfer mode will be automatically set to this register just only setting the SEL/RESEL ID register later, until a system reset or software reset is performed.

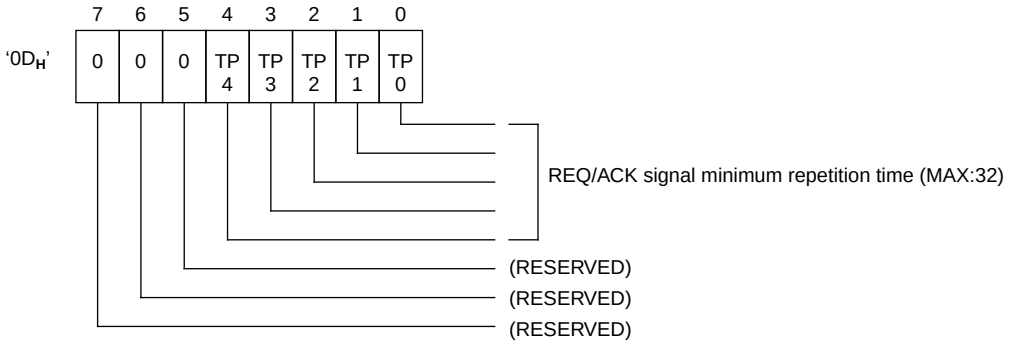
MEMO

3.2.15 Transfer Period Register (READ/WRITE)

The transfer period register (READ/WRITE) specifies the synchronous transfer period for data phase transfers.

■ Transfer Period Register (READ/WRITE)

The transfer period register (READ/WRITE) bit configuration is shown below.



Set the synchronous transfer period for the bus device ID specified by the SEL/RESEL ID register (address 04_H). Note that higher transfer rate than the actual rate may be set if the input clock is high. (ex. 40 MHz: 04h or higher, 30 MHz: 03h or higher, 20 MHz: 02h or higher)

The following equation shows the relationship between the input block frequency and the maximum synchronous transfer speed.

(maximum synchronous transfer speed) × (transfer period) = input block frequency

(example) 10 Mbyte/s × 2 = 20 MHz

Table 3-5 shows the relationship between the transfer period register bit setting and the REQ/ACK signal minimum repetition time.

Table 3.2.15 Relationship between Transfer Period Register and REQ/ACK Signal Minimum Repetition Time

(unit: Tclf)

TP 4	TP 3	TP 2	TP 1	TP 0	TRANS PERIOD	ASSERT PERIOD	NEGATE PERIOD	TP 4	TP 3	TP 2	TP 1	TP 0	TRANS PERIOD	ASSERT PERIOD	NEGATE PERIOD
0	0	0	0	1	Prohibit	Prohibit	Prohibit	1	0	0	0	1	17	9	8
0	0	0	1	0	2	1	1	1	0	0	1	0	18	9	9
0	0	0	1	1	3	2	1	1	0	0	1	1	19	10	9
0	0	1	0	0	4	2	2	1	0	1	0	0	20	10	10
0	0	1	0	1	5	3	2	1	0	1	0	1	21	11	10
0	0	1	1	0	6	3	3	1	0	1	1	0	22	11	11
0	0	1	1	1	7	4	3	1	0	1	1	1	23	12	11
0	1	0	0	0	8	4	4	1	1	0	0	0	24	12	12
0	1	0	0	1	9	5	4	1	1	0	0	1	25	13	12
0	1	0	1	0	10	5	5	1	1	0	1	0	26	13	13
0	1	0	1	1	11	6	5	1	1	0	1	1	27	14	13
0	1	1	0	0	12	6	6	1	1	1	0	0	28	14	14
0	1	1	0	1	13	7	6	1	1	1	0	1	29	15	14
0	1	1	1	0	14	7	7	1	1	1	1	0	30	15	15
0	1	1	1	1	15	8	7	1	1	1	1	1	31	16	15
1	0	0	0	0	16	8	8	0	0	0	0	0	32	16	16

■ Setting Transfer Period Register

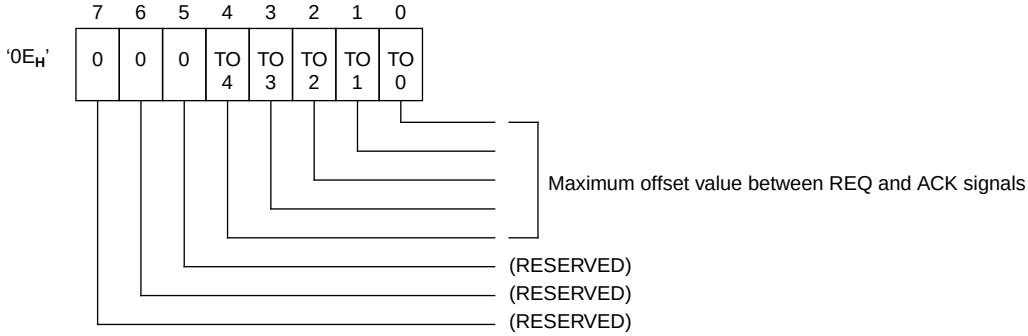
This register can store the transfer period for each connected devices. So, Please set this register after the counterpart's ID number is set to the SEL/RESEL ID register (address 04h). Once setting this register, the transfer period will be automatically set to this register from the next time or later, until a system reset or software reset is performed.

3.2.16 Transfer Offset Register (READ/WRITE)

The transfer offset registers specifies the maximum offset value for data phase synchronous transfers.

■ Transfer Offset Register (READ/WRITE)

The transfer offset register (READ/WRITE) bit configuration is shown below.



Set the synchronous transfer maximum offset value for the bus device ID specified by the SEL/RESEL ID register in the transfer offset register bits.

Table 3.2.16 shows the relationship between the transfer offset register bit setting and the maximum offset value for REQ and ACK signals.

Table 3.2.16 Relationship between Transfer Offset Register and Maximum Offset Value for REQ/ACK Signals

TO 4	TO 3	TO 2	TO 1	TO 0	TRANSFER-OFFSET
0	0	0	0	1	1
0	0	0	1	0	2
0	0	0	1	1	3
0	0	1	0	0	4
≈	≈	≈	≈	≈	≈
1	1	1	0	1	29
1	1	1	1	0	30
1	1	1	1	1	31
0	0	0	0	0	32

3.2.17 Window Address Register (WRITE)

The window address register (WRITE) specifies address 10h to 1Fh windows.

Window types include initial setting register window, MCS buffer window, user program memory window, and SCAM register window. (SCAM register window is available in MB86604L only.)

■ Window Address Register (WRITE)

The window address register (WRITE) bit configuration is shown below.

	7	6	5	4	3	2	1	0
'0F$_H$'	WA7	WA6	0	0	WA3	WA2	WA1	WA0

Next, the functionality of each bit is described.

BIT 7 to 6: Window selection

These bits select the window type for address 10_H to address $1F_H$.

WA7	WA6	Select window
0	0	MCS buffer window
0	1	SCAM register window
1	0	User program memory window
1	1	Initial setting window

Do not make SCAM register window setting in the MB86604A.

BIT 3 to 0: Window change

These bits change the window selection made by BIT 7 to 6. Set the first upper bit for the MCS buffer address and the four upper bits for the user program memory address.

Windows can be changed in 16 byte units.

Set “0” to BIT 3 to 0 to select the Initial Setting Window.

- MCS buffer window

WA7	WA6	0	0	WA3	WA2	WA1	WA0
0	0	0	0	0	0	0	X

First upper bit of MCS buffer address (32 bytes)

X: 0, 1 can be fixed

- User program window

WA7	WA6	0	0	WA3	WA2	WA1	WA0
1	0	0	0	X	X	X	X

Four upper bits of user program address (256 byte)

X: 0, 1 can be fixed

- SCAM register window

WA7	WA6	0	0	WA3	WA2	WA1	WA0
0	1	0	0	0	0	0	0

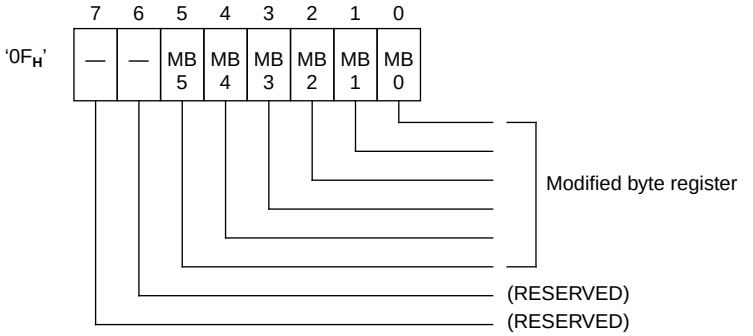
3.2.18 Modified Byte Register (READ)

The modified byte register (READ) is a 6-bit ring counter which indicates the number of bytes transferred by the SPC.

The meaning of this register's values changes by phase.

■ Modified Byte Register (READ)

The modified byte register (READ) bit configuration is shown below.



The meaning of modified byte register values changes by phase.

- In the message, command, status phases, it displays the number of bytes received by the SPC (Receive MCS buffer valid) or the number of bytes issued by the SPC (sent from the Send MCS buffer).
- In the data phase, it displays the number of bytes transferred between the SPC and the MPU or external memory.

The lower 6-bit values in the data byte register are set in the modified byte register and counted down for each single byte transfer.

When data transfer finishes with still remaining in the data register (abnormal completion), the number of remaining data bytes can be calculated from the data byte register and modified byte register values (see 5.9.4 for more on calculations). When the transfer ends with data still in the data register, the data in the data register becomes invalid.

Thus, in order to continue the data transfer, the bytes remaining in the data register must be calculated, the data pointer returned this amount, and the transfer operation started.

- In the Automatic selection response or Automatic receive modes, there are cases where receptions continues through various phases. The modified byte register displays the total number of bytes received during each phase.

3.2.19 BASIC Control Register Access

Indicates items to note during BASIC control register access.

■ **Interupt Status Register (READ), Command Step Register (READ)**

After the command is issued and until SPC BUSY (SPC status register: bit 6) becomes 0, do not access the interrupt status register (READ) and command step register (READ). Also, before a new command is issued, be sure to read interrupts and command steps for the previously executed command.

■ **Data block Register (READ/WRITE)**

Set data block register (READ/WRITE) right before issuing the command for data phase execution and when SPC BUSY (SPC status register: bit 6) is 0.

■ **Data Byte/MC Byte Registers (READ/WRITE)**

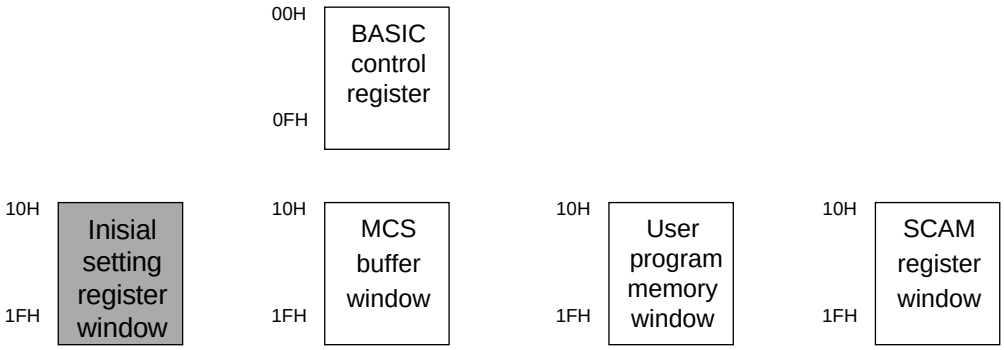
Set the data byte register (READ/WRITE) right before issuing the command for data phase execution and when SPC BUSY (SPC status register: bit 6) is 0.

Set the MC byte register (READ/WITE) right before issuing the commands which require that the MC byte register be set and when SPC BUSY (SPC status register: bit 6) is 0.

3.3 INITIAL SETTING REGISTER WINDOW

When the power is switched on, it is necessary to set initial register values because the internal registers are undefined.

After writing in the initial setting register windows, the SET UP REG command is issued to set initial values in SPC internal registers.



Note: SCAM register window is available in MB86604L only.

■ Initial Setting Register Window

Table 3-3 gives a list of registers in the initial setting register window.

Table 3.3 List of Initial Setting Register Window Registers

No.	Address					Write	Read
	A4	A3	A2	A1	A0		
10	1	0	0	0	0	Clock conversion setting register	←
11	1	0	0	0	1	Self ID setting register	←
12	1	0	0	1	0	Response mode setting register	←
13	1	0	0	1	1	SEL/RESEL mode setting register	←
14	1	0	1	0	0	SEL/RESEL retry setting register	←
15	1	0	1	0	1	SEL/RESEL timeout setting register	←
16	1	0	1	1	0	REQ/ACK timeout setting register	←
17	1	0	1	1	1	Asynchronous setup time setting register	←
18	1	1	0	0	0	Parity error detection setting registen	←
19	1	1	0	0	1	Interrupt enable setting register Group	←
1A	1	1	0	1	0	6/7 command length setting register	←
1B	1	1	0	1	1	DMA system setting register	←
1C	1	1	1	0	0	Auto-operation mode setting register	←
1D	1	1	1	0	1	SPC timeout setting register	←
1F	1	1	1	1	1	Revision indication register	←

■ Making Initial Setting Valid

The SET UP REG command must be issued to make the register values in the initial setting register window valid for the SPC.

These values do not become valid by host MPU write.
Issuing the SET UP REG command to the SPC sets up the internal circuits. When this is completed, a Command Complete interrupt is generated.
The clock conversion setting register (WRITE/READ) specifies the SPC input frequency.
Set bit values according to the input frequency.

The initial setting register window registers contain undefined data after the reset release. Therefore, write in all registers.

When the SPC does not receive the command or is operating automatically (Automatic selection/reselection response mode), the SET UP REG command is ignored and a Command Reject interrupt generated. In this case, SPC operation is not altered.

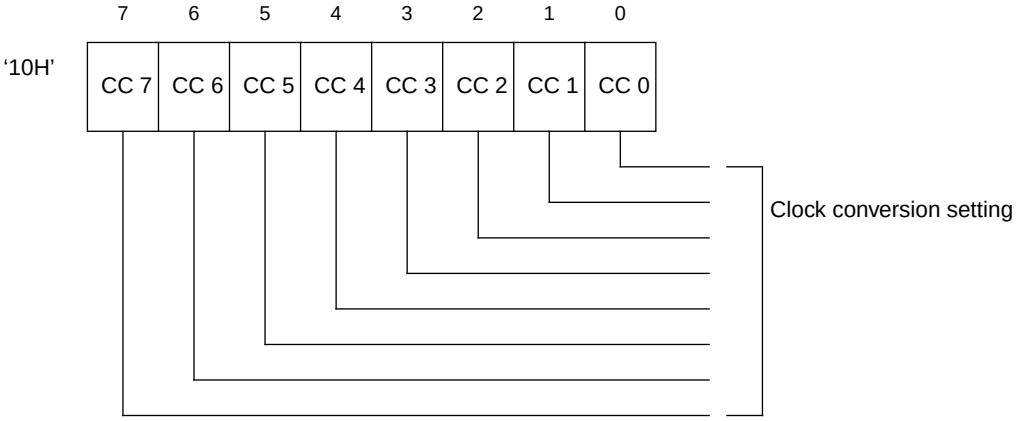
3.3.1 Clock Conversion Setting Register (WRITE/READ)

The clock conversion setting register (WRITE/READ) specifies the SPC input frequency.

SET bit values according to the input frequency.

■ Clock Conversion Setting Register (WRITE/READ)

The clock conversion setting register (WRITE/READ) bit configuration is shown below.



Set bit values according to the input frequency.

Table 3-8 shows the relationship between the clock conversion setting register bit setting and the input frequency.

Table 3.3.1 Relationship between Clock Conversion Setting Register Bit Setting and input Frequency

HEX	CC 7	CC 6	CC 5	CC 4	CC 3	CC 2	CC 1	CC 0	Input Frequency [MHz]	Internal Operating Frequency [MHz]
0B	0	0	0	0	1	0	1	1	20.0	20.0
10	0	0	0	1	0	0	0	0	20.0	10.0
32	0	0	1	1	0	0	1	0	30.0	15.0
38	0	0	1	1	1	0	0	0	30.0	10.0
53	0	1	0	1	0	0	1	1	40.0	20.0
59	0	1	0	1	1	0	0	1	40.0	13.3

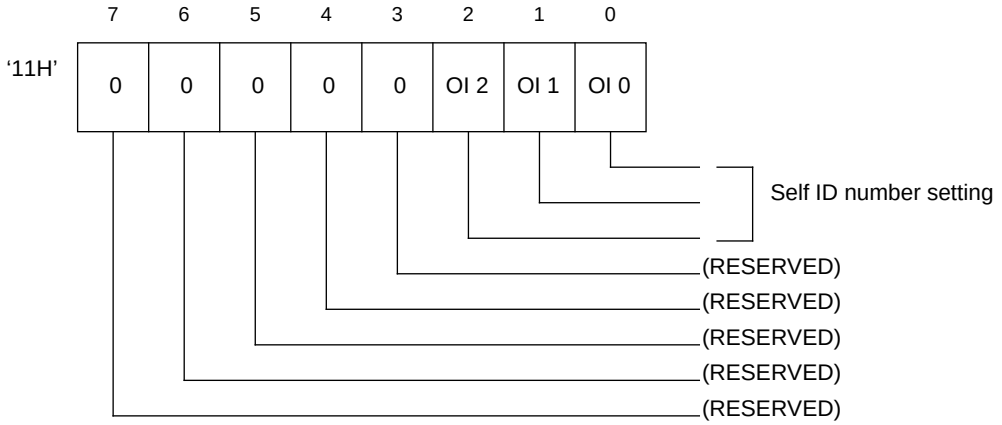
3.3.2 Self ID Setting Register (WRITE/READ)

The self ID setting register (WRITE/READ) specifies the SPC bus device ID number with a binary digit.

Use a binary digit for the SPC bus device ID number.

■ Self ID Setting Register (WRITE/READ)

Self ID setting register (WRITE/READ) bit configuration is shown below.

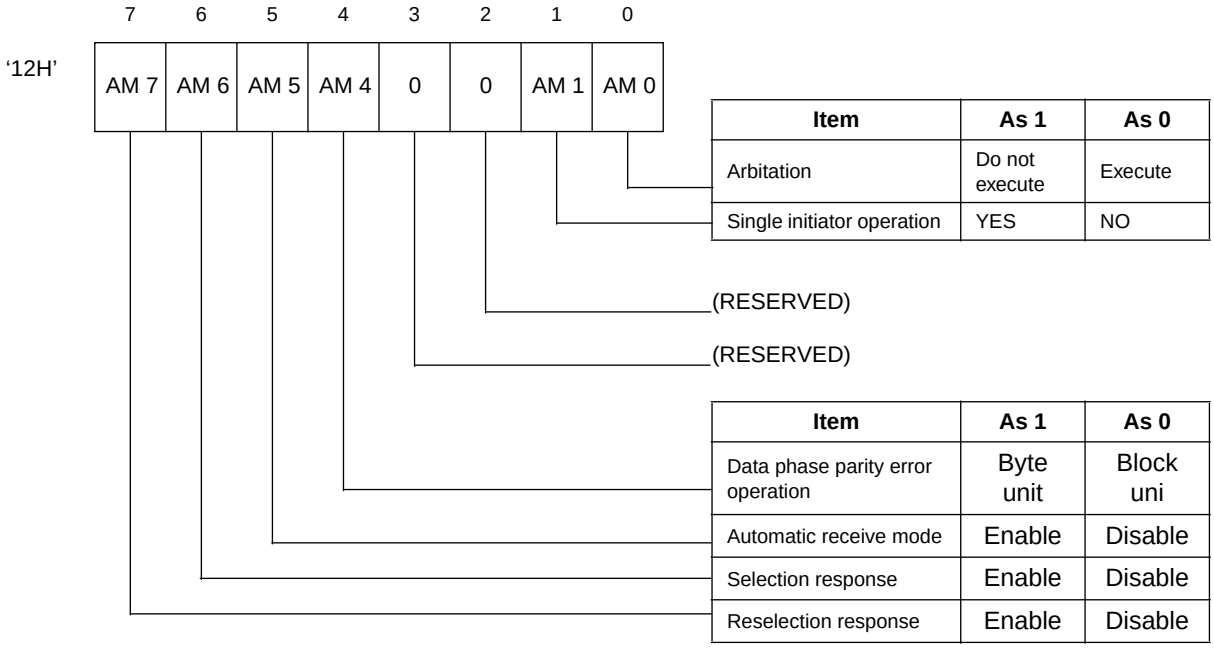


3.3.3 Response Mode Setting Register (READ/WRITE)

The Response mode setting register (READ/WRITE) specifies the SPC mode (see 3.3.3.1 to 3.3.3.6).

■ Response Mode Setting Register (READ/WRITE)

The response mode setting register (READ/WRITE) bit configuration is shown below.



Item	As 1	As 0
Arbitation	Do not execute	Execute
Single initiator operation	YES	NO

Item	As 1	As 0
Data phase parity error operation	Byte unit	Block uni
Automatic receive mode	Enable	Disable
Selection response	Enable	Disable
Reselection response	Enable	Disable

See 3.3.3.1 to 3.3.3.6 for more on bit functionality.

3.3.3.1 BIT 7 (Reselection Response)

BIT 7 (Reselection Response) specifies whether an initiator responds to target reselection.

■ **BIT 7 (Reselection Response)**

1: Respond to target reselect request and establish I-T nexus (see 3.3.4 for more on operation after establishing an I-T nexus).

0: No response to target reselect request (BSY signal not asserted in reselection phase).

3.3.3.2 BIT 6 (Selection Response)

BIT 6 (Selection Response) specifies whether a target responds to an initiator selection.

■ **BIT 6 (Selection Response)**

1: Respond to initiator select request and establish I-T nexus (see 3.3.4 for more on operation after establishing an I-T nexus).

0: No response to initiator select request (BSY signal not asserted in selection phase).

3.3.3.3 BIT 5 (Automatic Receive Mode)

BIT 5 (Automatic receive mode) specifies whether to receive messages in response to an attention condition produced by the initiator when functioning as the target. It also specifies whether to receive requested information (about the error phase) when functioning as the initiator and a phase error occurs during the execution of commands that involve phase transitions.

■ **BIT 5 (Automatic Receive Mode) Settings (Initiator Function)**

• When phase error occurs and target requests message in phase.

1: Phase error is recognized and a single message received.
When an ATN signal is being asserted, one message is received after it was negated.
With the final ACK signal still being asserted, an "Initial phase error & MSG-received" interrupt is reported.

0: Phase error is recognized and "Initial phase error" interrupt reported after negating ATN signal if it was being asserted.

• When phase error occurs and target requests status phase.

1: Phase error is recognized and a single status is received.
When an ATN signal is being asserted, one status is received after negation.
With the final ACK signal still being asserted, an "Initial phase error & status received" interrupt is reported. Whether or not the last ACK signal is negated depends on the ACK RESET mode setting. However, note that the command step reported at this time does not depend on the ACK RESET mode setting.(Regardless of the ACK RESET mode, the step numbers do not change.)

Regarding ACK RESET mode, refer to 3.3.13 Auto-Operation mode setting register.

Also, when an error occurs at the receipt of status, the error is reported. The command step numbers reported this case are the same as those in which a phase error occurs.

0: Phase error is recognized and "Initial phase error" interrupt reported.

The Automatic receive mode is invalid for phase errors that occur during a transfer (all bytes untransferred). When a phase error occurs during a transfer, a "phase error IN TRANSFER PROGRESS" interrupt is reported.

■ **BIT 5 (Automatic Receive Mode) Settings (Target Function)**

• When a command which ends in a phase other than the message out phase is executed and an ATN signal is asserted while the final ACK signal of the final phase is being asserted.

1: Attention condition produced by the initiator is detected, there is a transition to the message out phase, and a single message is received.
A "Command complete (ATN condition detected) & MSG received" interrupt is reported.

0: Attention condition produced by the initiator is detected, and a "Command complete (ATN condition detected)" interrupt is reported.
When an attention condition is detected during the execution of a command which ends with the message out phase, a "Command complete (ATN condition detected) interrupt is produced regardless of this bit setting.

• When a command is being executed as a target, and a transition to the next phase is taking place or an initiator attention condition is detected at the transfer data block boundary.

1: Attention condition produced by the initiator is detected, there is a transition to the message out phase, and as single message is received.

During data phase execution, the transfer is suspended at the transfer block boundary, there is a transition to the message out phase, and a single message is received.

A "Command stop (ATN condition detected) & MSG received" interrupt is reported.

0: Attention condition produced by the initiator is detected, and a "Command stop (ATN condition detected)" interrupt is reported.
During data in phase execution, the transfer is suspended at the transfer block boundary, and a "Command stop (ATN condition detected) interrupt is reported.

During data in phase execution, when a parity errors occurs on the target DMA or MPU side, the transfer is stopped between blocks (block unit stop), and an attention condition is produced by the initiator, "DMA parity error" or "MPU parity error" is reported without responding to the attention condition.

Note: When functioning as the target, after an attention condition is detected, there is a transition to the message out phase, and a single message is received. If another attention condition is detected after that, the following interrupt is reported.
Interrupt code = 61H
Sequence step = XXH (value after adding 1 to the number of steps of the phase prior to the transition to the message out phase)

■ **Storing Information Received by Automatic Operation**

• For Target operation:

If the command executed was not for command phase, the information is stored in Receive MCS buffer from address 0. If it was for command phase, the information is stored from the continuous address of CDB stored in Receive MCS buffer.

• For Initiator operation:

The information is stored in Receive MCS buffer from address 0.

3.3.3.4 BIT 4 (Operation for Data Phase Parity Error)

BIT 4 (Operation for Data Phase Parity Error) specifies SPC operation for when a data phase parity error is detected during SPC target operation.

■ **BIT 4 (Operation for Data Phase Parity Error)**

1: During data phase operation, the transfer is suspended when a parity error is detected, and an interrupt is reported (byte unit stop). See 5.9 for more information.

0: During data phase operation, the transfer is suspended after completing the transfer of the block where the parity error occurs, and an interrupt is reported (block unit stop).
However, when odd value is set in the data byte register, the parity error is detected in 2-block unit.

3.3.3.5 BIT 1 (Single Initiator Option)

BIT 1 (Single Initiator Option) specifies target operation.

When the SPC is incorporated in an application system operating in SCSI-1 mode and using the single initiator option, set this bit to 1.

■ **BIT 1 (Single Initiator Option)**

1: When the SPC is selected as the target, it responds even if the ID bit (target ID) on the data bus during the selection phase is just one bit. Bit 2 to 0 in the nexus status register indicate "000".

0: When the SPC is selected as the target, it does not respond if the ID bit on the data bus during the selection phase is 2 bits (for an initiator ID and the target ID).
When functioning as the initiator, do not set 1 at this bit, and self and target ID bits (2 bits) are always sent to the data bus during the selection phase regardless of this bit setting.

3.3.3.6 BIT 0 (Arbitration)

BIT 0 (Arbitration) specifies whether the SPC conducts arbitration.

When the SPC is incorporated in an application system operating in SCSI-1 mode and arbitration is not conducted, set this bit to 1.

■ **BIT 0 (Arbitration)**

1: Do not execute arbitration phase.

0: Execute arbitration phase.

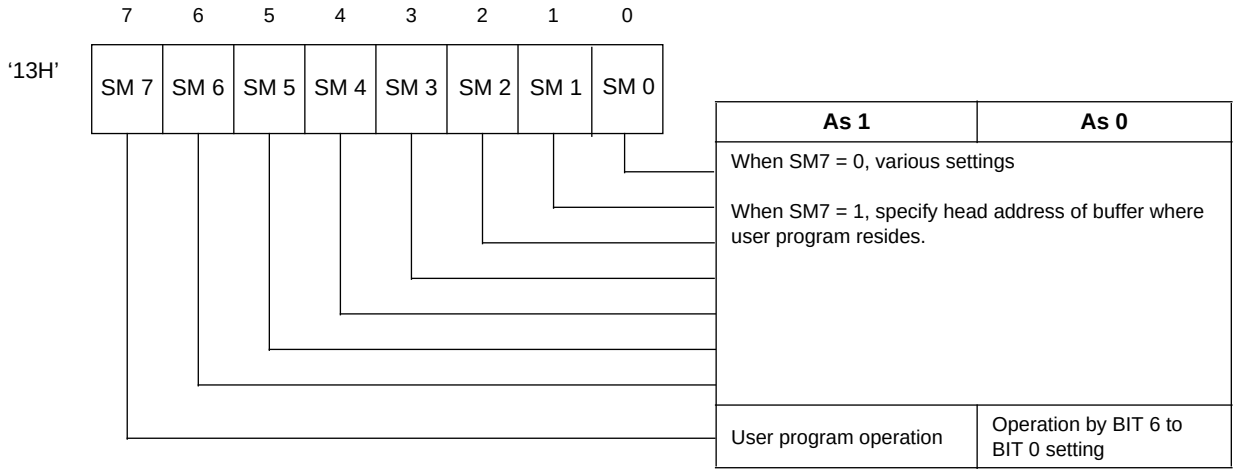
MEMO

3.3.4 SEL/RESEL Mode Setting Register (WRITE/READ)

The SEL/RESEL mode setting register (WRITE/READ) specifies automatic response for selection or reselection.

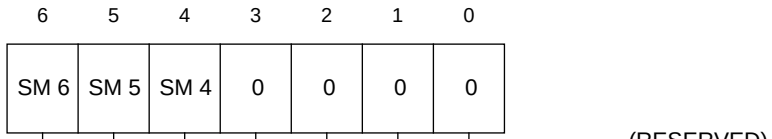
■ SEL/RESEL Mode Setting Register (WRITE/READ)

The SEL/RESEL mode setting register (WRITE/READ) bit configuration is shown below.



As 1	As 0
When SM7 = 0, various settings When SM7 = 1, specify head address of buffer where user program resides.	
User program operation	Operation by BIT 6 to BIT 0 setting

(When SM7 = 0)



Item	As 1	As 0
Phase after selection	Message out phase only	Message out or command phase
Automatic selection response	Enable	Disable
Automatic reselection response	Enable	Disable

Next, the functionality of each bit is described.

BIT 7: User program operation

This bit specifies whether to make a response originating from select/reselect with functions set previously in the user program or SPC. The meaning of BIT 6 to 0 is altered by this bit setting.

1: The head address of user program memory is written in BIT 6 to 0.
When the SPC is selected or reselected, process execution begins from the command in the user memory address specified by BIT 6 to 0.
Set the user program memory address at the same time as the command register (see Figure 3.2.8).

0: Provide response already prepared for SPC.

BIT 6: Automatic reselection response

When BIT 7 is 0, this bit setting affects initiator operation for reselection.

1: Respond to reselect request and operate until a message is received (see 3.3.4.1).

0: Respond to reselect request and report a "RESELECTED" interrupt.

BIT 5: Automatic selection response

When BIT 7 is 0, this bit setting affects target operation for selection.

1: Respond to select request and operate until a message or command is received (see 3.3.4.2).

0: Respond to select request and report "SELECTED" or "SELECTED WITH ATN" interrupt.

BIT 4: Post-selection phase

When BIT 7 is 0, this bit specifies the target automatic response. Specifies whether to limit the post-selection phase to the message phase (SCSI-2 mode) or to allow either the message phase or the command phase.

1: Limited to message out phase and when the selection phase ATN signal is not asserted by the initiator, an "AUTO mode phase error" interrupt is reported.

0: Depending on the state of the selection phase ATN signal, either the message out phase or command phase is selected. (Refer to 3.3.3.3 Automatic Receive operation whether or not the message is received transitting to Message-out phase when the SPC detects ATN condition generated by an initiator after receiving a command in command phase.)

3.3.4.1 Initiator Automatic Reselection Response

Figure 3.4 is a flowchart of the initiator automatic reselection response when BIT 6 of the SEL/RESEL mode setting register is set to 1, and Table 3-9 shows the corresponding operation sequence (step code chart).

■ **Initiator Automatic Reselection Response**

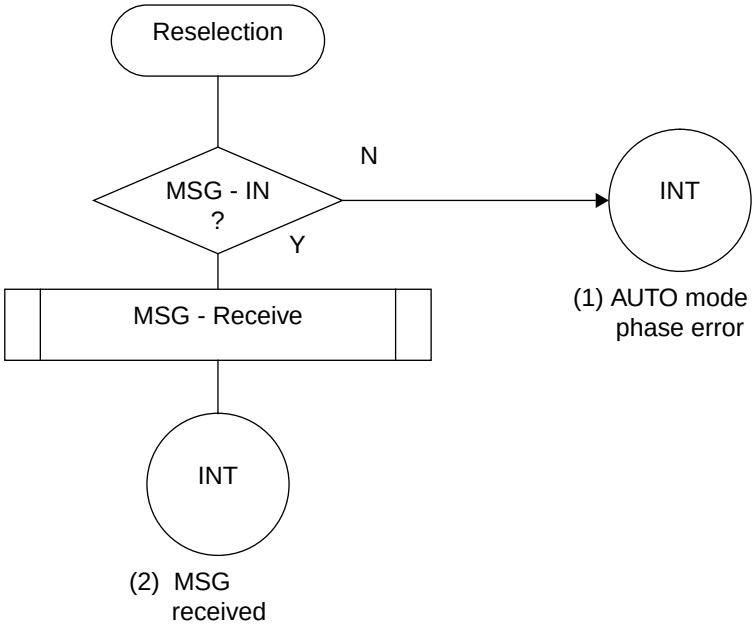


Figure 3.3.4.1 Initiator Automatic Reselection Response Flowchart

Table 3.3.4.1 Initiator Automatic Reselection Response Sequence (Step Code Chart)

Flow chart	Step	Description
Reselection	0	* Detect Reselect Establish nexus
MSG - IN	1	Wait for MSG-IN phase Execute MSG-IN phase
End	2	* Complete MSG receive

MEMO

3.3.4.2 Target Automatic Selection Response

Figure 3.5 is a flowchart of the target automatic selection response when BIT 5 of the SEL/RESEL mode setting register is set to 1, and Table 3-10 shows the corresponding operation sequence (step code chart).

■ **Target Automatic Selection Response**

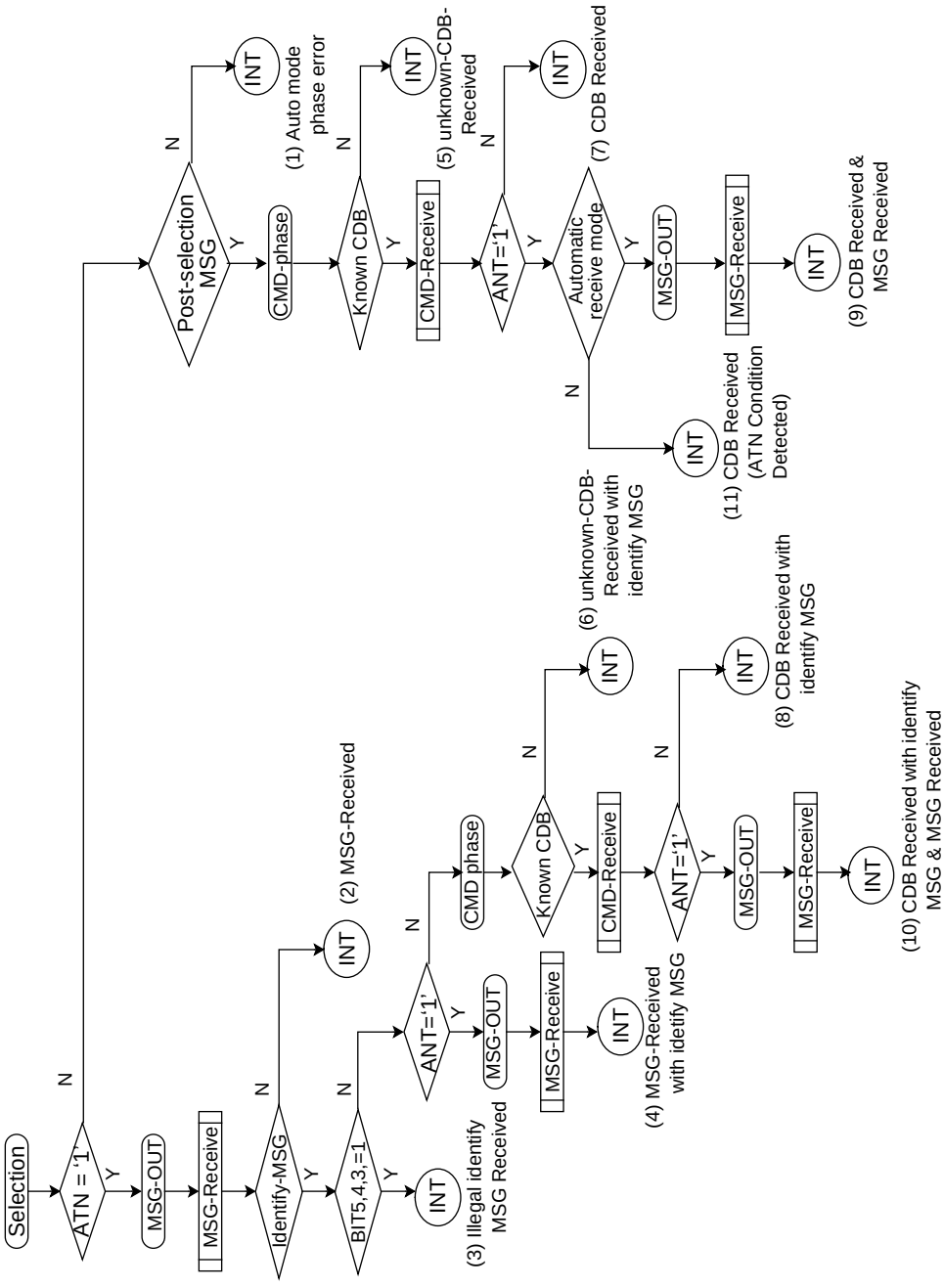
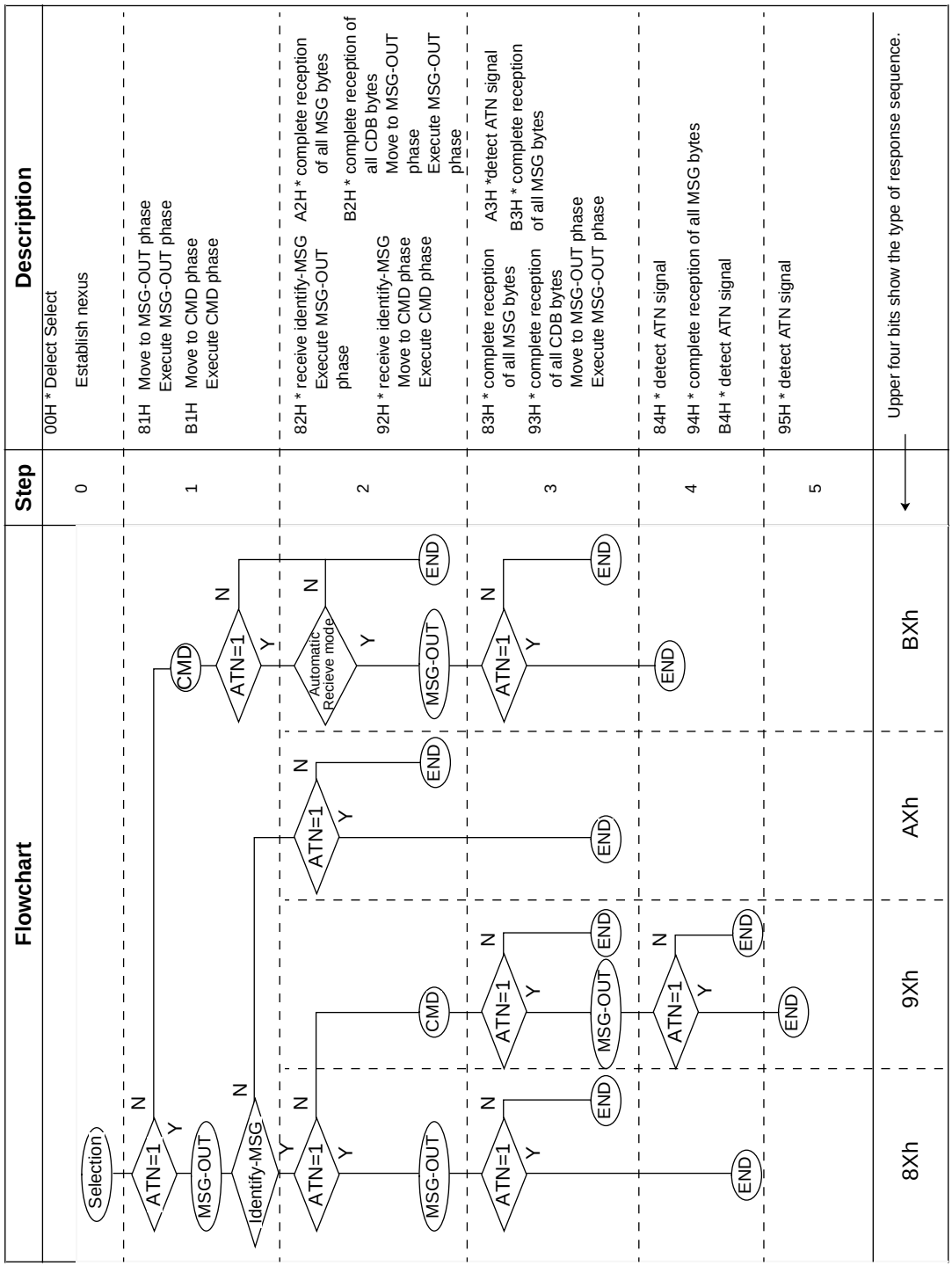


Figure 3.3.4.2a Target Automatic Selection Response Flowchart



Note: When the "identify-MSG" bits 5,4,3 are 1, an "illegal identify-MSG" interrupt is produced. At that time, the command status register code is the following.

- if the initiator is continuously asserting the ATN signal 82_H
- if the initiator is not continuously asserting the ATN signal 92_H

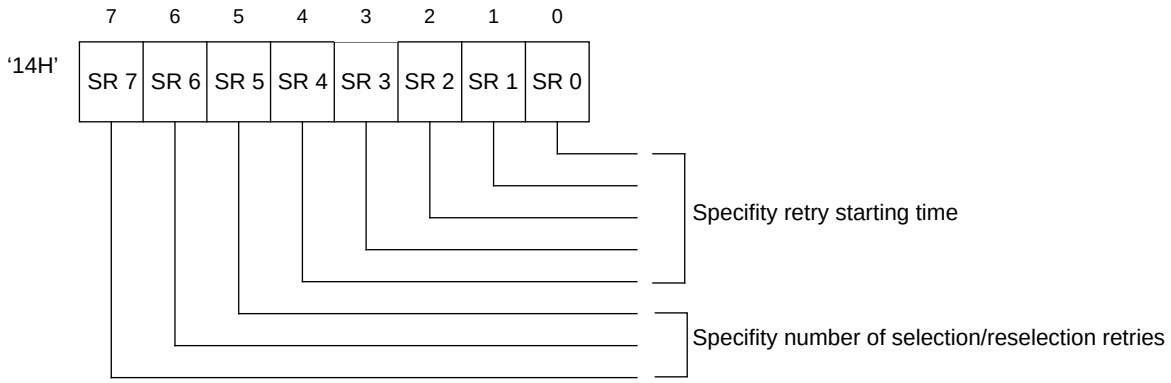
Table 3.3.4.2b Target Automatic Selection Response Sequence (Step Code Chart)

3.3.5 SEL/RESEL Retry Setting Register (READ/WRITE)

The SEL/RESEL retry setting register (READ/WRITE) specifies the number of selection/reselection retries and the retry starting time.

■ SEL/RESEL Retry Setting Register (READ/WRITE)

The SEL/RESEL retry setting register (READ/WRITE) bit configuration is shown below.



Next, the functionality of each bit is described.

BIT 7 to 5: Specifies number of selection/reselection retries

For commands which include selection/reselection, retry is performed the number of times specified by these bits even if selection/reselection timeout occurs. Specify the number of retries in accordance with Table 3-11.

Table 3.3.5 Relationship between SEL/RESEL Retry Setting Register and Number of Retries

SR 7	SR 6	SR 5	Number	SR 7	SR 6	SR 5	Number
1	1	1	Infinite	0	1	1	8
1	1	0	225	0	1	0	2
1	0	1	32	0	0	1	1
1	0	0	16	0	0	0	0

BIT 4 to 0 Specify retry starting time.

When selection/reselection timeout occurs during the initial selection/reselection, a retry is started after the passage of the time specified by this bit from a bus free state.

The time until a retry is determined by the following equation. However, when 00 is specified, there is not time count and retry is started once a bus free state exists.

$t_{CLF} \times C_{NV} \times [(SR^4 \times 2^{19}) + (SR3 \times 2^{17}) + (SR2 \times 2^{15}) + (SR1 \times 2^{13}) + (SR0 \times 2^{11})]$

t_{CLF}: Input frequency 1 cycle

C_{NV}: Calculate as

(value shown by clock conversion register BIT 4, BIT 3)

SR4 to SR0: BIT 4 to 0 setting values

(Example)

When the input frequency is 30 MHz and internal operating clock is 10 MHz:
t = 33.33 ns and C_{NV} = 3 so

$33.33 \times 3 \times [(SR4 \times 2^{19}) + (SR3 \times 2^{17}) + (SR2 \times 2^{15}) + (SR1 \times 2^{13}) + (SR0 \times 2^{11})]$

Thus, the time range which can be set in this register is as follows.

Maximum (SR4 to SR0 = 1) : 69 msec

Minimum (SR4 to SR1 = 0, SR0 = 1): 204 μsec

3.3.6 SEL/RESEL Timeout Setting Register (WRITE/READ)

The SEL/RESEL timeout setting register (WRITE/READ) specifies the selection/ reselection timeout delay time.

■ SEL/RESEL Timeout Setting Register (WRITE/READ)

The SEL/RESEL timeout setting register (WRITE/READ) bit configuration is shown below.

	7	6	5	4	3	2	1	0
'15H'	ST 7	ST 6	ST 5	ST 4	ST 3	ST 2	ST 1	ST 0

Specify the selection/reselection timeout time in this register.

The timeout time is determined by the following equation. However, when set to 00_H, timeout detection is not performed.

$t_{CLF} \times C_{NV} \times [(ST7 \times 2^{25}) + (ST6 \times 2^{23}) + (ST5 \times 2^{21}) + (ST4 \times 2^{19}) + (ST3 \times 2^{17}) + (ST2 \times 2^{15}) + (ST1 \times 2^{13}) + (ST0 \times 2^{11})]$

t_{CLF}: Input frequency 1 cycle

C_{NV}: Calculate as

(value shown by clock conversion register BIT 4, BIT 3)

ST7 to ST0: BIT 7 to 0 setting values

(Example)

When the input frequency is 30 MHz and internal operating clock is 10 MHz:
t_{CLF} = 33.33 ns and C_{NV} = 3 so

$33.33 \times 3 \times (ST7 \times 2^{25}) + (ST6 \times 2^{23}) + (ST5 \times 2^{21}) + (ST4 \times 2^{19}) + (ST3 \times 2^{17}) + (ST2 \times 2^{15}) + (ST1 \times 2^{13}) + (ST0 \times 2^{11})$

Thus, the time range which can be set in this register is as follows.

Maximum (SR7 to SR0 = 1): 4.4 sec
Minimum (ST7 to ST1 = 0, ST0 = 1): 204 µsec

Note: When 00H is set in this register, the timeout time is determined by the following equation.

$t_{CLF} \times C_{NV} \times 2^{27}$

(Example)

When the input frequency is 30 MHz and internal operating clock is 10 MHz:
t_{CLF} 33.33 ns and C_{NV} = 3 so, $33.33 \times 3 \times 2^{27} \fallingdotseq$ 13.4 sec

3.3.7 REQ/ACK Timeout Setting Register (WRITE/READ)

The REQ/ACK timeout setting register (WRITE/READ) specifies the REQ/ACK signal timeout time.

■ REQ/ACK Timeout Setting Register (WRITE/READ)

The REQ/ACK timeout setting register (WRITE/READ) bit configuration is shown below.

	7	6	5	4	3	2	1	0
'16H'	RT 7	RT 6	RT 5	RT 4	RT 3	RT 2	RT 1	RT 0

The specified REQ/ACK signal timeout times are explained below.

For asynchronous transfers (target): Time after the REQ signal is asserted and until the initiator asserts the ACK signal.

For asynchronous transfers (initiator): Time after the ACK signal is asserted and until the target asserts the REQ signal.

For synchronous transfers (target only): Time after the REQ signal is asserted until an ACK signal which makes the offset value 0 is received.

The timeout time is determined by the following equation. However, timeout detection is not performed when set at 00_H.

$t_{CLF} \times C_{NV} \times (RT7 \times 2^{28}) + (RT6 \times 2^{26}) + (RT5 \times 2^{24}) + (RT4 \times 2^{22}) + (RT3 \times 2^{20})$
$+ (RT2 \times 2^{18}) + (RT1 \times 2^{16}) + (RT0 \times 2^{14})$

t_{CLF}: Input frequency 1 cycle
C_{NV}: Calculate as

(value shown by clock conversion register BIT 4, BIT 3), when the clock conversion value is $0X_H$

RT7 to RT0: BIT 7 to 0 setting values

(Example)

When the input frequency is 30 MHz and internal operating clock is 10 MHz:
t_{CLF} = 33.33 ns and C_{NV} = 3 so

$33.33 \times 3 \times (RT7 \times 2^{28}) + (RT6 \times 2^{26}) + (RT5 \times 2^{24}) + (RT4 \times 2^{22}) + (RT3 \times 2^{20})$
$+ (RT2 \times 2^{18}) + (RT1 \times 2^{16}) + (RT0 \times 2^{14})$

Thus, the time range which can be set in this register is as follows.

Maximum (when RT7 to RT0 = 1): 35.5 sec

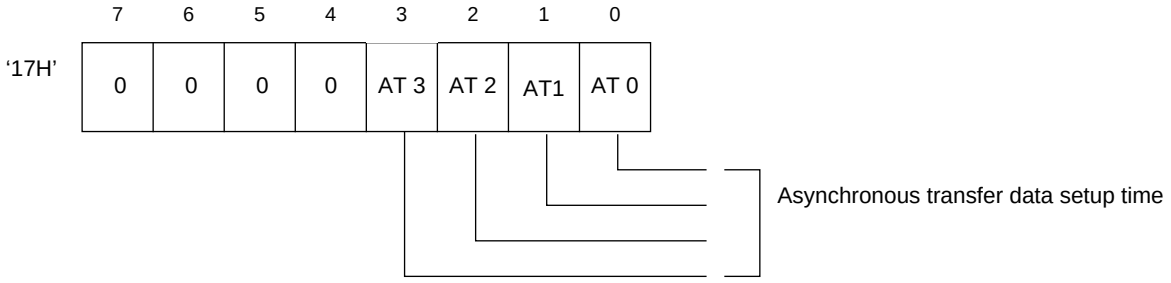
Minimum (when RT7 to RT1 = 0, RT0 = 1): 1.6 msec

3.3.8 Asynchronous Setup Time Setting Register (WRITE/READ)

The asynchronous setup time setting register (WRITE/READ) specifies the timing of asserting REQ/ACK signals for asynchronous data transfers.

■ Asynchronous Setup Time Setting Register (WRITE/READ)

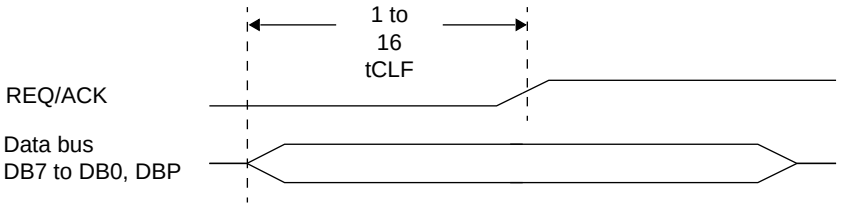
The asynchronous setup time setting register (WRITE/READ) bit configuration is shown below.



The specified data setup times are explained below.

Target: Time from placing the data in the data bus until the REQ signal is asserted.

Initiator: Time from placing the data in the data bus until the ACK signal is asserted.



AT 3	AT 2	AT 1	AT 0	SET UP TI ME
0	0	0	1	1tclf
0	0	1	0	2tclf
0	0	1	1	3tclf
≈				≈
1	1	1	1	15tclf
0	0	0	0	16tclf

MEMO

3.3.9 Parity Error Detect Setting Register (WRITE/READ)

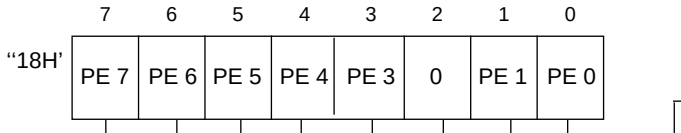
The parity error detect setting register (WRITE/READ) specifies the parity type, generation, and check in accordance with the MPU, DMA configuration.

Parity checks cannot be performed until this register is set up (using the SET UP REG command).

Parity generation enable/disable is only valid for the data phase.

■ Parity Error Detect Setting Register (WRITE/READ)

The parity error detect setting register (WRITE/READ) bit configuration is shown below.



Item	As 1	As 0
DMA parity generate	Yes	No
DMA parity check	Yes	No
(Reserve)		
MPU parity generate	Yes	No
MPU parity check	Yes	No
System side parity type	Odd	Even
SCSI parity generate	Yes	No
SCSI parity check	Yes	No

Next, the functionality of each bit is described.

BIT 7: SCSI parity check

1: Functioning as the initiator , a parity check is performed for message in phase, status phase, and data in phase input data and data out phase output data.

Functioning as the target, a parity check is performed for message output phase, command phase, and data out phase input data and data in phase output data.

0: Parity check is not performed.

BIT 6: SCSI parity generate

1: Functioning as the initiator, parity generate is performed for data in phase input data and data out phase output data.

Functioning as the target, parity generate is performed for data out phase input data and data in phase output data.

0: Parity is not generated.

BIT 5: System side parity type

This bit specifies the MPU/DMA parity type for application systems incorporating the SPC.

1: Odd parity

0: Even parity

BIT 4: MPU parity check

1: During register write, data parity check is performed as data is written to the SPC registers (check is not performed for register read operations).

During data register read/write, a parity check is performed for data written in data registers while the data phase is being executed for program transfer and data read from the data registers.

0: Parity check is not performed.

BIT 3: MPU parity generate

1: Parity is generated for data written in the data register while the data phase is being executed for program transfer and data read from the data registers.

0: Parity is not generated.

During register write, parity is not generated regardless of this bit setting. Also, during register read, parity is generated without any conditions (do not perform MPU parity checks for systems without a parity bit for the MPU data bus).

BIT 1: DMA parity check.

1: When the data phase is being executed for DMA transfers, a parity check is performed for data input from the DMA data bus and data output to the DMA data bus.

0: Parity check is not performed.

BIT 0: DMA parity generate

1: When the data phase is being executed for DMA transfers, parity is generated for data input from the DMA data bus and data output to the DMA data bus.

0: Parity is not generated.

See appendix B for more on how parity check and parity generate are performed within the SPC circuits.

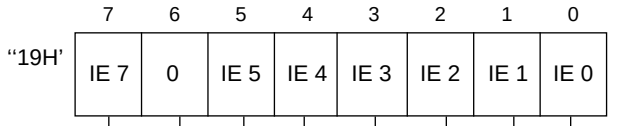
Note: During the external bus access, the parity check and generate are not performed regardless of the register setting. Also, when performing the SCSI bus parity check in an application which does not have the parity bit on the MPU and DMA buses, please make "MPU and DMA parity generations" Enable. Furthermore, please make "Pull-up" for UDP, LDP, UDMDP, and LDMDP pins with approx. 10 kΩ resistor.

3.3.10 Interrupt Enable Setting Register (WRITE/READ)

The interrupt enable setting register (WRITE/READ) masks interrupt signals (INT) by interrupt cause groups.

■ Interrupt Enable Setting Register (WRITE/READ)

The interrupt enable setting register (WRITE/READ) bit configuration is shown below.



Item	As 1	As 0
SCSI RESET CONDITION	Enable	Disable
TRANSFER ERROR	Enable	Disable
PHASE ERROR	Enable	Disable
REPORT	Enable	Disable
SEL/RESELECTION	Enable	Disable
AUTO SELECT/ RESELECT	Enable	Disable
(RESERVED)		

Next, the functionality of each bit is described.

BIT 7: All interrupt causes

1: Depends on BIT 5 to 0 Setting

0: Disable for all interrupt causes regardless of the BIT 5 to 0 setting.

The upper three bit values of the interrupt code shown in the interrupt status register (address 04_H) correspond to bit values of this register (Table 3.3.10).

See 5.7 for more on interrupt codes. Also, SPC reports interrupts to it's interrupt status register (04_H) regardless of this register setting.

Table 3.3.10 Relationship between Interrupt Enable Setting Register and Interrupt Codes

IE BIT	Interrupt code			Interrput code type
	IS7	IS6	IS5	
0	0	0	0	SCSI reset
1	0	0	1	Interrupt related to transfers
2	0	1	0	Interrupt related to phase transitions
3	0	1	1	Interrupt related to reports
4	1	0	0	Interrupt related to selection/reselection
5	1	0	1	Interrupt related to automatic selection/reselection

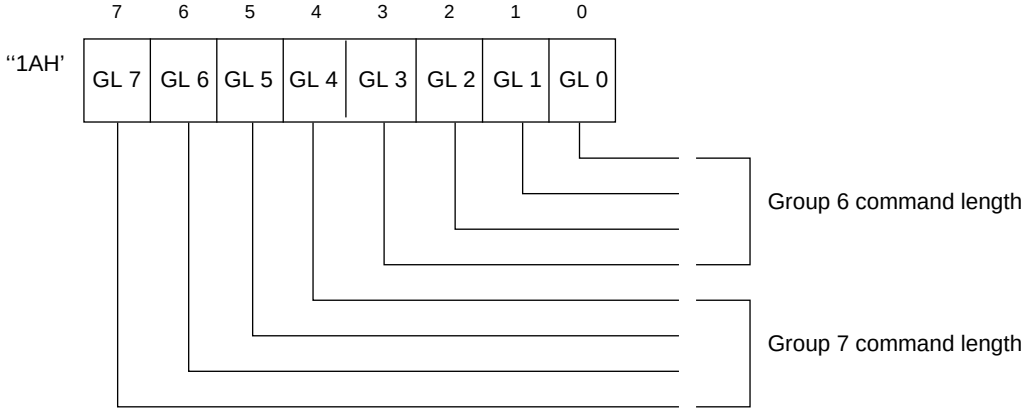
MEMO

3.3.11 Group 6/7 Command Length Setting Register (WRITE/READ)

The group 6/7 command length setting register (WRITE/READ) specifies the group 6/7 command length.

■ Group 6/7 Command Length Setting Register (WRITE/READ)

The group 6/7 command length setting register (WRITE/READ) bit configuration is shown below.



Set the values shown in Table 3-13 and Table 3-14 in the register bits in accordance with the group 6/7 command length

.

Table 3.3.11 List of Bit Setting for Group 6/7 Command Length Setting Register (Group 6)

GL 3	GL 2	GL 1	GL 0	Group 6 command length
0	0	0	0	Not specified
0	0	0	1	2 byte
0	0	1	0	4 byte
0	0	1	1	6 byte
≈				≈
1	1	1	0	28 byte
1	1	1	1	30 byte

Table 3.3.12 List of Bit Setting for Group 6/7 Command Length Setting Register (Group 7)

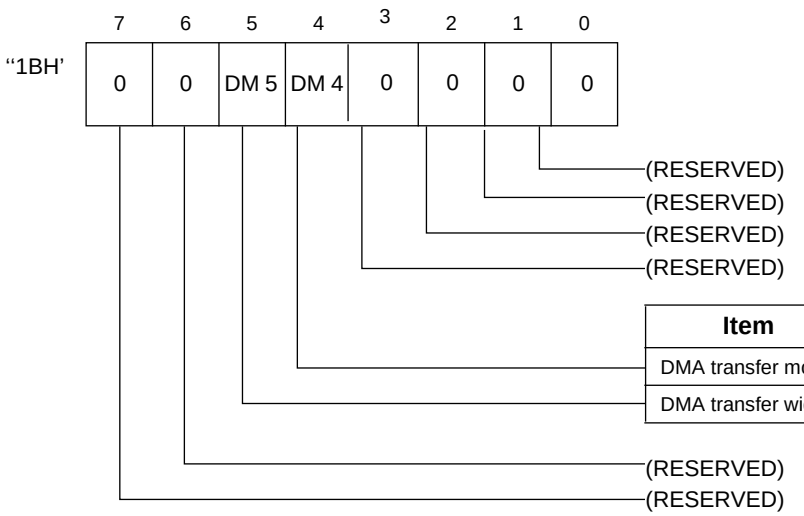
GL 7	GL 6	GL 5	GL 4	Group 7 command length
0	0	0	0	Not specified
0	0	0	1	2 byte
0	0	1	0	4 byte
0	0	1	1	6 byte
≈				≈
1	1	1	0	28 byte
1	1	1	1	30 byte

3.3.13 DMA System Setting Register (WRITE/READ)

The DMA system setting register (WRITE/READ) specifies the DMA transfer mode.

■ DMA System Setting Register (WRITE/READ)

The DMA system setting register (WRITE/READ) bit configuration is shown below.



Item	As 1	As 0
DMA transfer mode	Burst mode	Handshake mode
DMA transfer width	16 bits	8 bits

Next, the functionality of each bit is described.

BIT 5: Set DMA transfer width

This bit specifies the DMA transfer width.

1: 16 bits

0: 8 bits

BIT 4: DMA transfer mode

This bit specifies the DMA transfer mode.

1: Burst transfer mode
While DMA transfer is possible, DREQ signal output is kept in an active state.

0: Handshake mode
Transfer is performed by DREQ signal output and DACK signal input.

Handshake: One format for reading and writing data between devices. In this case, the following transfers are performed between the SPC and DMAC.

(1) When data is transferred from the SPC to the DMAC

1. SPC sets up the data to be sent.
2. SPC asserts the DREQ signal.
3. When the DMAC detects that the DREQ signal has been asserted, it receives the data and asserts the $\overline{\text{DACK}}$ signal.
4. When the SPC detects that the $\overline{\text{DACK}}$ signal has been asserted, it negates the DREQ signal.

(2) When data is transferred from the DMAC to the SPC

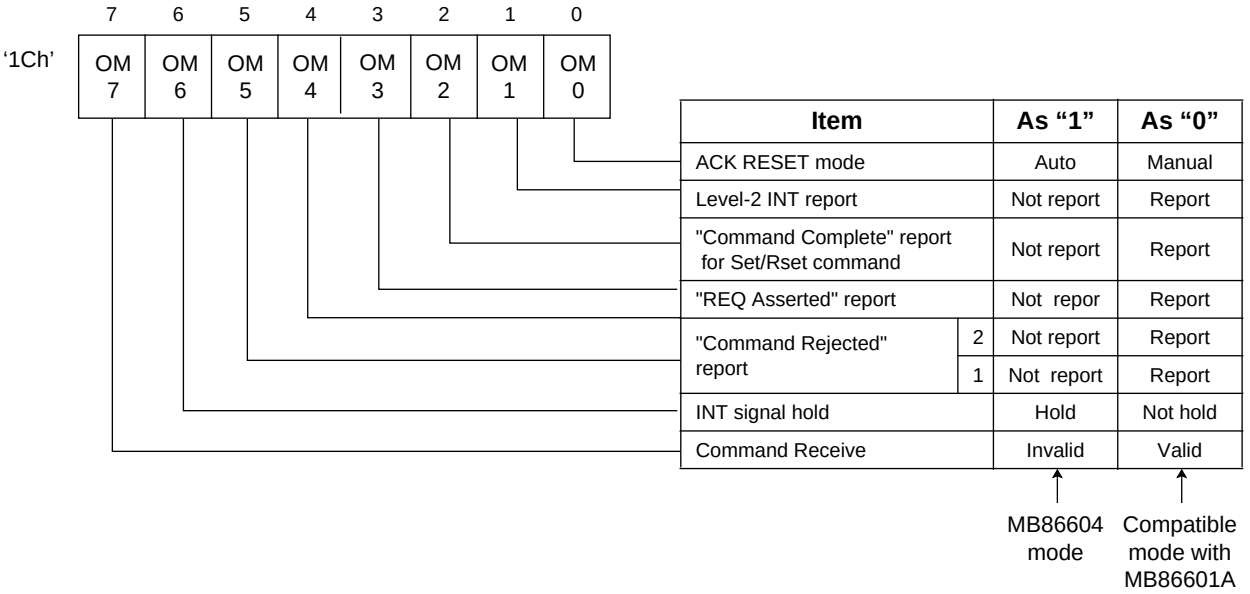
1. SPC asserts the DREQ signal.
2. After the DMAC detects that the DREQ signal has been asserted, the transfer data is sent to the bus.
3. DMAC asserts the $\overline{\text{DACK}}$ signal.
4. When the SPC detects that the $\overline{\text{DACK}}$ signal has been asserted, the DREQ signal is negated.

The next time that data is detected, the steps 1. to 4. are repeated. However, until the $\overline{\text{DACK}}$ signal is negated, the next DREQ signal cannot be asserted.

3.3.14 Automatic Operation Mode Setting Register (WRITE/READ)

This register sets the SPC various operation modes.

■ **Automatic Operation Mode Setting Register (WRITE/READ)**



Item		As "1"	As "0"
ACK RESET mode		Auto	Manual
Level-2 INT report		Not report	Report
"Command Complete" report for Set/Rset command		Not report	Report
"REQ Asserted" report		Not repor	Report
"Command Rejected" report	2	Not report	Report
	1	Not report	Report
INT signal hold		Hold	Not hold
Command Receive		Invalid	Valid

Each bit function is described in the next section.

Bit 7: Command Receive Invalid Mode

This bit specifies whether or not SPC accepts the command received if the interrupt factor (cause) is held in the Interrupt Status Register (address 04_H).

When "1": SPC ignores the command received and does not report the "Command Rejected" interrupt. Once all the interrupt factor is read out from the interrupt status register, it accepts the command.

When "0": SPC accepts and executes the command received.

Bit 6: INT Signal Hold Mode

This bit specifies whether or not the SPC holds the INT signal until all the interrupt factor and step code is read out.

When "1": SPC holds the INT signal until all the interrupt and step code held in the SPC is read out.

When "0": SPC makes the INT signal "Inactive" once an interrupt code is read out, even if SPC is holding multiple interrupt and step codes. Also, SPC makes the INT signal "Inactive" at which SPC BSY bit (Bit 6 of SPC Status Register) becomes "1" even if any interrupt code is read out.

Bit 5: Command Rejected Report Mode -1

This bit specifies whether or not the SPC reports the "Command Rejected" interrupt when the command is received during SPC BUSY state (SPC BSY=1).

When "1": For example, while SPC is responding to the Selection/Reselection; if a command which selects/reselects is issued, SPC reports only the interrupt for the selection/reselection. However, when the selection/reselection command is issued after reading out the interrupt factor for the selection/reselection response, SPC reports the "Command Invalid" interrupt.

When "0": SPC reports the "Command Rejected" interrupt after reporting the interrupt for the command currently in-process or selection/reselection response.

Bit 4: Command Rejected Report Mode -2

This bit specifies whether or not the SPC reports the "Command Rejected" interrupt when SPC responded to the selection/reselection from other device while SPC is executing a selection/reselection command or waiting for it.

When "1": SPC reports only the interrupt for selection/reselection and does not report the "Command Rejected" interrupt.

When "0": SPC reports the interrupt for selection/reselection after reporting the "Command Rejected" interrupt.

Bit 3: REQ Asserted Report Mode

This bit specifies whether or not the SPC reports the "REQ Asserted" interrupt when detecting the REQ signal asserted by a target. (For Initiator operation)

When "1": SPC does not report the "REQ Asserted" interrupt. However, SPC reports it when detecting the REQ signal in the data-phase, regardless of this bit setting.

When "0": SPC reports the "REQ Asserted" interrupt.

Bit 2: Command Complete Report Mode for SET/RESET commands

This bit specifies whether or not SPC reports the "Command Complete" interrupt for the SET/RESET commands.
This is valid in the user program operation.

When "1": SPC does not report the "Command Complete".

When "0": SPC reports the "Command Complete".

The applicable SET/RESET commands for this mode are as follows:

1) Initiator: SET ATN (0Ah), RESET ATN (0Bh), SET ACK (0Ch), RESET ACK (0Dh)

2) Target: SET REQ (31h), RESET REQ (32h)

3) Common: SET RST (48h), RESET RST (49h)

Bit 1: Level-2 Interrupt Report Mode

This bit specifies whether or not the SPC reports the Level-2 interrupts occurred in the user program operation.

When "1": SPC does not report the Level-2 interrupts and does not retain them in the Interrupt Status Register. The result of command execution is retained in the accumulator and within the SPC. The retained values are as follows:

1) If "Command Complete": retains "00h".

2) If "Initial Phase Error" or "ATN Condition Detected": retains "01h".

The result of the execution can be identified by the conditional branch instruction following the execution. See Section 5.4.2 and 5.5.2 for this mode, which shows the examples of user program flow.

When "0": SPC reports the Level-2 interrupts and sequence step codes. In this case, reading out and operating the interrupt and step codes are required every execution of discrete command using MOVE instruction. See Section 5.4.1 and 5.5.1 for this mode, which shows the examples of user program flow.

Bit 0: Auto ACK RESET Mode

This bit specifies how to negate the final ACK signal for the transfer command. (For Initiator operation)

When "1": SPC automatically negates the final ACK signal during the transfer command. (Auto ACK reset)

When "0": SPC does not negate the final ACK signal automatically, but negates it with "RESET ACK" command. (Manual ACK reset)

1) Commands which complete with automatically negating the final ACK regardless of this bit setting:

- SEND 1-MSG with ATN (19h), - SEND or RECEIVE DATA from MPU or DMA / SEND or RECEIVE DATA to MPU or DMA with Padding (14h, 15h, 16h, 17h)

2) Commands/Operations which complete with negating the final ACK by "RESET ACK" Command (Manual ACK Reset) regardless of this bit setting:

- RECEIVE N-BYTE-MSG (07h), - RECEIVE MSG (1Ah), - RESET ATN (0Bh), - When "Initial Phase Error & MSG Received" interrupt is generated. - When Auto-Reselection response is operating.

3) Commands/Operations which depend on this bit setting:

- Transfer Commands except 1) and 2) above.

- SET ATN (0Ah)

- When "Initial Phase Error & Status Received" interrupt is generated.

Please note that the command step number in the Auto ACK Reset Mode will be (08h + Step code with Manual ACK reset mode). However, the following commands and interrupts are the exceptions for this command step number:

- SEND or RECEIVE DATA from MPU or DMA / SEND or RECEIVE DATA to MPU or DMA with Padding (14h, 15h, 16h, 17h)

- RECEIVE N-MSG (07h), RECEIVE MSG (1Ah), RESET ATN (0Bh)

- When "Initial Phase Error & MSG Received" is generated.

- When "Initial Phase Error & Status Received" is generated.

- When Auto-Reselection response is operating.

MEMO

3.3.15 SPC Timeout Setting Register (WRITE/READ)

This register specifies the timeout value for SPC BUSY state.

■ SPC Timeout Setting Register

The configuration of this register is shown below.

	7	6	5	4	3	2	1	0
‘‘1DH’	TO 7	TO 6	TO 5	TO 4	TO 3	TO 2	TO 1	TO 0

Set the SPC timeout value at this register. When SPC is in Busy state longer than the time specified in this register, SPC makes TMOUT pin "Active" and sets "1" to the SPC Timeout bit (bit 4 of SPC Status Register) to report "SPC timeout".

SPC does not detect the timeout if "00h" is set at this register.

The SPC timeout is calculated by the following equation:

$t_{CLF} \times C_{NV} \times [(TO7 \times 2^{30}) + (TO6 \times 2^{28}) + (TO5 \times 2^{26}) + (TO4 \times 2^{24}) + (TO3 \times 2^{22}) + (TO2 \times 2^{20}) + (TO1 \times 2^{18}) + (TO0 \times 2^{16})]$

t_{CLF}: Input clock Frequency (1 cycle)

C_{NV}: Value indicated in Bit 4 and Bit 3 of Clock Conversion Register.

TO7 - TO0: The above mentioned register values

3.3.16 Revision Indication Register (WRITE/READ)

This register indicates the SPC chip revision.

■ **Revision Indication Register (WRITE/READ)**

The configuration of this register is shown below.

	7	6	5	4	3	2	1	0
‘‘1FH’	RV 7	RV 6	RV 5	RV 4	RV 3	RV 2	RV 1	RV 0

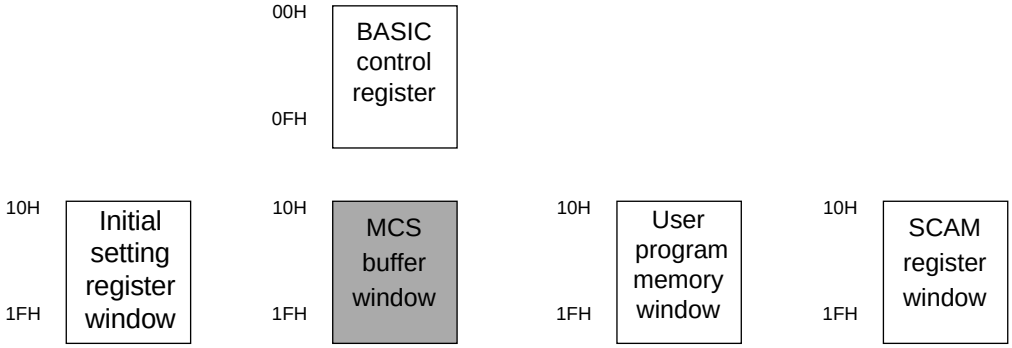
In order to see the device revision code, please take the following ways.

1. Write "00h" to the address "1Fh" at the initial setting.
2. Issue the "Setup REG" command.
3. Read out the value for the address "1Fh" after completing the "Setup REG" command.

The value except "00h" read out is the chip revision.

3.4 MCS BUFFER WINDOW

A 32-byte memory for sending and receiving messages, commands, and statuses with two 16-byte windows.



Note: SCAM register window is available in MB86604L only.

■ **MCS Buffer Window**

The MCS buffer window buffer list is shown in Table 3-15.

Table 3.4 MCS Buffer Window Buffer List

No	Address					Write	Read
	A4	A3	A2	A1	A0		
10	1	0	0	0	0	SEND MCS buffer	RECEIVE MCS buffer
11	1	0	0	0	1	SEND MCS buffer	RECEIVE MCS buffer
12	1	0	0	1	0	SEND MCS buffer	RECEIVE MCS buffer
13	1	0	0	1	1	SEND MCS buffer	RECEIVE MCS buffer
14	1	0	1	0	0	SEND MCS buffer	RECEIVE MCS buffer
15	1	0	1	0	1	SEND MCS buffer	RECEIVE MCS buffer
16	1	0	1	1	0	SEND MCS buffer	RECEIVE MCS buffer
17	1	0	1	1	1	SEND MCS buffer	RECEIVE MCS buffer
18	1	1	0	0	0	SEND MCS buffer	RECEIVE MCS buffer
19	1	1	0	0	1	SEND MCS buffer	RECEIVE MCS buffer
1A	1	1	0	1	0	SEND MCS buffer	RECEIVE MCS buffer
1B	1	1	0	1	1	SEND MCS buffer	RECEIVE MCS buffer
1C	1	1	1	0	0	SEND MCS buffer	RECEIVE MCS buffer
1D	1	1	1	0	1	SEND MCS buffer	RECEIVE MCS buffer
1E	1	1	1	1	0	SEND MCS buffer	RECEIVE MCS buffer
1F	1	1	1	1	1	SEND MCS buffer	RECEIVE MCS buffer

■ **SEND MCS Buffer (WRITE)**

This buffer is for writing messages, commands, and statuses to be sent (32 bytes).

Start writing the messages, commands, and statuses to be sent form address 0.

The SPC starts sending data from address 0.

When a command that requires the SEND MCS buffer to be accessed is sent, access to the SEND MCS buffer is not allowed until that command is completed.

■ **RECEIVE MCS Buffer (READ)**

This buffer is for reading messages, commands, and statuses received by the SPC (32 bytes).

Start reading the received messages, commands, and statuses from address 0.

The SPC starts saving the received data from address 0.

When a command that requires the RECEIVE MCS buffer to be accessed is sent, access to the RECEIVE MCS buffer is not allowed until that command is completed.

Also, when data is received, be sure to read the RECEIVE MCS buffer (the number of read bytes is displayed in the modified byte register).

Until all the data is read out, the next message, command, or status cannot be received.

Note: Access the RECEIVE MCS Buffer in the following cases.

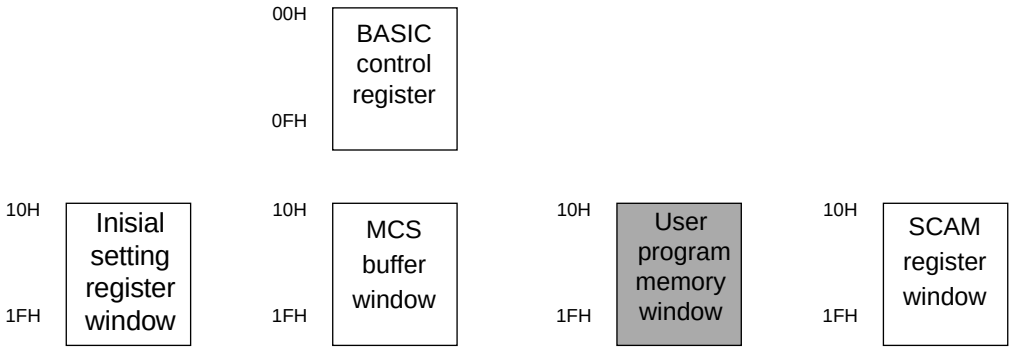
1) Initiator operation:
   Access until a RESET ACK command is issued after completion of the command which receives the data to the RECEIVE MCS Buffer or after completion of the automatic reselection response. When Auto ACK Reset mode is enabling, read out MCS buffer before issuing the next command.

2) Target operation:
   Access until the phase changes after completion of the command which receives the data to the RECEIVE MCS Buffer or after completion of the automatic selection response.

3.5 USER PROGRAM MEMORY WINDOW

A 256-byte memory for reading and writing user programs with 16 16-byte windows.

1.



Note: SCAM register window is available in the MB86604L only.

■ **User Program Memory Window**

A list of user program memory window memory areas is shown in Table 3-16.

Table 3.5 User Program Memory Window Memory Area List

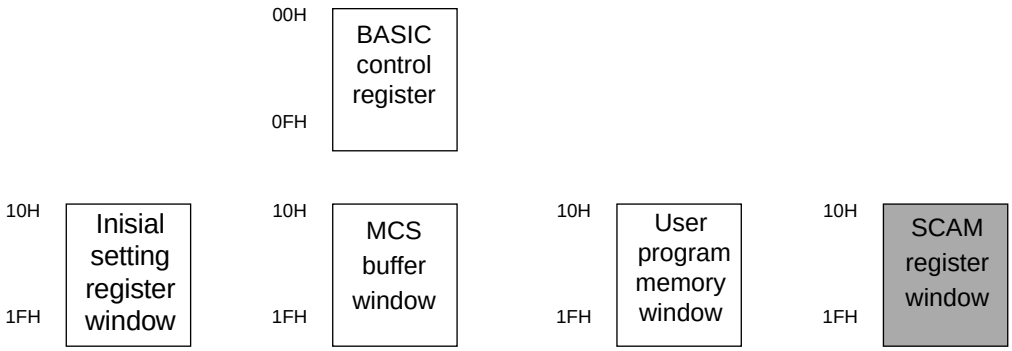
No	Address					Write	Read
	A4	A3	A2	A1	A0		
10	1	0	0	0	0	User program memory	User program memory
11	1	0	0	0	1	User program memory	User program memory
12	1	0	0	1	0	User program memory	User program memory
13	1	0	0	1	1	User program memory	User program memory
14	1	0	1	0	0	User program memory	User program memory
15	1	0	1	0	1	User program memory	User program memory
16	1	0	1	1	0	User program memory	User program memory
17	1	0	1	1	1	User program memory	User program memory
18	1	1	0	0	0	User program memory	User program memory
19	1	1	0	0	1	User program memory	User program memory
1A	1	1	0	1	0	User program memory	User program memory
1B	1	1	0	1	1	User program memory	User program memory
1C	1	1	1	0	0	User program memory	User program memory
1D	1	1	1	0	1	User program memory	User program memory
1E	1	1	1	1	0	User program memory	User program memory
1F	1	1	1	1	1	User program memory	User program memory

■ User Program Memory (READ/WRITE)

This memory is for reading and writing user program commands and data.

While the SPC is executing the user program, access to the user program memory and MCS buffers is denied.

3.6 SCAM REGISTER WINDOW



No	Address					Write	Read
	A4	A3	A2	A1	A0		
10	1	0	0	0	0	SCAM Data Bus Register	←
11	1	0	0	0	1	SCAM Control Register	←
12	1	0	0	1	0	(Reserved)	(Reserved)
13	1	0	0	1	1	(Reserved)	(Reserved)
14	1	0	1	0	0	(Reserved)	(Reserved)
15	1	0	1	0	1	(Reserved)	(Reserved)
16	1	0	1	1	0	(Reserved)	(Reserved)
17	1	0	1	1	1	(Reserved)	(Reserved)
18	1	1	0	0	0	(Reserved)	(Reserved)
19	1	1	0	0	1	(Reserved)	(Reserved)
1A	1	1	0	1	0	(Reserved)	(Reserved)
1B	1	1	0	1	1	(Reserved)	(Reserved)
1C	1	1	1	0	0	(Reserved)	(Reserved)
1D	1	1	1	0	1	(Reserved)	(Reserved)
1E	1	1	1	1	0	(Reserved)	(Reserved)
1F	1	1	1	1	1	(Reserved)	(Reserved)

SCAM register window is available in the MB86604L only.

3.6.1 SCAM REGISTER

The SCAM Register is initialized by the external reset or software reset command.

(1) SCAM Data Bus Register

Bank	Address		R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In									
01	10h		R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Initial Value			'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

This register is used to directly access the SCSI data bus from the system side for execution of the SCAM protocol. The signal level is reserved from those in the SCSI bus (writing 1 outputs low on the SCSI data bus).

(2) SCAM Control Register

Bank	Address		R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In									
01	11h		'0'	'0'	'0'	'0'	SEL	BSY	MSG	C/D	I/O
Initial Value			'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

This register is used to directly access the SCSI control signals from the system side for execution of the SCAM protocol. The signal level is reversed from those in the SCSI bus (writing 1 outputs low on the SCSI data bus).

Note:

1. When executing the SCAM protocol, set the SCSI driver to the open-drain mode. As for reading the SCSI bus, first write 0 to negate the SCSI driver outputs (Hi-Z) and then, read the SCSI bus.

2. When returning from SCAM protocol to the normal operation, make sure to initialize the SCAM data bus and SCAM control registers by writing 0 at all bits or executing the software reset command.

CHAPTER 4 COMMANDS

This chapter explains the various commands used by the SPC.

4.1 SPC COMMANDS

SPC commands can be grouped into four different types.

Initiator commands: Commands to establish the SPC as the initiator and commands for the initiator.

Target commands: Commands to establish the SPC as the target and commands for the target.

Common commands: Commands for the initiator or the target and commands for the operating devices.

Programmable commands: Commands used when the SPC is being run by a user program.

In addition, depending on how these commands are assigned, they can be segregated further as sequential commands, discrete commands, or special commands.

■ SPC Command Types Based on the Assignment Method

SPC commands can be assigned to the command register or to user program memory. Depending on the assignment method, these commands can be separated into the following three types.

Sequential commands: Command which conducts a sequence which is to some extent consecutive (including phase transitions). This type of command can only be assigned to the command register. It cannot be assigned to user program memory.

Discrete commands: Command which conducts an operation that disassembles a sequential command. This type of command can be assigned to either the command register or to user program memory.

Special commands: Command which is used when the SPC is run by a user program. This type of command can only be assigned to user program memory.

Table 4.1 shows a list of SPC command types based on assignment method differences.

Table 4.1 List of Command Types Based on Assignment Method Differences

Assignment method / Command type	Command register assignment	User program memory assignment
Sequential command	○ (1 byte instruction)	×
Discrete command	○ (1 byte instruction)	○ (1 or 2 byte instruction)
Special command	×	○ (1 or 2 byte instruction)

○: Can be specified ×: Cannot be specified

4.2 LIST OF INITIATOR COMMANDS

Table 4-2 shows a list of initiator commands. See appendix C for more on the step codes of each command.

■ List of Initiator Commands

Table 4.2 List of Initiator Commands

	No.	Command code									Operands (for programs)	Command name
Sequential	1	00_H	0	0	0	0	0	0	0	0	(Impossible)	SELECT & CMD
	2	01_H	0	0	0	0	0	0	0	1	(Impossible)	SELECT & 1-MSG & CMD
	3	02_H	0	0	0	0	0	0	1	0	(Impossible)	SELECT & N-Byte-MSG & CMD
	4	03_H	0	0	0	0	0	0	1	1	(Impossible)	SELCT & 1-MSG
	5	04_H	0	0	0	0	0	1	0	0	(Impossible)	SELCT & N-Nyte-MSG
	6	05_H	0	0	0	0	0	1	0	1	(Impossible)	SEND N-Byte-MSG
	7	06_H	0	0	0	0	0	1	1	0	(Impossible)	SEND N-Byte-CMD
	8	07_H	0	0	0	0	0	1	1	1	(Impossible)	RECEIVE N-Byte-MSG
Discrete	9	08_H	0	0	0	0	1	0	0	0	—	SELECT
	10	09_H	0	0	0	0	1	0	0	1	—	SELECT with ATN
	11	$0A_H$	0	0	0	0	1	0	1	0	—	SET ANT
	12	$0B_H$	0	0	0	0	1	0	1	1	—	RESET ANT
	13	$0C_H$	0	0	0	0	1	1	0	0	—	SET ACK
	14	$0D_H$	0	0	0	0	1	1	0	1	—	RESET ACK
	15	10_H	0	0	0	1	0	0	0	0	—	SEND DATA from MPU
	16	11_H	0	0	0	1	0	0	0	1	—	SENT DATA from DMA
	17	12_H	0	0	0	1	0	0	1	0	—	RECEIVE DATA from MPU
	18	13_H	0	0	0	1	0	0	1	1	—	RECEIVE DATA from DMA
	19	14_H	0	0	0	1	0	1	0	0	—	SEND DATA from MPU (Padding)
	20	15_H	0	0	0	1	0	1	0	1	—	SEND DATA from DMA (Padding)
	21	16_H	0	0	0	1	0	1	1	0	—	RECEIVE DATA from MPU (Padding)
	22	17_H	0	0	0	1	0	1	1	1	—	RECEIVE DATA from DMA (Padding)
	23	18_H	0	0	0	1	1	0	0	0	Address of message to be sent	SEND 1-MSG
	24	19_H	0	0	0	1	1	0	0	1	Address of message to be sent	SEND 1-MSG with ATN
	25	$1A_H$	0	0	0	1	1	0	1	0	Write address of message	RECEIVE MSG
	26	$1B_H$	0	0	0	1	1	0	1	1	Address of command be sent	SEND CMD
	27	$1C_H$	0	0	0	1	1	1	0	0	Write address of status	RECEIVE STATUS

4.2.1 List of Initiator Sequential Command Functions

Table 4-3 shows a list of initiator sequential command names and their functions.

■ **List of Initiator Sequential Command Functions**

Table 4.2.1 List of Initiator Sequential Command Functions

No.	Command name	Function	Initial settings
1	SELECT & CMD	• Perform selection • Send CDB*	• Write the target ID number in the SEL/RESEL ID register • Write the CDB in the SEND MCS buffer
2	SELECT & 1-MSG & CMD	• Perform selection • Send a single message • Send CDB	• Write the target ID number in the SEL/RESEL ID register • Write the message and then the CDB in the SEND MCS buffer
3	SELECT & N-Byte-MSG & CMD	• Perform selection • Send an N-byte message • Send CDB	• Write the target ID number in the SEL/RESEL ID register • Write the message and then the CDB in the SEND MCS buffer • Write the number of message bytes in the MC byte register
4	SELECT & 1-MSG	• Perform selection • Send a single message	• Write the target ID number in the SEL/RESEL ID register • Write the message in the SEND MCS buffer
5	SELECT & N-Byte-MSG	• Perform selection • Send an N-byte message (used for sending two or more messages)	• Write the target ID number in the SEL/RESEL ID register • Write the message in the SEND MCS buffer • Write the number of message bytes in the MC byte register
6	SEND N-Byte-MSG	Send an N-byte message (used for sending two or more messages)	• Write the message in the SEND MCS buffer • Write the number of message bytes in the MC byte register
7	SEND N-Byte-CMD	Send an N-byte CDB (used for sending an undefined CDB)	• Write the CDB in the SEND MCS buffer • Write the number of CDB bytes in the MC byte register
8	RECEIVE N-Byte-MSG	Receive an N-byte message (used for receiving a message which is 33 bytes or more in length)	• Write the number of message bytes in the MC byte register

CDB: Command descriptor block

* In the manual ACK Reset mode, the transfer completion is reported with asserting the final ACK signal. So, negating the ACK signal is required before issuing a command for the next phase using "RESET ACK" command. (See section 3.3.13.) Also, note that "02_H" or more values must be set in the MC Byte Register if the command requiring the register setting is issued.

MEMO

4.2.2 List of Initiator Discrete Command Functions

Table 4.2.2 shows a list of initiator discrete command names and their functions.

■ List of Initiator Discrete Command Functions

Table 4.2.2 List of Initiator Discrete Command Function

No.	Command name	Function	Initial settings
9	SELECT	Perform selection	Write ID number to the SEL/RESEL ID register
10	SELECT with ATN	Perform selection and assert ATN signal	Write ID number to the SEL/RESEL ID register
11	SET ATN	Assert ATN signa	None
12	RESET ATN	Negate ATN signal	None
13	SET ACK	Assert ACK signal	None
14	RESET ACK	Negate ACK signal	None
15	SEND DATA from MPU	Send data by the program transfer mode *1	Write the number of data blocks and the number of bytes sent to the data block register and the data byte register.
16	SEND DATA from DMA	Send data by the DMA transfer mode *1	Write the number of data blocks and the number of bytes sent to the data block register and the data byte register.
17	RECEIVE DATA to MPU	Receive data by the program transfer mode *1	Write the number of data blocks and the number of bytes received to the data block register and the data byte register.
18	RECEIVE DATA to DMA	Receive data by the DMA transfer mode *1	Write the number of data blocks and the number of bytes received to the data block register and the data byte register.
19	SEND DATA from MPU (PADDING)	Send the counter amount of data by the program transfer mode and perform PADDING transfer until the target changes phase *2	Write the number of data blocks and the number of bytes sent to the data block register and the data byte register.
20	SEND DATA from DMA (PADDING)	Send the counter amount of data by the DMA transfer mode and perform PADDING transfer until the target changes phase *2	Write the number of data blocks and the number of bytes sent to the data block register and the data byte register.
21	RECEIVE DATA to MPU (PADDING)	Receive the counter amount of data by the program transfer mode and discard data being received until the target changes phase *2	Write the number of data blocks and the number of bytes received to the data block register and the data byte register.

Table 4.2.2 List of Initiator Discrete Command Functions *(Continued)*

No.	Command name	Function	Initial settings
22	RECEIVE DATA to DMA (PADDING)	Receive the counter amount of data by the DMA transfer mode and discard data being received until the target changes phase *2	Write the number of data blocks and the number of bytes received to the data block register and the data byte register.
23	SEND 1-MSG	Send a single message *1	Write the message to the SEND MCS buffer
24	SEND 1-MSG with ATN	Send a single message while continuing to assert the ATN signal *2	Write the message to the SEND MCS buffer
25	RECEIVE MSG	Send a single message *1	None
26	SEND CMD	Send a single CDB *1	Write the CDB to the SEND MCS buffer
27	RECEIVE STATUS	Receive a single status *1	None

CDB: Command descriptor block

*1 In the manual ACK Reset mode, the transfer completion is reported with asserting the final ACK signal. So, negating the ACK signal is required before issuing a command for the next phase using "Reset ACK" command. (See section 3.3.13)

*2 Command is complete negating the final ACK signal regardless of the ACK Reset mode setting.

■ **Starting and Stopping PADDING Transfers**

When the transfer of the number of data bytes specified by the data block register and the data byte register is complete, the SPC starts a PADDING transfer.

In cases where the data block register and the data byte register initial values are set to 0, the PADDING transfer starts from the first byte.

PADDING transfer continues until the target changes phase.

4.3 LIST OF TARGET COMMANDS

Table 4-5 shows a list of target commands.

See appendix D for more on the step codes of each command.

■ List of Target Commands

Table 4.3 List of Target Commands

	No.	Command code									Operands (for programs)	Command name
Sequential	1	20_H	0	0	1	0	0	0	0	0	(Impossible)	RESELECT & 1-MSG
	2	21_H	0	0	1	0	0	0	0	1	(Impossible)	RESELECT & N-BYTE-MSG
	3	22_H	0	0	1	0	0	0	1	0	(Impossible)	RESELECT & 1-MSG TERMINATE
	4	23_H	0	0	1	0	0	0	1	1	(Impossible)	RESELECT & 1-MSG & LINK TERMINATE
	5	24_H	0	0	1	0	0	1	0	0	(Impossible)	TERMINATE
	6	25_H	0	0	1	0	0	1	0	1	(Impossible)	LINK TERMINATE
	7	26_H	0	0	1	0	0	1	1	0	(Impossible)	DISCONNECT SEQUENCE
	8	27_H	0	0	1	0	0	1	1	1	(Impossible)	SEND N-BYTE-MSG
	9	28_H	0	0	1	0	1	0	0	0	(Impossible)	RECEIVE N-BYTE-CMD
	10	29_H	0	0	1	0	1	0	0	1	(Impossible)	RECEIVE N-BYTE-MSG
	11	$2A_H$	0	0	1	0	1	0	1	0	(Impossible)	RESELECT & N-MSG TERMINATE
	12	$2B_H$	0	0	1	0	1	0	1	1	(Impossible)	RESELECT & N-MSG & LINK TERMINATE
	13	$2C_H$	0	0	1	0	1	1	0	0	(Impossible)	DISCONNECT SEQUENCE 2
Discrete	14	30_H	0	0	1	1	0	0	0	0	—	RESELECT
	15	31_H	0	0	1	1	0	0	0	1	—	SET 'REQ'
	16	32_H	0	0	1	1	0	0	1	0	—	RESET 'REQ'
	17	33_H	0	0	1	1	0	0	1	1	—	DISCONNECT
	18	34_H	0	0	1	1	0	1	0	0	—	SEND DATA FROM MPU
	19	35_H	0	0	1	1	0	1	0	1	—	SEND DATA FROM DMA
	20	36_H	0	0	1	1	0	1	1	0	—	RECEIVE DATA FROM MPU
	21	37_H	0	0	1	1	0	1	1	1	—	RECEIVE DATA FROM DMA
	22	38_H	0	0	1	1	1	0	0	0	Address of message to be sent	SEND 1-MSG
	23	39_H	0	0	1	1	1	0	0	1	Write address of message	RECEIVE MSG
	24	$3A_H$	0	0	1	1	1	0	1	0	Address of status to be sent	SEND STATUS
	25	$3B_H$	0	0	1	1	1	0	1	1	Write address of CDB	RECEIVE CMD

MEMO

4.3.1 List of Target Sequential Command Functions

Table 4-6 shows a list of target sequential command names and their functions.

■ List of Target Sequential Command Functions

Table 4.3.1 List of Target Sequential Command Functions

No.	Command name	Function	Initial settings
1	RESELECT & 1-MSG	• Performs reselection • Moves to message in phase • Sends a single message	• Write the initiator ID number in the SEL/RESEL ID register • Write the message in the SEND MCS buffer
2	RESELECT & N-Byte-MSG	• Performs reselection • Moves to message in phase • Sends an N-byte message	• Write the initiator ID number in the SEL/RESEL ID register • Write the message in the SEND MCS buffer • Set the number of message bytes in the MC byte register
3	RESELECT & 1-MSG & TERMINATE	• Performs reselection • Moves to message in phase • Sends a single message • Performs TERMINATE sequence	• Write the initiator ID number in the SEL/RESEL ID register • Write the message, status (1 byte), and then the next message (1 byte) in the SEND MCS buffer
4	RESELECT & 1-MSG & LINK TERMINATE	• Performs reselection • Moves to message in phase • Sends a single message • Performs LINK TERMINATE sequence	• Write the initiator ID number in the SEL/RESEL ID register • Write the message, status (1 byte), and then the next message (1 byte) in the SEND MCS buffer
5	TERMINATE	• Moves to status phase • Sends a 1-byte status • Moves to message in phase • Sends a 1-byte message • Disconnects	• Write the status (1 byte) and then the message (1 byte) in the SEND MCS buffer
6	LINK TERMINATE	• Moves to status phase • Sends a 1-byte status • Moves to message in phase • Sends a 1-byte message	• Write the status (1 byte) and then the message (1 byte) in the SEND MCS buffer
7	DISCONNECT SEQUENCE	• Moves to message in phase • Sends a 2-byte message • Disconnects	• Write the message (2 bytes) in the SEND MCS buffer
8	SEND N-Byte-MSG	• Moves to message in phase • Send an N-byte message (used for sending two or more messages)	• Write the message in the SEND MCS buffer • Write the number of message bytes in the MC byte register

CDB: Command descriptor block

Table 4.3.1 List of Target Sequential Command Functions *(Continued)*

No.	Command name	Function	Initial settings
9	RECIEVE N-Byte-CMD	• Move to command phase • Receive an N-byte CDB (used for receiving an undefined CDB)	• Write the number of CDB bytes in the MC byte register
10	RECEIVE N-Byte-MSG	• Move to message out phase • Receive an N-byte message (used for receiving a message which is 33 bytes or more in length)	• Write the number of message bytes in the MC byte register
11	RESELECT & N-Byte-MSG & TEMINATE	• Performs reselection • Moves to message in phase • Sends an N-byte message • Performs TERMINATE sequence	• Write the initiator ID number in the SEL/RESEL ID register • Write the message, status (1 byte), and then the next message (1 byte) in the SEND MCS buffer • Set the number of bytes of the first message in the MC byte register
12	RESELECT & N-Byte-MSG & TERMINATE	• Performs reselection • Moves to message in phase • Sends an N-byte message • Performs LINK TERMINATE sequence	• Write the initiator ID number in the SEL/RESEL ID register • Write the message, status (1 byte), and then the next message (1 byte) in the SEND MCS buffer • Set the number of bytes of the first message in the MC byte register
13	DISCONNECT SEQUENCE 2	• Moves to message in phase • Sends a 1-byte message • Disconnects	• Write the message (1 byte) in the SEND MCS buffer

Note: Set "02_H" or more value in the MC Byte Register if a command requiring for the register setting is issued. (Do not set "01_H".)

4.3.2 List of Target Discrete Command Functions

Table 4-7 shows a list of target discrete command names and their functions.

■ List of Target Discrete Command Functions

Table 4.3.2 List of Target Discrete Command Functions

No.	Command name	Function	Initial settings
14	RESELECT	Perform reselection	Write the initiator ID number in the SEL/RESEL ID register
15	SET REQ	Assert REQ signal	None
16	RESET REQ	Negate REQ signal	None
17	DISCONNECT	Release BSY signal	None
18	SEND DATA from MPU	•Move to data in phase • Send data by the program transfer mode	Write the number of blocks and the number of bytes of the data sent to the data block register and the data byte register
19	SEND DATA from DMA	• Move to data in phase • Receive data by the DMA transfer mode	Write the number of blocks and the number of bytes of the data sent to the data block register and the data byte register
20	RECEIVE DATA from MPU	• Move to data out phase • Receive data by the program transfer mode	Write the number of blocks and the number of bytes of the data received to the data block register and the data byte register
21	RECIEVE DATA from DMA	• Move to data out phase • Send data by the DMA transfer mode	Write the number of blocks and the number of bytes of the data received to the data block register and the data byte register
22	SEND 1-MSG	• Move to message in phase • Send a single message	Write the message to the SEND MCS buffer
23	RECEIVE MSG	• Move to message out phase • Receive a single message	None
24	SEND STATUS	• Move to status phase • Send a 1-byte status	Write the status (1 byte) to the SEND MCS buffer
25	RECEIVE CMD	• Move to command phase • Receive CDB	None

CDB: Command descriptor block

4.4 LIST OF COMMON COMMANDS

Table 4-8 shows a list of common commands.

■ List of Common Commands

Table 4.4 List of Common Commands

No.	Command code									Operands (for programs)	Command name
1	40_H	0	1	0	0	0	0	0	0	(Impossible)	SEFTWARE RESET
2	41_H	0	1	0	0	0	0	0	1	(Impossible)	TRANSFER RESET
3	42_H	0	1	0	0	0	0	1	0	(Impossible)	SCSI RESET
4	43_H	0	1	0	0	0	0	1	1	(Impossible)	SET UP REG
5	44_H	0	1	0	0	0	1	0	0	(Impossible)	INIT DIAG START
6	45_H	0	1	0	0	0	1	0	1	(Impossible)	TARG DIAG START
7	46_H	0	1	0	0	0	1	1	0	(Impossible)	DIAG END
8	47_H	0	1	0	0	0	1	1	1	(Impossible)	COMMAND PAUSE
9	48_H	0	1	0	0	1	0	0	0	(Impossible)	SET RST
10	49_H	0	1	0	0	1	0	0	1	(Impossible)	RESET RST

4.4.1 List of Common Command Functions

Table 4-9 shows a list of common command names and their functions.

■ List of Common Command Functions

Table 4.4.1 List of Common Command Functions

No.	Command name	Function	Initial settings
1	SOFTWARE RESET	• Out the SPC internal circuits in a reset state • Move the SPC internal circuits from a reset state to a READY state	None
2	TRANSFER RESET	Put the SPC internal circuits related to transfers in a reset state	None
3	SCSI RESET	Send the RST signal which is a SCSI bus signal	None
4	SET UP REG	Make the initial settings valid	None
5	INIT DIAG START	Enables the SPC to perform self diagnosis and simulate transfers as the initiator	None
6	TRAG DIAG START	Enables the SPC to perform self diagnosis and simulate transfers as the target	None
7	DIAG END	Move from the self diagnosis mode to the standard mode	None
8	COMMAND PAUSE	• Force a transfer to stop during execution of the data phase • Suspend selection/reselection	None
9	SET RST	• Send SCSI RST signal and keep it	None
10	RESET RST	• Make reset SCSI RST signal set by SET RST command	None

■ Software Reset Command

When the SOFTWARE RESET command is sent, a Command Complete interrupt is not produced. The SPC comes to a full stop.

In order to remove the software reset state, it is necessary to send another SOFTWARE RESET command (move to Ready state).

■ Self Diagnosis

The SPC performs a self diagnosis when the INIT DIAG START command or the TARG DIAG START command is sent.

Self diagnosis is performed for the following items and after completion, the self diagnosis results are reported.

- Arbitration sequence
- Selection/reselection sequence

  Also, depending on the command sent, the self diagnosis can be followed by a simulation of initiator or target transfers.

- When the INIT DIAG START command is sent, the SPC can simulate a transfer as the initiator.
- When the TARG DIAG START command is sent, the SPC can simulate a transfer as the target.

See 5.10 for more on this topic.

■ Command Pause Command

- The transfer can be forced to stop only during the data phase.

  For SCSI Output

  During a DMA transfer, the DREQ signal output is immediately suspended, and during a program transfer, 0 is shown in the DATA REQ bit (SPC status register: bit 2).

  Transfers to the SCSI bus (data output) continue until the data register is EMPTY.

  For SCSI Input

  New REQ signal and ACK signal issues to the SCSI bus are suspended.

  MPU, DMA side transfers (DMA transfers, program transfers) are performed until the data register is EMPTY.

  As the target, the REQ/ACK handshake is fully complete, but as the initiator, it is possible to receive additional REQ signals.

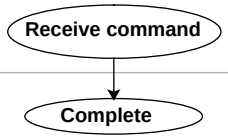
- Suspension of selection/reselection is only possible prior to execution of the selection/ reselection phase.
  When the SPC has already started selection/reselection, the COMMAND PAUSE command is invalid and selection/reselection continues being executed. ("Command Rejected" interrupt is not generated.)
- After the COMMAND PAUSE command is received and SPC completed the process, a COMMAND PAUSE interrupt is reported.

4.4.2 Common Command Step Codes

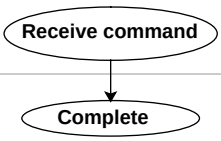
The common command step codes are shown below.

■ **Common Command Step codes**

- No.2 TRANSFER RESET
- No.3 SCSI RESET
- No.4 SET UP REG
- No.5 INIT DIAG START
- No.6 TARG DIAG START
- No.7 DIAG END
- No.9 SET RST
- No.10 RESET RST

Flowchart	Step	Discription
	0 1	* Receive command Command Complete

- No.8 COMMAND PAUSE

Flowchart	Step	Discription
	0 1	* Receive command Command processing COMMAND PAUSE

MEMO

4.5 PROGRAMMABLE COMMANDS

The user program is stored ahead of time in the user program memory, and operation can be started by writing the user program memory head address in the command register.

Programmable commands can be either discrete commands or special commands and have a command length of 1 byte or 2 bytes.

■ Programmable Commands

Table 4.5 shows a list of programmable command field assignments.

Table 4.5 List of Programmable Command Filed Assignments

	Command code (1st byte)	Operands (2nd byte)
Discrete commands	Commands for sending messages, commands, and status phases	Memory address where data to be sent resides
	Commands for receiving messages, commands, and status phases	Memory address where data to be received will be stored
	Commands for receiving or sending data phases Commands which do not perform transfers	—
Special commands	**AND** command	Data for AND calculation or memory address where data for AND calculation resides
	TEST AND command	Data for AND calculation or memory address where data for AND calculation resides
	COMPARE command	Data for COMPARE calculation or memory address where data for COMPARE calculation resides
	Conditional branch command	Jump head address
	MOVE command	Memory address for MOVE
	STOP command	User status code
	NOP command	—

■ Points to Watch Creating a User Program

Pay attention to the following points when creating a user program.

• Points to watch when using an AND command, TEST AND command, COMPARE command, or MOVE command

When the interrupt status register and the command step register are specified by an **AND** command, **TEST AND** command, **COMPARE** command, or **MOVE** command (register memory case only), since the interrupt register and the command register are FIFO type registers, when multiple causes are being held, the SPC will display the next interrupt status or command step.
When multiple **AND** commands directed at the above-mentioned registers are executed, the register content must be stored by a **MOVE** command (register memory). By using the **MOVE** command (memory accumulator) in this way, it is possible to execute multiple **AND** commands.
Also, when multiple **TEST AND** commands or **COMPARE** commands are executed, specify the accumulator register from the second command (the **TEST AND** command or **COMPARE** command executes the command after loading the specified register's content in the accumulator and accumulator content does not change).

• Automatic mode during user program operation

The Automatic receive mode is invalid.
The Automatic selection/reselection response mode is valid, and this mode is given priority over user program operations.
When the SPC is selected/reselected while a discrete command which includes the user program's own Select/Reselect is being executed, the user program operation is suspended and the SPC moves to Automatic selection/reselection. At this point, the user program is ignored.
Even when the Automatic selection/reselection response mode is not established, once the SPC is selected/reselected, the user program operation is suspended and the SPC responds to the select/reselect.

• Interrupt during user program operation

When an interrupt cause such as an error occurs during user program operation, the SPC response varies depending on the interrupt cause level.

Level 1: The user program operation is suspended, and an interrupt is sent to the host MPU. After that, the SPC switches to READY state.

Level 2: The user program operation is maintained, and an interrupt is not sent to the host MPU. The interrupt status is stored in the status register and becomes subject to the calculation command.

However, when Level 2 interrupt report mode is disabling (not report), the interrupt status is not stored in the status register.
Level 2 interrupts are indicated below. All other interrupts are treated as level 1 interrupts.

Interrupt type	Operation mode	Interrupt content
REPORT	INIT/TARG	Command completed
	TARG	ATN condition detected
Phase error	INIT	Initial phase error

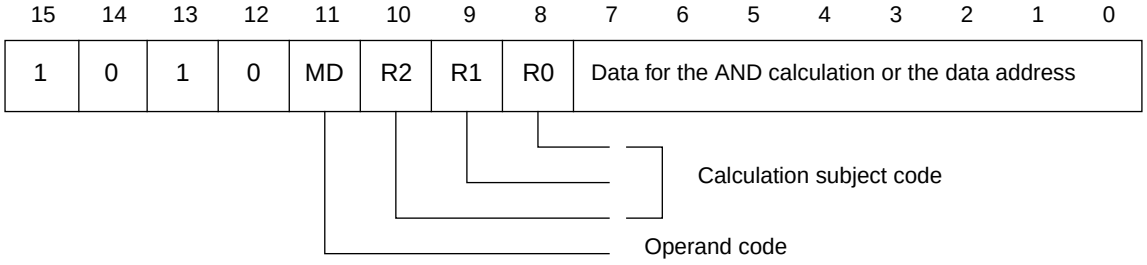
4.5.1 AND Command

The AND command loads the values indicated by the calculation subject code in the accumulator and performs a logical product calculation between the values shown in the accumulator and the operand.
The calculation result is placed in the accumulator.

In the logical product calculation, when both bits values in the calculation are 1, the calculation result is 1. On the other hand, when one value is 0 and the other 1 or when both values are 0, the calculation result is 0.

■ **AND Command**

The **AND** command bit configuration is shown below.



Next, the functionality of each bit is described.

BIT 1: Operand code

This bit specifies the meaning of the operand.

1: The address where the logical product subject data resides is placed in the operand.
0: The logical product subject data is placed in the operand.

BIT 10 to 8: Calculation subject code

This bit specifies the values for which the logical product calculation will be performed. The relationship between these bit settings and the registers which are used in the logical product calculation is shown in Table 4.5.1.

Table 4.5.1 Relationship between AND Command Bit Settings (BIT 10 to 8) and the Values Used in the Logical Product Calculation

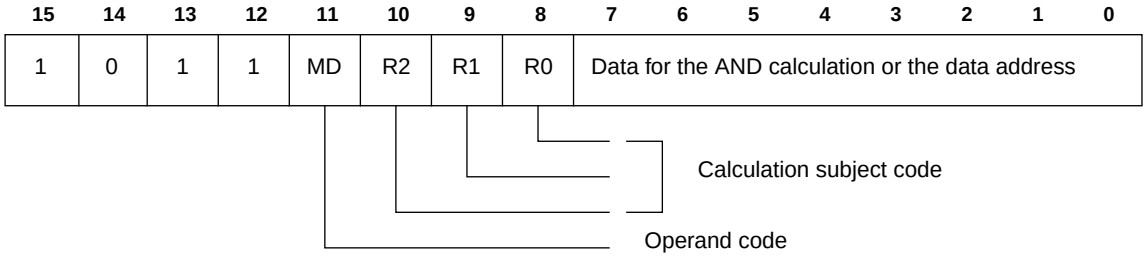
R2	R1	R0	Logical product calculation values
0	0	0	1st byte received in message, command, or status phase
0	0	1	2nd byte received in message, command, or status phase
0	1	0	SPC status register
0	1	1	Nexus status register
1	0	0	Interrupt status register
1	0	1	Command step register
1	1	0	SCSI control signal status register
1	1	1	Accumulator

4.5.2 TEST AND Command

The TEST AND command loads the values indicated by the calculation subject code in the accumulator and performs a logical product calculation between the values shown in the accumulator and the operand.

The calculation result is stored in the SPC and can be used by the conditional branch command. The accumulator value does not change.

■ **TEST AND Command**



Next, the functionality of each bit is described.

BIT 1: Operand code

This bit specifies the meaning of the operand.

1: The address where the logical product subject data resides is placed in the operand.
0: The logical product subject data is placed in the operand.

BIT 10 to 8: Calculation subject code

This bit specifies the values for which the logical product calculation will be performed. The relationship between these bit settings and the registers which are used in the logical product calculation is shown in Table 4.5.2.

Table 4.5.2 Relationship between TEST AND Command Bit Settings (BIT 10 to 8) and the Values Used in the Logical Product Calculation

R2	R1	R0	Logical product calculation values
0	0	0	1st byte received in message, command, or status phase
0	0	1	2nd byte received in message, command, or status phase
0	1	0	SPC status register
0	1	1	Nexus status register
1	0	0	Interrupt status register
1	0	1	Command step register
1	1	0	SCSI control signal status register
1	1	1	Accumulator

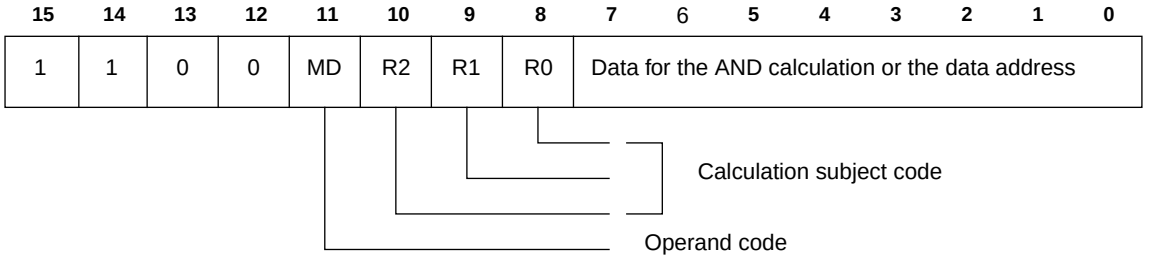
4.5.3 COMPARE Command

The COMPARE command loads the values indicated by the calculation subject code in the accumulator and performs a comparison calculation between the values shown in the accumulator and the operand.

The calculation result is stored in the SPC and can be used by the conditional branch command. The accumulator value does not change.

■ COMPARE Command

The **COMPARE** command bit configuration is shown below.



Next, the functionality of each bit is described.

BIT 11: Operand code

This bit specifies the meaning of the operand.

1: The address where the logical product subject data resides is placed in the operand.
0: The logical product subject data is placed in the operand.

BIT 10 to 8: Calculation subject code

This bit specifies the values for which the comparison calculation will be performed. The relationship between these bit settings and the registers which are used in the comparison calculation is shown in Table 4.5.3.

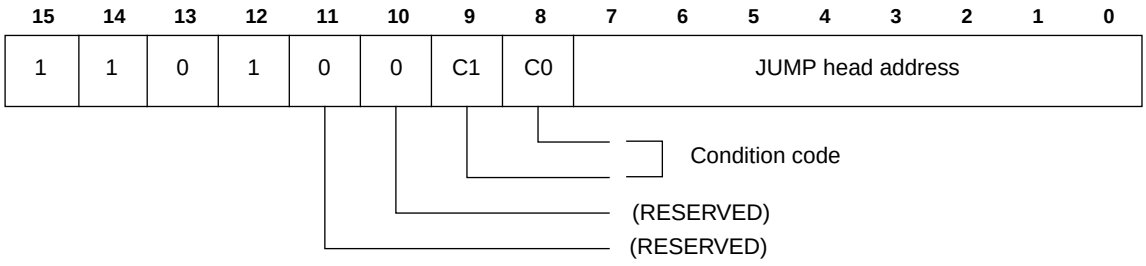
Table 4.5.3 Relationship between COMPARE Command Bit Settings (BIT 10 to 8) and the Values Used in the Comparison Calculation

R2	R1	R0	Comparison calculation values
0	0	0	1st byte received in message, command, or status phase
0	0	1	2nd byte received in message, command, or status phase
0	1	0	SPC status register
0	1	1	Nexus status register
1	0	0	Interrupt status register
1	0	1	Command step register
1	1	0	SCSI control signal status register
1	1	1	Accumulator

4.5.4 Conditional Branch Command

The conditional branch command executes conditional branch operations that utilize COMPARE command or TEST AND command results as well as unconditional branch operations.

■ Conditional Branch Command



The conditional branch command bit configuration is shown below.

BIT 9 and 8: Condition code

This bit specifies the branch condition.

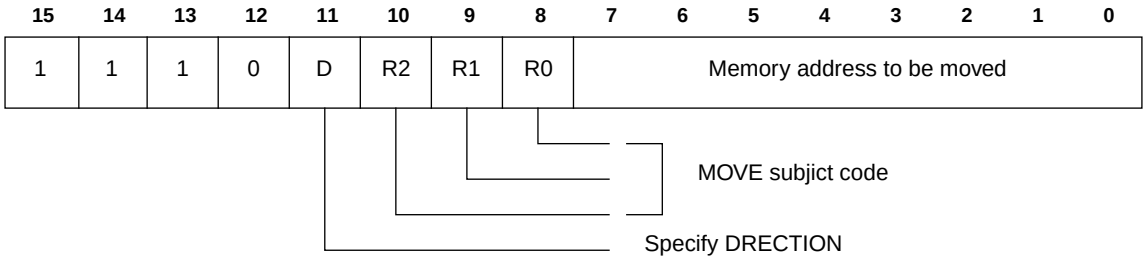
C1	C2	Operation
0	1	When the calculation result is not 0, execution control moves to the address indicated by the operand, and when the calculation result is 0, the next command is executed (conditional branch).
1	0	When the calculation result is 0, execution control moves to the address indicated by the operand, and when the calculation result is not 0, the next command is executed (conditional branch).
0	0	Execution control moves to the address indicated by the operand unconditionally (unconditional branch).
1	1	Undefined

4.5.5 MOVE Command

The MOVE command is used to copy the values specified by the MOVE subject code to the memory address specified by the operand or to copy the memory address values specified by the operand to the accumulator.

■ MOVE Command

The **MOVE** command bit configuration is shown below.



Next, the functionality of each bit is described.

BIT 11: Specify DIRECTION

This bit specifies the direction in which the data is moved.

1: The values specified by the MOVE subject code are copied to the address specified by the operand (register → memory)

0: The address value specified by the operand is copied to the accumulator (memory → accumulator).

BIT 10 to 8: MOVE subject code

This bit specifies the values to be moved. The relationship between these bit settings and the values subject to the MOVE is shown in Table 4.5.5.

Table 4.5.5 Relationship between MOVE Command Bit Settings (BIT 10 to 8) and the Values Subject to the MOVE Operation

R2	R1	R0	MOVE operation calculation values
0	0	0	1st byte received in message, command, or status phase
0	0	1	2nd byte received in message, command, or status phase
0	1	0	SPC status register
0	1	1	Nexus status register
1	0	0	Interrupt status register
1	0	1	Command step register
1	1	0	SCSI control signal status register
1	1	1	Accumulator

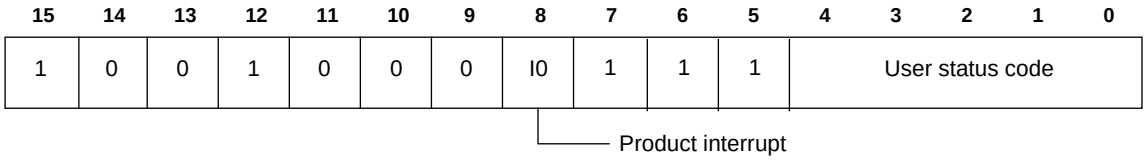
4.5.6 STOP Command

The STOP command is used to stop the user program operation.

When the execution of this command is completed, the SPC has moved from user program operation to standard operation (moves to an idle state and waits to receive commands from the command register).

■ STOP Command

The **STOP** command bit configuration is shown below.



Next, the functionality of each bit is described. BIT 7 to 5 are already set to 1 and cannot be changed.

BIT 8: Produce interrupt

This bit specifies whether an interrupt is sent to the host MPU.

1: Interrupt is sent.

0: Interrupt is not sent.

When there is an interrupt request to the host MPU from the SPC, BIT 7 of the SPC status register displays a 1 regardless of this bit setting.

BIT 4 to 0: User status code

This bit specifies the user status code.

4.5.7 NOP Command

The NOP command stands for NO OPERATION which means that nothing is executed and the SPC moves on to the next step command.

■ NOP Command

The **NOP** command bit configuration is shown below.

7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	0

CHAPTER 5 OPERATIONS

This chapter explains SPC operations.

5.1 SPC INITIALIZATION

The SPC is reset by the following procedures.

- **When the RESET pin input = L (hardware reset)**
- **Sending a SOFTWARE RESET command (software reset)**
- **Sending a TRANSFER RESET command (transfer reset)**
- **When a SCSI reset condition exists (reset condition)**
- **After switching on the power, be sure to perform a hardware reset.**

■ SPC Initialization

The SPC is reset by the following procedures.
After switching on the power, be sure to perform a hardware reset.

- Hardware reset
  All control circuits are reset. Make the $\overline{\text{RESET}}$ pin active for 4CLK or longer.
  During a hardware reset, it is not possible to write to the internal registers.
- Software reset
  All control circuits are reset.
  During a software reset, except for writing the **SOFTWARE RESET** command to the command register, it is not possible to send other commands or to write to other internal registers.

Just like the hardware reset, the internal register values are initialized. However, the initial setting register holds the register values established before the **SOFTWARE RESET** command was issued and by sending the SET UP REG command again, the register values established before the **SOFTWARE RESET** command was issued are reinstated.

Also, the MCS buffer and user program memory are not initialized, retaining their values.

- Transfer reset
  The SPC data transfer circuits are reset.
  The SCSI connection state does not change. Also, the selection/reselection phase is not affected.

When the SPC is functioning as the initiator, if the **TRANSFER RESET** command is sent at the same time as a target data transfer (input) starts, only the first byte is received (only for asynchronous transfers).

- Reset Condition
  When the SPC is connected, output from all SCSI pins is suspended.
  The internal registers are not initialized.

MEMO

5.2 INITIAL SETTINGS FOR SPC INTERNAL REGISTERS

After switching on the power, some of the internal registers remain undefined.

Once the hardware reset or software reset is removed, establish the initial settings for the SPC internal registers.

■ Initial Settings for SPC Internal Registers

Table 5.2a and Table 5.2b provide a list of the internal register states (BASIC control registers and initial settings registers) after a hardware reset or software reset.

Also, Figure 5.1 illustrates the initial setting flow after a hardware reset or software reset.

Table 5.2a Internal Register State After Hardware Reset or Software Reset (BASIC Control Registers)

ADR	Register	Initialization value								Read value							
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
00h	DATA	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
01h	DATA	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
02h	SPC STATUS	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	1
03h	NEXUS STATUS	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
04h	SELID/INT-STATUS	0	—	—	—	—	0	0	0	0	0	0	0	0	0	0	0
05h	CMD/CMD STEP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
06h	DATA BLOCK	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
07h	DATA BLOCK	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
08h	DATA BYTE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
09h	DATA BYTE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0Ah	DATA BYTE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0Bh	DIAG/SCSI CNTL	X	X	X	—	X	X	X	X	(Shows SCSI signal)							
0Ch	TRANS MODE	0	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
0Dh	TRANS PERIOD	—	—	—	0	0	0	0	0	0	0	0	0	0	0	0	0
0Eh	TRANS OFFSET	—	—	—	0	0	0	0	0	0	0	0	0	0	0	0	0
0Fh	WINDOW/MBC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Notes:

1 The transfer parameters stored by individual ID (transfer-mode/period/offset) are all initialized.
2 The value of X bits does not change after hardware reset or software reset. The value after switching on the power is undefined.
3 The transfer reset clears Bit 1 and Bit 0 of SPC status register and Bit 5 to Bit 0 of Modified-Byte-Register.

Table 5.2b Internal Register State After Hardware Reset or Software Reset (Initial Setting Registers)

ADR	Register	Initialization value								Read value							
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
10h	CLK-CNV	0	0	0	0	1	0	1	1	X	X	X	X	X	X	X	X
11h	OWN-ID	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X
12h	Response operation mode	0	0	X	0	0	0	0	X	X	X	X	X	X	X	X	X
13h	SEL/RESEL operation mode	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
14h	SEL/RESEL RETRY	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
15h	SEL/RESEL TOUT	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
16h	REQ/ACK TOUT	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X
17h	ASYNC SETUP-TIME	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X
18h	PARITY ENABLE	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X
19h	INT ENABLE	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X
1Ah	GROUP 6/7	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1Bh	DMA SYSTEM	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X
1Ch	Auto Operation mode	X	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1Dh	SPC TIMEOUT	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X
1Fh	Revision indication	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Note: The value of X bits does not change after hardware reset or software reset. The value after switching on the power is undefined. Registers values do not change by Transfer Reset.

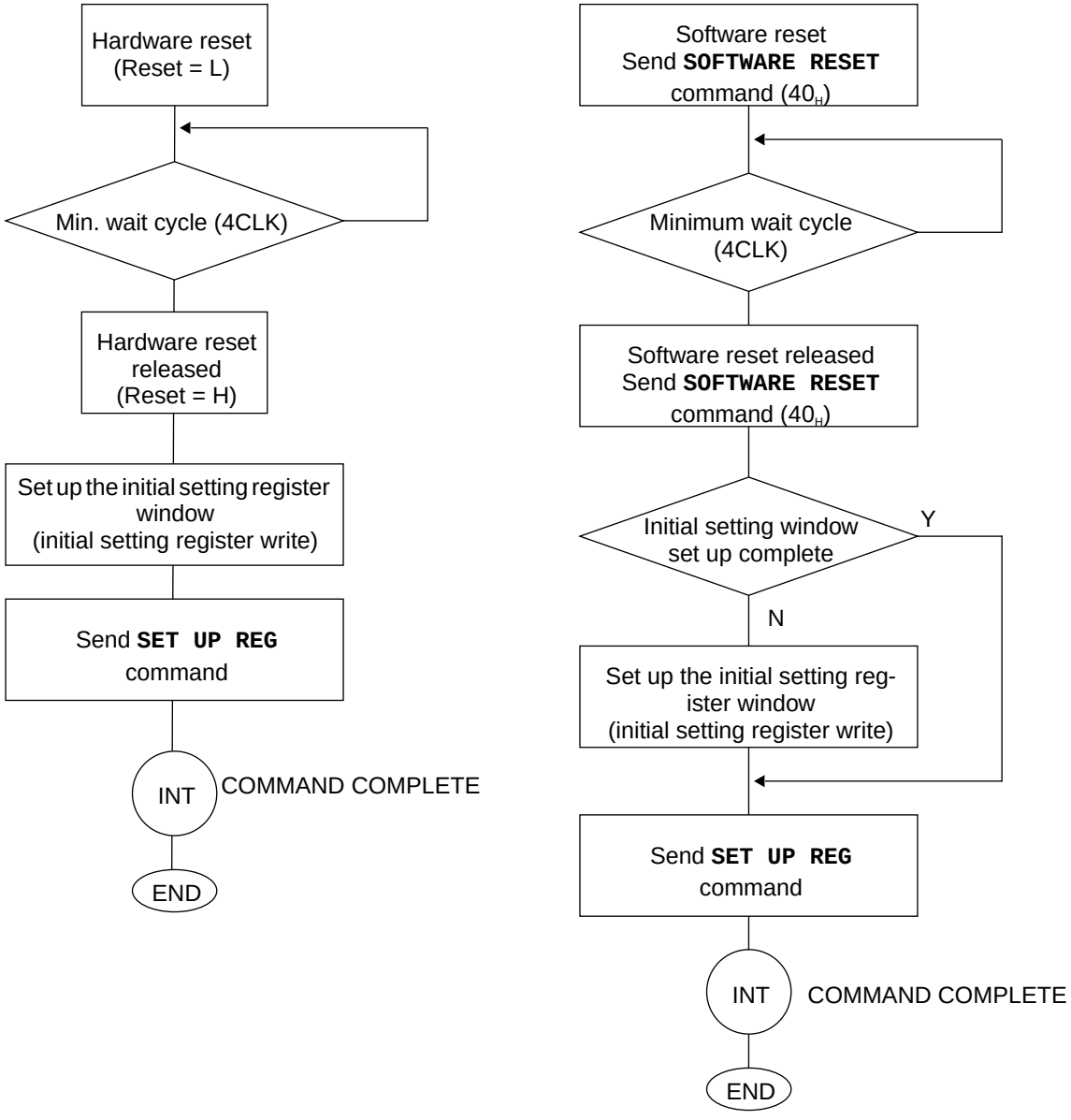


Figure 5.2 Initial Setting Flow After Hardware Reset or Software Reset

MEMO

5.3 SCSI BUS PHASE

When the SPC is selected/reselected by another bus device, a response sequence is automatically executed if the response conditions are fulfilled.

Also, there are three types of operations after the selection/reselection response.

- **Immediately after the selection/reselection, report a Selected, Selected with ATN, or Reselected interrupt.**
- **Execute the Automatic selection response mode or Automatic reselection response mode which are already set up in the SPC.**
- **Perform a user program operation.**

■ Selection from Another Bus Device

When the SPC is selected/reselected by another bus device, a response sequence is automatically executed if the response conditions are fulfilled.

(1) Selection phase response

a. BIT 6 of the Response mode setting register (selection response) is 1.

b. SCSI data bus data (ID data) satisfies the following conditions.

- The bit setting in the self ID setting register is ”true.”
- When BIT 1 of the Response mode setting register (single initiator option) is 1, there is 1 ”true” bit.
- When BIT 1 of the Response mode setting register (single initiator option) is 0, there are 2 ”true” bits.
- The parity bit is correct (a parity check is conducted regardless of BIT 7 of the parity error detect setting register).

(2) Reselection phase response

a. BIT 7 of the Response mode setting register (reselection response) is 1.

b. SCSI data bus data (ID data) satisfies the following conditions.

- The bit setting in the self ID setting register is ”true.”
- There are 2 ”true” bits.
- The parity bit is correct (a parity check is conducted regardless of BIT 7 of the parity error detect setting register).

■ Operation after the Response

There are three types of operations after the selection/reselection response.

- Immediately after the selection/reselection, report a Selected, Selected with ATN, or Reselected interrupt.
- Execute the Automatic selection response mode or Automatic reselection response mode which are already set up in the SPC.

  Automatic selection response mode:Place 0 in BIT 7 and 1 in BIT 5 of the SEL/RESEL mode setting register.

  Automatic reselection response mode:Place 0 in BIT 7 and 1 in BIT 6 of the SEL/RESEL mode setting register.
- Perform a user program operation.

  Place 1 in BIT 7 and the user program start address in BIT 6-0 of the SEL/RESEL mode setting register.

  See 5.4 and 5.5 for a user program example.

■ Selection/Reselection Phase

By sending a command that includes selection/reselection, a sequence from arbitration to selection/reselection is executed.

It is also possible to start directly from the selection phase without executing the arbitration phase.

When the operation following the response is the Automatic selection/reselection response mode or user program operation, the SPC may already be operating when the command including selection/reselection is received.

Confirm that BIT 6 of the SPC status register (BUSY state) is 0 when sending a command that includes selection/reselection.

■ MSG, CMD, STATUS Phases

Use the MCS buffer for transfers taking place in the message phase, command phase, or status phase.

See 5.3.1 for more on these phases.

■ Data Phase

The following types of transfers are possible in the data phase.

Program transfer mode: In this mode, the MPU conducts the transfer by a program. This MPU program uses the data register (00/01 address) to prepare data for sending and to read data received.

DMA transfer mode: In this mode, the data transfer is performed by the DMA controller. The DMA interface is used to prepare data for sending and to read data received.

5.3.1 Message, Command, and Status Phases

Use the MCS buffer for transfers taking place in the message phase, command phase, or status phase.

■ MSG, CMD, STATUS Phases

Watch the following points when executing the message phase, command phase, or status phase.

(1) Send

- Write data for sending from address 0 of the SEND MCS buffer and then send the command.
- When sending a command which requires that the number of bytes be specified, be sure to indicate the number of bytes in the MC byte register ahead of time.
- During the interval after sending a command and until the SPC reports a completion, do not write to the SEND MCS buffer.
- After the transfer is completed, the number of transfer bytes is shown in the modified byte register (when the transfer is ended by an error, the number of bytes transferred normally is shown).

(2) Receive

- When the previous phase is a reception from a non-data phase, read the RECEIVE MCS buffer before sending a command.
- When sending a command which requires that the number of bytes be specified, be sure to indicate the number of bytes in the MC byte register ahead of time.
- During the interval after sending a command and until the SPC reports a completion, do not read from the RECEIVE MCS buffer.
- After the transfer is completed, the number of transfer bytes is shown in the modified byte register (when the transfer is ended by an error, the number of bytes transferred normally is shown).
- After the transfer is completed, read the data received from address 0 of the RECEIVE MCS buffer.

5.3.1.1 Send Flowchart for Message, Command, Status Phases

A flowchart for the send operation in the message, command, and status phases is shown in Figure 5.3.1.1

When N bytes are transferred, be sure to indicate the number of transfer bytes in the MC byte register before implementation.

■ **Send Flowchart for Message, Command, Status Phases**

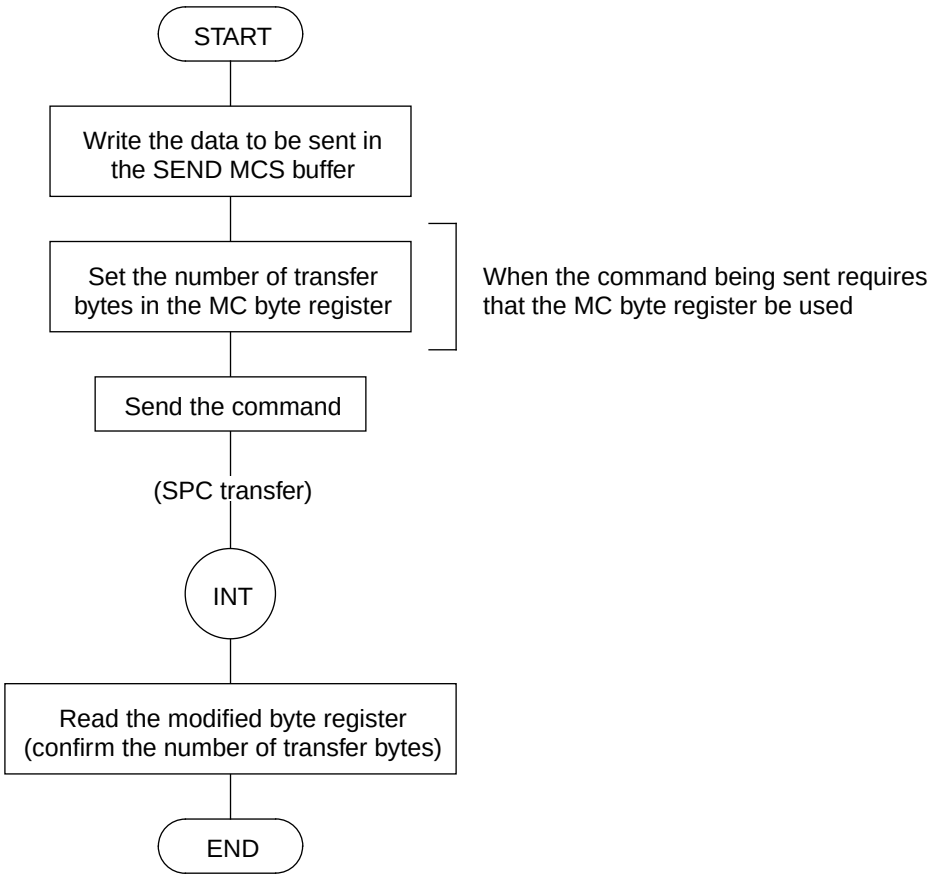


Figure 5.3.1.1 Send Flowchart for Message, Command, Status Phases

5.3.1.2 Receive Flowchart for Message, Command, Status Phases

A flowchart for the receive operation in the message, command, and status phases is shown in Figure 5.3.1.2

When N bytes are transferred, be sure to indicate the number of transfer bytes in the MC byte register before implementation.

■ **Receive Flowchart for Message, Command, Status Phases**

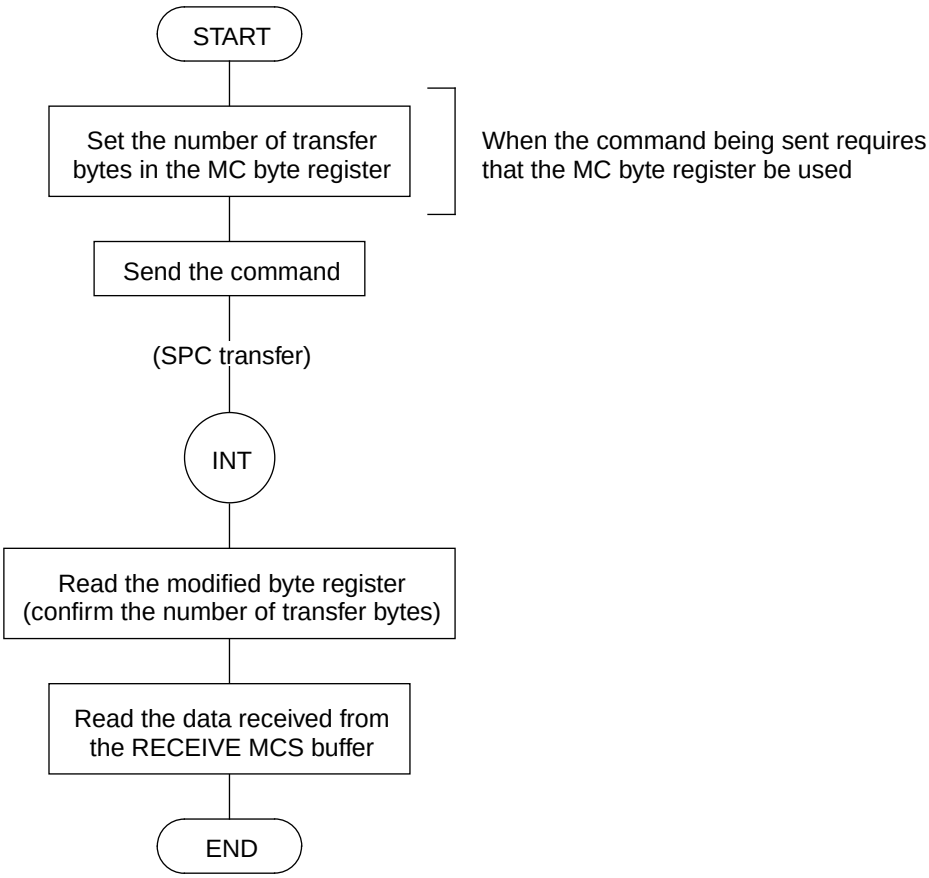


Figure 5.3.1.2 Receive Flowchart for Message, Command, Status Phases

5.3.2 Data Phase

The following types of transfers are possible in the data phase.

Program transfer mode: In this mode, the MPU conducts the transfer by a program. This MPU program uses the data register (00/01 addresses) to prepare data for sending and to read data received.

DMA transfer mode: In this mode, the data transfer is performed by the DMA controller. The DMA interface is used to prepare data for sending and to read data received.

■ Program Transfer Mode

In the program transfer mode, be sure to confirm the SPC state shown in the SPC status register during the transfer.

- Until BIT 2 of the SPC status register (DATA TRANS REQ) becomes 1, do not access the data register.
- BIT 6 (BUSY state) can be used to determine when the transfer is finished.

■ DMA Transfer Mode

The DMA transfer mode has a handshake transfer option which utilizes the DREQ signal and $\overline{\text{DACK}}$ signal and a burst transfer option in which the DREQ signal remains active as long as the data register is in the Ready state. The DMA transfer type can be specified by the DMA system setting register.

- After sending the DMA transfer command, do not activate the $\overline{\text{DACK}}$ signal, $\overline{\text{IORD}}$ signal, or $\overline{\text{IOWR}}$ signal until the DREQ signal becomes H (while the DREQ signal is L).
- When a COMMAND PAUSE command is sent during output (SPC functioning as the target), the DREQ signal is L asynchronously with the $\overline{\text{DACK}}$ signal, $\overline{\text{IORD}}$ signal, and $\overline{\text{IOWR}}$ signal.

■ Points to Watch in Data Phase Execution

Watch the following points in data phase execution.

- Set Synchronous transfer mode transfer parameters (transfer mode, transfer period, transfer offset) before sending the **SEND DATA** command or **RECEIVE DATA** command.
- When the SPC is functioning as the initiator and data in phase is confirmed before a command is sent, data from the target is received in the following manner. However, the ACK signal is not asserted.

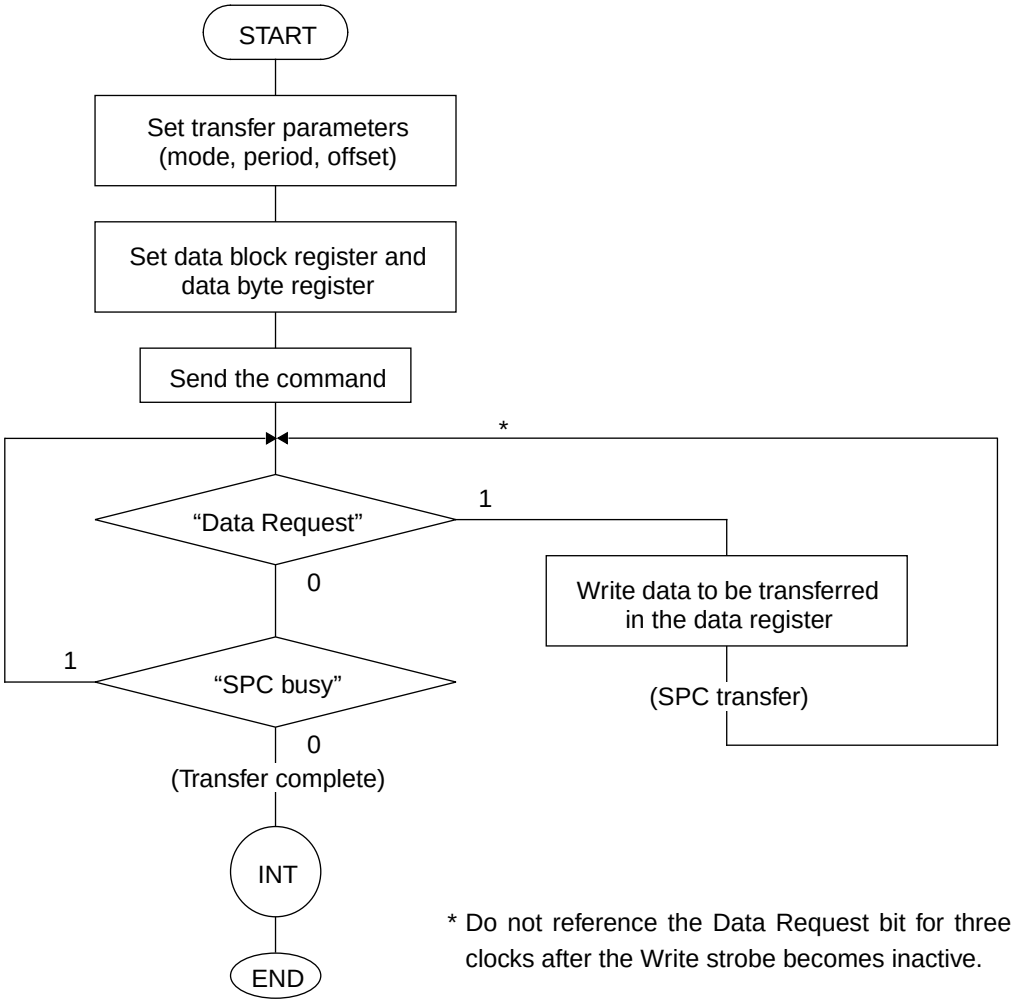
  Asynchronous: 1 byte

  Synchronous: Number of bytes equivalent to the number of REQ pulses received
- When the SPC is functioning as the initiator and a data transfer is performed in the Synchronous mode, set things up so that a single data phase transfer is performed for a single command.
  Do not send the **SEND DATA** command or the **RECEIVE DATA** command two or more times in succession.

5.3.2.1 Send Flowchart for Program Transfer Mode

Figure 5.3.2.1 shows a flowchart for send operations in the program transfer mode. Be sure to set the transfer parameters for synchronous transfers.

■ **Send Flowchart for Program Transfer Mode**



* Do not reference the Data Request bit for three clocks after the Write strobe becomes inactive.

Figure 5.3.2.1 Send Flowchart for Program Transfer Mode

5.3.2.2 Receive Flowchart for Program Transfer Mode

Figure 5.3.2.2 shows a flowchart for receive operations in the program transfer mode. Be sure to set the transfer parameters for synchronous transfers.

■ **Receive Flowchart for Program Transfer Mode**

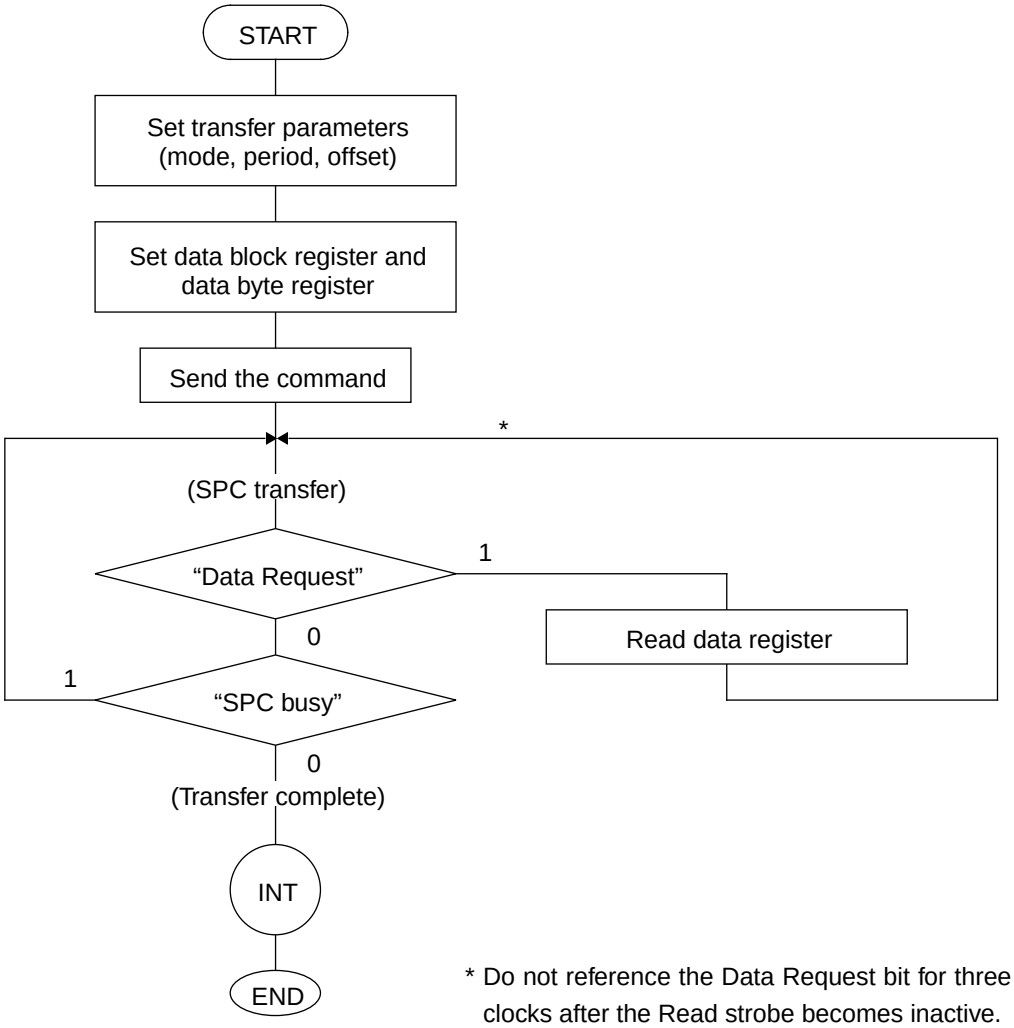


Figure 5.3.2.2 Receive Flowchart for Program Transfer Mode

5.4 USER PROGRAM OPERATION (INITIATOR)

An initiator user programming example which uses a user program to execute command queuing is shown below.

■ Preliminary Conditions for Initiator Program Execution

The following preliminary conditions are required for execution of the initiator user program example explained here.

- The processing results for each phase (interrupt status, command step) are stored sequentially from address 90_H. (For example 1)
- When an Initial Phase Error occurs, the phase is determined and operation is stopped after reception.
- ACK Reset mode is set as "Auto" and Level 2 INT report mode is set as "not report" in the Auto operation Mode Register (1ch). (For example 2)

Figure 5.4 shows a summary flowchart of an initiator programming example. For a more detailed flowchart of program operations, see Appendix E.

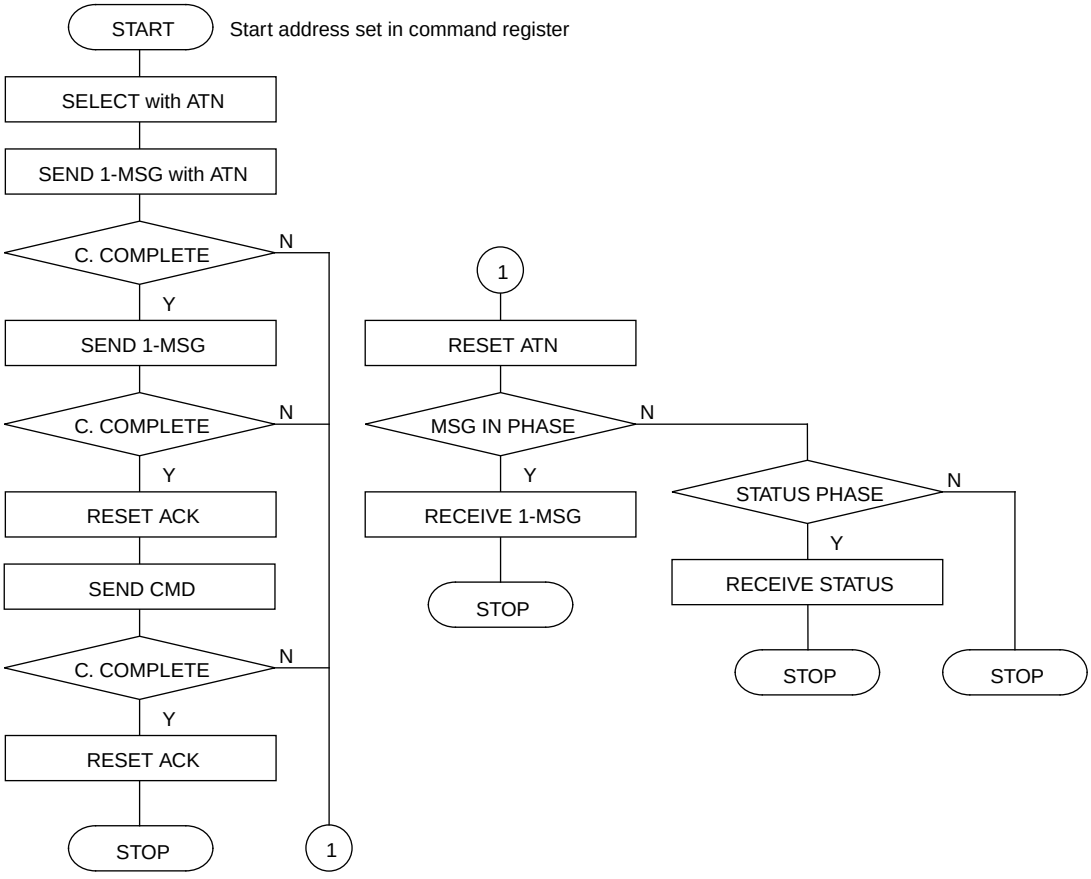


Figure 5.4 Summary Flowchart of Initiator User Programming Example

MEMO

5.4.1 Explanation of Initiator User Programming Example 1 (Address 00_H to 34_H)

Table 5.4.1 explains the initiator user programming example (address 00_H to 34_H).

■ **Explanation of Initiator User Programming Example (Address 00_H to 34_H)**

Table 5.4.1 Explanation of Initiator User Programming Example (Address 00_H to 34_H)

No.	Address	Code	Explanation
1	00	09	**SELECT WITH ATN** command
2	01 02	EC 90	MOVE the interrupt status register to address 90_H
3	03 04	ED 91	MOVE the command step register to address 91_H
4	05 06	19 B0	**SEND 1-MSG** with ATN command. The data to be transferred is 1 byte starting from $B0_H$.
5	07 08	EC 92	MOVE the interrupt status register to address 92_H
6	09 0A	ED 93	MOVE the command step register to address 93_H
7	0B 0C	E0 92	MOVE the data stored in address 92_H to the accumulator
8	0D 0E	CF 40	Perform a compare calculation for the accumulator value and the data stored in address 40_H
9	0F 10	D1 50	When the calculation result is not 0, branch to address 50_H
10	11 12	18 B2	**SEND 1-MSG** command. The data to be transferred is 2 bytes starting from $B2_H$.
11	13 14	EC 94	MOVE the interrupt status register to address 94_H
12	15 16	ED 95	MOVE the command step register to address 95_H
13	17 18	E0 94	MOVE the data stored in address 94_H to the accumulator
14	19 1A	CF 40	Perform a compare calculation for the accumulator value and the data stored in address 40_H
15	1B 1C	D1 50	When the calculation result is not 0, branch to address 50_H
16	1D	0D	**RESET ACK** command
17	1E 1F	EC 96	MOVE the interrupt status register to address 96_H
18	20 21	ED 97	MOVE the command step register to address 97_H
19	22 23	1B B8	**SEND CMD** command. The data to be transferred is 6 bytes starting from $B8_H$
20	24 25	EC 98	MOVE the interrupt status register to address 98_H
21	26 27	ED 99	MOVE the command step register to address 99_H

Table 5.4.1 Explanation of Initiator User Programming Example (Address 00_H to 34_H) *(Continued)*

No.	Address	Code	Explanation
22	28 29	E0 98	MOVE the data stored in address 98_H to the accumlator
23	2A 2B	CF 40	Perform a compare calculation for the accumulator value and the data stored in address 40_H
24	2C 2D	D1 50	When the calculation result is not 0, branch to address 50_H
25	2E	0D	**RESET ACK** command
26	2F 30	EC 9A	MOVE the interrupt status register to address $9A_H$
27	31 32	ED 9B	MOVE the command step register to address $9B_H$
28	33 34	91 E0	**STOP** command. The interrupt code reported is $E0_H$ (normal comletion).

5.4.2 Explanation of Initiator User Programming Example 1 (Address 40_H to $E0_H$)

Table 5.4.2 explains the initiator user programming example 1(address 40_H to $E0_H$).

■ **Explanation of Initiator User Programming Example (Address 40_H to $E0_H$)**

Table 5.4.2 Explanation of Initiator User Programming Example (Address 40_H to $E0_H$)

No.	Address	Code	Explanation
—	40	60	Store the 60_H value for comparison with the interrupt status register
29	50	0B	**RESET ATN** command
30	51 52	EC 9C	MOVE the interrupt status register to address to $9C_H$
31	53 54	ED 9D	MOVE the command step register to address $9D_H$
32	55 56	C6 8F	Perform a comparison (COMPARE) with the SCSI control register
33	57 58	D1 70	When the calculation result is not 0, branch to address 70_H
34	59 5A	1A C0	**RECEIVE 1-MSG** command. The data received is accepted starting from $C0_H$.
35	5B 5C	EC 9E	MOVE the interrupt status register to address $9E_H$
36	5D 5E	ED 9F	MOVE the command step register to address $9F_H$
37	5F 60	91 E2	**STOP** command. The interrupt code reported is $E2_H$ (a single message is receivedafter an Initial Error Phase occurs).
38	70 71	C6 8B	Perform a comparison (COMPARE) between the SCSI control register and $8B_H$
39	72 73	D1 7C	When the calculation result is not 0, branch to address $7C_H$
40	74 75	1C E0	**RECEIVE STATUS** command. The data received is accepted starting from $E0_H$.
41	76 77	EC A1	MOVE the interrupt status register to address $A0_H$
42	78 79	EC A1	MOVE the command step register to address $A1_H$
43	7A 7B	91 E3	**STOP** command. The interrupt code reported is $E3_H$ (a status is received after an Initial Error Phase occurs).
44	7C 7D	91 E4	**STOP** command. The interrupt code reported is $E4_H$ (when the next phase is not RECEIVE MCS after an Initial Error Phase occurs).
—	90 91	—	Store **SELECT with ATN** command execution results (interrupt status and command step. Same below)
—	92 93	—	Store **SEND 1-MSG with ATN** command execution results
—	94 95	—	Store **SEND 1-MSG** command execution results
—	96 97	—	Store **RESET ACK** command execution results

Table 5.4.2 Explanation of Initiator User Programming Example (Address 40_H to $E0_H$) *(Continued)*

No.	Address	Code	Explanation
—	98 99	—	Store **SEND CMD** command execution results
—	9A 9B	—	Store **RESET ACK** command execution results
—	9C 9D	—	Store **RESET ATN** command execution results
—	9E 9F	—	Store **RECEIVE 1-MSG** command execution results
—	A0 A1	—	Store **RECEIVE STATUS** command execution results
—	B0	XX	Store data transferred with **SEND 1-MSG** with ATN command (1 byte)
—	B2 B3	XX XX	Store data transferred with **SEND 1-MSG** command (2 bytes)
—	B8 B9 BA BB BC BD	XX XX XX XX XX XX	Store data transferred with **SEND CMD** command (6 bytes)
—	CD to DF	—	When a single message is received after an Initial Phase Error occurs, this data is stored.
—	E0	—	When a single message is received after an Initial Phase Error occurs, this data is stored

5.4.3 Explanation of Initiator User Programming Example 2 (Address 00_H to 70_H)

Table 5.4.3 explains the initiator user programming example 2 (address 00_H to 70_H)

■ Explanation of Initiator User Programming Example (Address 00_H to 70_H)

Table 5.4.3 Explanation of Initiator User Programming Example (Address 00_H to 70_H)

No.	Address	Code	Explanation
1	00	09	**SELECT WITH ATN** command
2	01 02	19 30	**SEND 1-MSG with ATN** command. Data to be sent is 1-byte from address 30_H.
3	03 04	D1 10	When "Initial Phase Error" occurred, branch to address 10_H.
4	05 06	18 31	**SEND 1-MSG** command. The data to be transferred is 2 bytes starting from address 31_H.
5	07 08	D1 10	When "Initial Phase Error" occurred, branch to address 10_H.
6	09 0A	1B 40	**SEND CMD** command. Data to be transferred is 6 bytes from address 40_H.
7	0B 0C	D1 10	MOVE the data stored in address 92_H to the accumulator When "Initial Phase Error" occurred, branch to address 10_H.
8	0D 0E	91 E0	**STOP** command. The interrupt code reported is $E0_H$. (normal completion)
—	0F		
9	10	0B	**RESET ATN** command.
10	11 12	C6 8F	Compare data in the SCSI control register with $8F_H$.
11	13 14	D1 20	When the result of the comparison is not "0", branch to address 20_H.
12	15 16	1A 50	**RECEIVE MSG** command. Data received is stored from address 50_H.
13	17	0D	**RESET ACK** command.
14	18 19	91 E2	**STOP** command. The interrupt code reporeted is $E2_H$. (1-MSG received following "Initial Phase Error".)
15	20 21	C6 8B	Compare data in the SCSI control register with $8B_H$.
16	22 23	D1 28	When the result of the comparison is not "0", branch to address 28_H.
17	24 25	1C 70	**RECEIVE STATUS** command. Data received is stored from address 70_H.

Table 5.4.3 Explanation of Initiator User Programming Example (Address 00_H to 70_H) *(Continued)*

No.	Address	Code	Explanation
18	26 27	91 E3	**STOP** command. The interrupt code reported is $E3_H$ (Status received following "Initial Phase Error".)
19	28 29	91 E4	**STOP** command. The interrupt code reported is $E4_H$ (When the next phase is not Receive MCS after "Initial Phase Error".)
—	30	XX	Store the data to be sent by **SEND 1-MSG WITH ATN** command.
—	31 32	XX XX	Store the 2 bytes data to be sent by **SEND 1-MSG** command.
—	40 to 4F	XX to XX	Store the data to be sent by **SEND CMD** command.
—	50 to 6F	—	When 1-MSG is received after "Initial Phase Error", store the data.
—	70	—	When 1-status is received after "Initial Phase Error", store the data.

5.5 USER PROGRAM OPERATION (TARGET USER PROGRAMMING EXAMPLE)

A target user programming example which uses a user program to execute command queuing is shown below.

■ Preliminary Conditions for Target Program Execution

The following preliminary conditions are required for execution of the target user program example explained here.

- The processing results for each phase (interrupt status, command step) are stored sequentially from address 50_H. (For example 1)
- When Command Complete (ATN Condition Detected) occurs, data is not received and operation is stopped.
- ACK Reset mode is set as "Auto" and Level 2 INT report mode is set as "not report" in the Auto operation Mode Register (1ch). (For example 2)

Figure 5.5 shows a summary flowchart of a target user programming example. For a more detailed flowchart of program operations, see Appendix F.

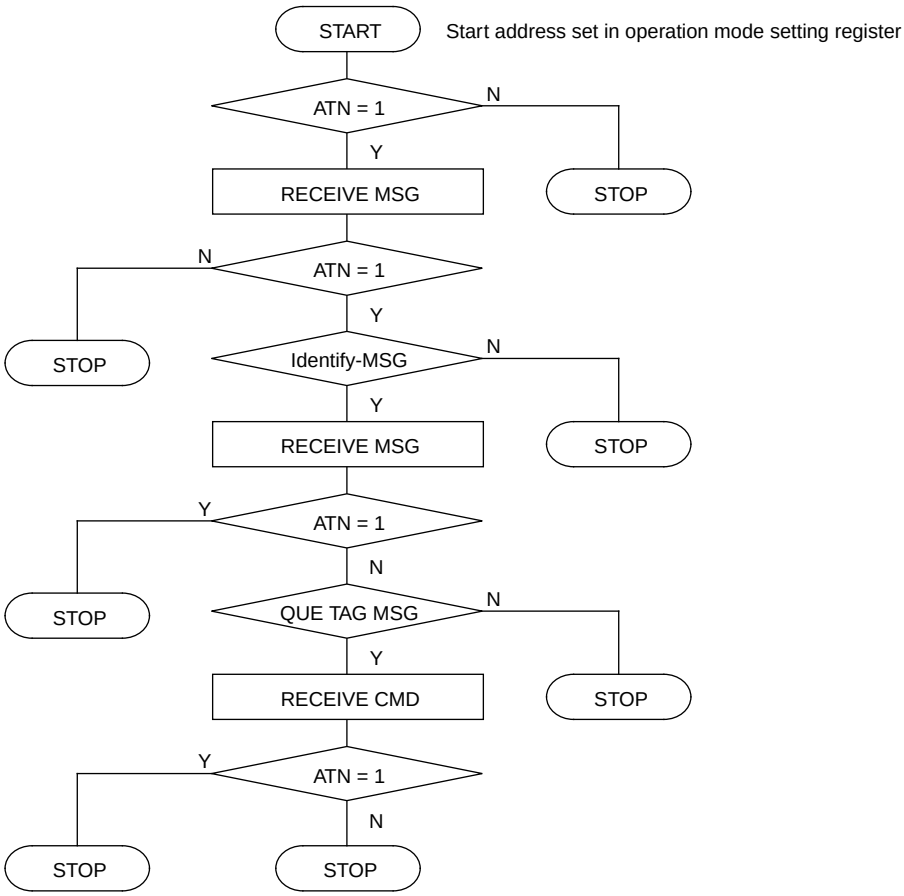


Figure 5.5 Summary Flowchart of Target User Programming Example

5.5.1 Explanation of Target User Programming Example 1 (Address 00_H to 21_H)

Table 5.5.1 explains the target user programming example 1 (address 00_H to 21_H).

■ Explanation of Target User Programming Example 1 (Address 00_H to 21_H)

Table 5.5.1 Explanation of Initiator User Programming Example 1(Address 00_H to 21_H)

No.	Address	Code	Explanation
1	00 01	B6 20	Logiacal product calculation (AND) for the SCSI control signal status register and 20_H.
2	02 03	D2 40	When the calculation result is 0, branch to address 40_H.
3	04 05	39 60	**RECEIVE MSG** command. The data received is stored in address 60_H.
4	06 07	EC 50	MOVE the interrupt status register to address 50_H.
5	08 09	ED 51	MOVE the command step register to address 51_H.
6	0A 0B	E0 50	MOVE the data stored in address 50_H to the accumulator.
7	0C 0D	C7 61	Perform a comparison for the accumulator value and 61_H.
8	0E 0F	D1 42	When the calculation result is not 0, branch to address 42_H.
9	10 11	B0 80	Logical product calculation (AND) for the first byte of the message received and 80_H.
10	12 13	D2 44	When the calculation result is 0, branch to address 44_H.
11	14 15	39 80	**RECEIVE MSG** command. The data received is stored starting from address 80_H.
12	16 17	EC 52	MOVE the interrupt status register to address 52_H.
13	18 19	ED 53	MOVE the command step register to address to 53_H.
14	1A 1B	E0 52	MOVE the data stored in address 52_H to the accumulator.
15	1C 1D	C7 60	Perform a comparison for the accumulator value and 60_H.
16	1E 1F	D1 46	When the calculation result is not 0, branch to address 46_H.
17	20 21	A0 F0	Logical product calculation (AND) for the first byte of the message received and $F0_H$.

5.5.2a Explanation of Target User Programming Example 1 (Address 22_H to $4B_H$)

Table 5.5.2a explains the target user programming example 1 (address 22_H to $4B_H$).

■ Explanation of Target User Programming Example 1 (Address 22_H to $4B_H$)

Table 5.5.2a Explanation of Initiator User Programming Example 1 (Address 22_H to $4B_H$)

No.	Address	Code	Explanation
18	22 23	C7 20	Comparison for accumulator value and 20_H.
19	24 25	D1 48	When the calculation result is not 0, branch to address 48_H.
20	26 27	3B A0	**RECEIVE CMD** command. The data received is stored starting from address $A0_H$.
21	28 29	EC 54	MOVE the interrupt status register to address 54_H.
22	2A 2B	ED 55	MOVE the command step register to address 55_H.
23	2C 2D	E0 54	MOVE the data stored in address 54_H to the accumulator.
24	2E 2F	C7 60	Perform a comparison for the accumulator value and 60_H.
25	30 31	D1 4A	Wehn the calculation result is not 0, branch to address $4A_H$.
26	32 33	91 E0	**STOP** command. The interrupt code reported is $E0_H$ (normal completion).
27	40 41	91 E1	**STOP** command. The interrupt code reported is $E1_H$ (the ATN signal is not being asserted).
28	42 43	91 E2	**STOP** command. the interrupt code reported is $E2_H$ (after the first message is received, the ATN signal is not asserted).
29	44 45	91 E3	**STOP** command. The interrupt code reported is $E3_H$ (the first message received is not Identify-MSG).
30	46 47	91 E4	**STOP** command. the interrupt code reported is $E4_H$ (an attention condition is detected when the second message is received).
31	48 49	91 E5	**STOP** command. The interrupt code reported is $E5_H$ (the second message received is not Queue Tag MSG).
32	4A 4B	91 E6	**STOP** command. The interrupt code reported is $E6_H$ (command is received and an attention condition detected).

MEMO

5.5.2b Explanation of Target User Programming Example 2 (Address 00_H to $8F_H$)

Table 5.5.2b explains the target user programming example 2 (address 00_H to $8F_H$)

■ Expanation of Target User Programming Example 2 (Address to 00_H to $8F_H$)

Table 5.5.2b Explanation of Initiator User Programming Example 2(Address 00_H to $8F_H$)

No.	Address	Code	Explanation
1	00 01	B6 20	Logical product calculation (AND) for the SCSI control signal status register and 20_H.
2	02 03	D2 20	When the calculation result is 0, branch to address 20_H.
3	04 05	39 30	**RECEIVE MSG** command. the data received is stored in address 30_H.
4	06 07	D2 22	When the result of command execution is "Command Complete", branch to address 22_H.
5	08 09	B0 80	Logical product calculation (AND) for the first byte of MSG received and 80_H.
6	0A 0B	D2 24	When the result of AND operation is "0", branch to address 24_H.
7	0C 0D	39 50	Receive MSG command. The data received is stored in address 50_H.
8	0E 0F	D1 26	When the calculation result is "Command Complete (ATN Condition Detected)", branch to address 26_H.
9	10 11	A0 F0	Logical product calculation (AND) for the first byte of the message received and $E0_H$.
10	12 13	C7 20	Compare the accumulator value with 20_H.
11	14 15	D1 28	When the result is not "0", branch to address 28_H.
12	16 17	3B 70	**RECEIVE CMD** command. The data received is stored from address 70_H.
13	18 19	D1 2A	When the result is not "0", branch to address $2A_H$.
14	1A 1B	91 E0	**STOP** command. The interrupt code reported is $E0_H$ (normal completion).
15	20 21	91 E1	**STOP** command. The interrupt code reported is $E1_H$ (ATN not asserted in selection phase.)
16	22 23	91 E2	**STOP** command. The interrupt code reported is $E2_H$ (ATN not asserted after receiving the first MSG.)
17	24 25	91 E3	**STOP** command. The interrupt code reported is $E3_H$ (The first MSG received was not "Identify MSG".)

Table 5.5.2b Explanation of Initiator User Programming Example 2(Address 00_H to $8F_H$) *(Continued)*

No.	Address	Code	Explanation
18	26 27	91 E4	**STOP** command. The interrupt code reported is $E4_H$ (ATN Condition detectecd after receiving the second MSG.)
19	28 29	91 E5	**STOP** command. The interrupt code reported is $E5_H$ (The 2nd MSG received was not "Queue Tag MSG".)
20	2A 2B	91 E6	**STOP** command. The interrupt code reported is $E6_H$ (ATN Condition detected after receiving CMD)
—	30 to 4F	—	Store the data received by the first Receive MSG command.
—	50 to 6F	—	Store the data received by the second Receive MSG command.
—	70 to 8F	—	Store the data received by the Receive CMD command.

5.5.3 Explanation of Target User Programming Example 1 (Address 50_H to BF_H)

Table 5.5.3 explains the target user programming example 1 (address 50_H to BF_H)

■ Explanation of Target User Programming Example 1 (Address 50_H to BF_H)

Table 5.5.3 Explanation of Target User Programming Example 1 (Address 50_H to BF_H)

No.	Address	Code	Explanation
—	50 51	—	Store first **RECEIVE MSG** command execution results (interrupt status and command step. Same below.)
—	52 53	—	Store second **RECEIVE MSG** command execution results
—	54 55	—	Store **RECEIVE CMD** command execution results
—	60 to 7F	—	Store data received from first **RECEIVE MSG** command
—	80 to 9F	—	Store data received from second **RECEIVE MSG** command
—	A0 to BF	—	Store data received from **RECEIVE CMD** command

MEMO

5.6 COMMAND QUEUING EXECUTION

The following items are necessary for execution of command queuing.

- **User program**
- **Assign a portion of the user program region as a command queuing region**
- **Queuing line management by the host MPU**

■ Example of Command Queuing Execution (Target)

As seen in Figure 5.6a, a command queuing example with a system configuration of two initiator systems and 2 LUN is explained.

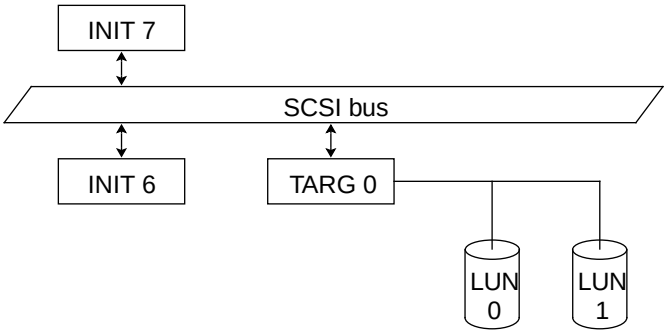
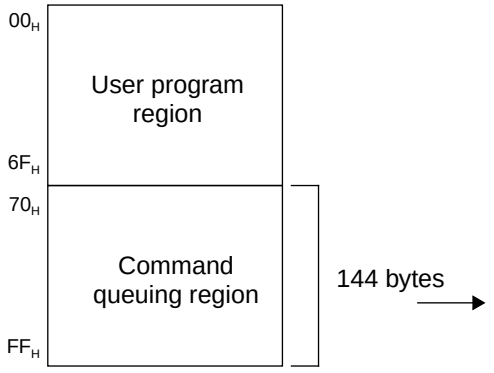


Figure 5.6a System Configuration (Example of Command Queuing Execution)

In a case where three CDB (command descriptor block) are written to a single queue, as shown in Figure 5.6b, a portion of the user program region is assigned as a queuing region and a region for holding 3 CDBs is allocated to each queue (12 bytes of space are required for a single CDB).



	LUN 0	LUN 1
INIT 7	70_H to 93_H address	94_H to $B7_H$ address
INIT 6	$B8_H$ to DB_H address	DC_H to FF_H address

Figure 5.6b Allocation of Command Queuing Region

The user program **`RECEIVE CMD`** command (operand undefined) is used for storing the received CDB in the address allocated for each queue within the user program region.

■ Command Queuing Operation Sequence

The target Automatic selection response mode is established.

When the initiator performs command queuing in the manner shown in Figure 5.6c, a “MSG received with Identify MSG” interrupt is reported.

After the host MPU confirms the initiator ID number by means of the SPC status register (2), analyze the Identify MSG and determine the queuing LUN.

Based on this, set up one of the four possible queues.

After that, use the user program **`RECEIVE CMD`** command to write the CDB in the appropriate command queuing region.

At that time, the **`RECEIVE CMD`** command operand should be specified by host MPU management.

It is recommended that the host MPU handle the head address of the queuing region containing the Queue TAG MSG, Queue TAG Value, and CDB as a single block.

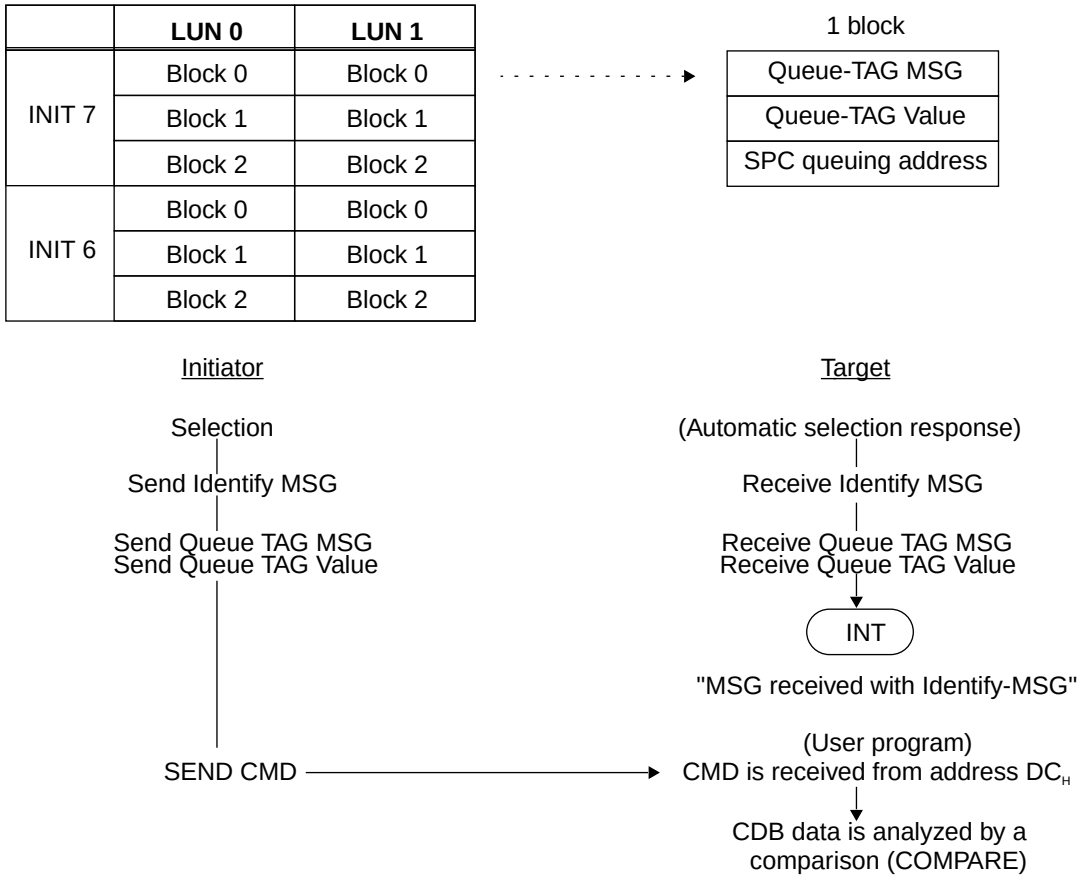


Figure 5.6c Command Queuing Operation Sequence

5.7 INTERRUPT CODES

Table 5.7 shows a list interrupt codes and the accompanying causes.

■ Interrupt Codes

Table 5.7 List of Interrupt Codes

Item		INIT or TARG	AUTO1	AUTO2	Interrupt code								Interrupt cause
1	SCSI RESET	I/T	—	—	0	0	0	0	0	0	0	1	RESET-condition detected
2	TRANSFER ERROR	I/T	—	—	0	0	1	0	0	1	0	0	SCSI parity error
			—	—	0	0	1	0	0	0	1	0	MPU parity error
			—	—	0	0	1	0	0	0	0	1	DMA parity error
			—	—	0	0	1	0	0	1	1	0	SCSI & MPU parity error
			—	—	0	0	1	0	0	1	0	1	SCSI & DMA parity error
			—	—	0	0	1	0	1	0	0	1	Offset error
			—	—	0	0	1	0	1	0	1	0	Short transfer period error
			—	—	0	0	1	0	1	0	1	1	Offset error & Short transfer period error
			—	—	0	0	1	0	1	1	0	0	REQ/ACK Timeout
		I	—	—	0	0	1	1	0	0	0	1	Disconnected in transfer progress
			—	—	0	0	1	1	0	0	1	0	Phase-error in transfer progress
3	PHASE ERROR	I	N	—	0	1	0	1	0	1	0	0	Initial phase-error
			Y	—	0	1	0	1	0	1	0	1	Initial phase-error & MSG received
			Y	—	0	1	0	1	0	1	1	0	Initial phase-error & status received
		T	N	—	0	1	0	0	0	0	1	0	Command stop (ATN condition detected)
			Y	—	0	1	0	0	0	0	1	1	Command stop (ATN condition detected) & MSG received
4	REPORT	I/T	—	—	0	1	1	0	0	0	0	0	Command Complete
			—	—	0	1	1	0	0	1	0	0	Command Rejected
			—	—	0	1	1	0	0	1	0	1	Command Invalid
			—	—	0	1	1	0	0	1	1	0	REG parity error detected
			—	—	0	1	1	0	0	1	1	1	Command Pause
		I/T	—	—	0	1	1	0	1	0	0	0	Self diagnosis result is Good
			—	—	0	1	1	0	1	0	0	1	Self diagnosis result is No Good

Table 5.7 List of Interrupt Codes *(Continued)*

Item		INIT or TARG	AUTO 1	AUTO 2	Interrupt code								Interrupt cause
4	REPORT	I	—	—	0	1	1	1	0	0	0	0	Disconnected
			—	—	0	1	1	1	0	0	0	1	REQ Asserted
		T	N	—	0	1	1	0	0	0	0	1	Command complete (ATN condition detected)
			Y	—	0	1	1	1	0	0	1	0	Command complete (ATN condition detected) & MSG received
5	SELECTION/ RESELECTION	I/T	—	—	1	0	0	0	0	0	1	0	Selection/Reselection timeout
		I	—	N	1	0	0	1	0	0	0	0	Reselected
		T	—	N	1	0	0	0	0	0	0	0	Selected
			—	N	1	0	0	0	0	0	0	1	Selected with ATN
6	AUTO SEL/ RESELECT/ REPORT	I	—	Y	1	0	1	1	0	0	0	0	AUTO mode phase error
			—	Y	1	0	1	1	0	0	0	1	MSG received
		T	—	Y	1	0	1	0	0	0	0	0	AUTO mode phase error
			—	Y	1	0	1	0	0	0	0	1	MSG received
			—	Y	1	0	1	0	0	0	1	0	Illegal identify-MSG received
			—	Y	1	0	1	0	0	0	1	1	MSG received with identify-MSG
			—	Y	1	0	1	0	0	1	0	0	unknown CDB received
			—	Y	1	0	1	0	0	1	0	1	unknown CDB received with identify-MSG
			—	Y	1	0	1	0	0	1	1	0	CDB received
			—	Y	1	0	1	0	0	1	1	1	CDB received with identify-MSG
			Y	Y	1	0	1	0	1	0	0	0	CDB received & MSG received
			—	Y	1	0	1	0	1	0	0	1	CDB received with identify-MSG & MSG-received
			N	Y	1	0	1	0	1	0	1	0	CDB received (ATN condition detected)

AUTO 1: Automatic receive mode
AUTO 2: Automatic selection/reselection response mode
Y = set N = not set

5.7.1 Interrupts Related to SCSI RESET

Table 5.7.1 provides a list of interrupts codes, interrupt causes, and interrupt conditions for interrupts related to SCSI RESET.

■ **Interrupts Related to SCSI RESET**

Table 5.7.1 List of Interrupts Related to SCSI RESET

Interrupt code	Interrupt cause	Interrupt condition
01_H	RESET-CONDITION DETECTED	Reset state is detected on the SCSI bus

5.7.2 Interrupts Related to Transfers

Table 5.7.2 provides a list of interrupts codes, interrupt causes, and interrupt conditions for interrupts related to transfers.

■ Interrupts Related to Transfers

Table 5.7.2 List of Interrupts Related to Transfers

Interrupt code	Interrupt cause	Interrupt condition
24_H	SCSI parity error	A parity error is detected in SCSI interface input/output data during the phase currently being executed.
22_H	MPU parity error	A parity error is detected in MPU interface input/output data during a program transfer (data phase).
21_H	DMA parity error	A parity error is detected in DMA interface input/output data during a DMA transfer (data phase).
26_H	SCSI & MPU parity error	A parity error is detected in MPU interface and SCSI interface input/output data during a program transfer (data phase).
25_H	SCSI & DMA parity error	A parity error is detected in DMA interface and SCSI interface input/output data during a DMA transfer (data phase).
29_H	Offset error	More REQ/ACK signals are received during a synchronous transfer than the specified offset value.
$2A_H$	Short transfer period error	The SPC cannot keep up during a synchronous transfer because the period of the REQ/ACK signal received is short.
$2B_H$	Offset error & Short transfer period error	An offset error and a short transfer period error are detected during a synchronous transfer.
$2C_H$	REQ/ACK timeout	The specified REQ/ACK timeout time is exceeded.
31_H	Disconnected in transfer progress	A bus free state is detected during phase execution (during a transfer).
32_H	Phase error in transfer progress	A phase error occurs during phase execution (during a transfer).

5.7.3 Interrupts Related to Phase Transitions

Table 5.7.3 provides a list of interrupts codes, interrupt causes, and interrupt conditions for interrupts related to phase transitions.

■ Interrupts Related to Phase Transitions

Table 5.7.3 List of Interrupts Related to Phase Transitions

Interrupt code	Interrupt cause	Interrupt condition
54_H	Initial phase-error*1	The phase necessary for executing the received command is not the same as the phase requested on the SCSI bus when the initial REQ signal is received.
55_H	Initial phase-error & MSG received*2	An Initial phase error occurs, and a single message is received in response to a target message receive request.
56_H	Initial phase-error & STATUS received*2	An Initial phase error occurs, and a single status is received in response to a target status receive request.
42_H	Command Stop (ATN condition detected)*3	A phase transition is impeded by an initiator generated attention condition. Or a transfer stops at a transfer block boundary during data phase execution (at this time, the data register is Empty).
43_H	Command Stop (ATN condition detected) & MSG received*2	A phase transition specified by a command is impeded by an initiator generated attention condition, the SPC moves to a message out phase, and a single message is received. Or the SPC moves to a message out phase at a transfer block boundary and a single message is received during data phase execution (at this time, the data register is Empty).

*1 When a phase error occurs and a data in phase is requested, the SPC performs the following operations.

- When the nexus target is in an asynchronous transfer mode, one byte of data is received.
- When the nexus target is in a synchronous transfer mode, the number of received REQ signal bytes is received (ACK signal is not asserted).

*2 Only when the Automatic receive mode is established.

*3 Only when the Automatic receive mode is not established.

MEMO

5.7.4 Interrupts Related to REPORT

Table 5.7.4 provides a list of interrupts codes, interrupt causes, and interrupt conditions for interrupts related to REPORT.

■ Interrupts Related to REPORT

Table 5.7.4 List of Interrupts Related to REPORT

Interrupt code	Interrupt cause	Interrupt condition
60_H	Command complete	The received command is completed normally.
64_H	Command Rejected	• A command is received when the SPC is operating on automatic (Automatic selection/reselection response). • A new command is received during execution of a command. • The SPC receives a SEND MSG command, but the extended message which should be sent is 33 bytes or longer.*1
65_H	Command Invalid	• A target command is received while the SPC is functioning as the initiator. • An initiator command is received while the SPC is functioning as the target. • A sequential command exists in the user program memory. • A command which requires that the data block register or the data byte register MC byte register) be established is received but the register value is 0. • A transfer is not possible because the group 6/7 CDB length setting is 0.*2 • A group 3/4 CDB is received or a transfer command is received.*2 • A command which starts from a transfer phase is received when a nexus is not established. • A command which starts from Select/Reselect is received when a nexus is established.
66_H	REG parity error detected	A parity error is detected in a register established by the host MPU.
67_H	Command Pause	• A Command Pause command is received and processing completed. • The message received is 33 bytes or longer.*3
68_H	Self diagnosis result “GOOD”	A self diagnosis is performed and the result is normal.
69_H	Self diagnosis result “NO GOOD”	A self diagnosis is performed and the result is abnormalities exist.
70_H	Disconnected	• A bus free state is detected when the SPC is connected as the initiator. • Detected when a transfer is not being conducted.

Table 5.7.4 List of Interrupts Related to REPORT *(Continued)*

Interrupt code	Interrupt cause	Interrupt condition
71_H	REQ Asserted*4	• The target requests that the next phase be executed. • This interrupt is only applicable when a nexus exists and no command is received.
61_H	Command complete (ATN condition detected)*5	• The command is completed normally, and an attention condition generated by the initiator is detected. • The ATN signal is detected only when the ATN signal is asserted while the ACK signal of the last phase of the executed command is being asserted.
62_H	Command complete (ATN condition detected) & MSG received*6	• The command is completed normally, a response is made to the attention condition generated by the initiator, and a single message is received. • The ATN signal is detected only when the ATN signal is asserted while the ACK signal of the last phase of the executed command is being asserted.

*1 Send messages which are 33 bytes or longer in the following manner.

- Initiator

SEND N-BYTE-MSG → (COMMAND COMPLETE) → SET ATN → RESET-ACK

(COMMAND COMPLETE) ←

- Target

SEND N-BYTE-MSG → SEND N-BYTE-MSG

*2 Transfers are not conducted during send operations.
For receive operations, this interrupt is reported after the first byte is written in the RECEIVE MCS buffer (written in the specified user program address during user program operation).

*3 Occurs during receive operations.
This interrupt is reported after the first two bytes are written in the RECEIVE MCS buffer (written in the specified user program address during user program operation).
Use the **`RECEIVE N-byte MSG`** command to receive the remaining bytes.

*4 The SPC performs the following operations when the data in phase is requested.

- When the nexus target is in an asynchronous transfer mode, one byte of data is received.
- When the nexus target is in a synchronous transfer mode, the number of received REQ signal bytes is received.

*5 Only when the Automatic receive mode is not established.

*6 Only when the Automatic receive mode is established.
When executing a command which ends in the message out phase, if another attention condition is detected, this interrupt is not reported (message not received) and instead a Command complete (ATN condition detected) interrupt is reported.

5.7.5 Interrupts Related to Selection/Reselection

Table 5.7.5 provides a list of interrupts codes, interrupt causes, and interrupt conditions for interrupts related to selection/reselection.

■ Interrupts Related to Selection/Reselection

Table 5.7.5 List of Interrupts Related to Selection/Reselection

Interrupt code	Interrupt cause	Interrupt condition
82_H	Selection/Reselection timeout	The specified selection/reselection timeout time is exceeded.
90_H	Reselected*1	The SPC is reselected by the target while functioning as the initiator
80_H	Selected*2	The SPC is selected by the initiator while functioning as the target.
81_H	Selected with ATN*2	The SPC is selected by the initiator while functioning as the target, and an attention condition generated by the initiator is detected.

*1 Only when the Automatic reselection response mode is not established.
*2 Only when the Automatic selection response mode is not established.

MEMO

5.7.6 Interrupts Related to Automatic Selection/Reselection Response

Table 5.7.6 provides a list of interrupts codes, interrupt causes, and interrupt conditions for interrupts related to Automatic selection/reselection response.

■ Interrupts Related to Automatic Selection/Reselection Response

Table 5.7.6 List of Interrupts Related to Automatic Selection/Reselection Response

Interrupt code	Interrupt cause	Interrupt condition
$B0_H$	AUTO mode phase error	Transition to phase other than the message in phase after the reselection phase.
$B1_H$	MSG received	A single message is received after the reselection phase.
$A0_H$	AUTO mode phase error	The ATN signal is not asserted during the selection phase. This interrupt is only applicable when the message out phase is established after the selection phase.
$A1_H$	MSG-received	A single message other than the Identify MSG is received after the selection phase.
$A2_H$	Illegal Identify-MSG received	The Identify MSG is received after the selection phase, but 1 is set in BIT 5, 4, 3.
$A3_H$	MSG received with Identify-MSG	The Identify MSG is received after the selection phase, and then a single message is received.
$A4_H$	unknown-CDB-received	An attempt is made to receive a single command after the selection phase, but the group 6 and 7 CDB length is unknown (*2) so only one byte is received. This interrupt is only applicable when the message phase or the command phase (*1) are established after the selection phase.
$A5_H$	unknown-CDB-received with Identify-MSG	The Identify-MSG is received after the selection phase, and an attempt is made after that to receive a command, but the group 6 and 7 CDB length is unknown (*2) so only one byte is received.
$A6_H$	CDB-received	A single command is received after the selection phase. This interrupt is only applicable when the message phase or command phase are established after the selection phase.
$A7_H$	CDB-received with Identify-MSG	The Identify MSG is received after the selection phase, and then a single command is received.
$A8_H$	CDB received & MSG received	A single command is received after the selection phase. After that, a single message is received which is related to an attention condition. This interrupt is only applicable when the message phase or command phase are established after the selection phase.

Table 5.7.6 List of Interrupts Related to Automatic Selection/Reselection Response *(Continued)*

Interrupt code	Interrupt cause	Interrupt condition
$A9_H$	CDB-received with Identify-MSG & MSG received	The Identify-MSG is received after the selection phase, and then a single command is received. After that, a single message is received which is related to an attention condition.
AA_H	CDB received (ATN condition detected)	A single command is received because the ATN signal is not asserted during the selection phase. After that, an attention condition is detected.

*1 Set up the SEL/RESEL setting register for phase transitions after the selection phase.
*2 Set the group 6 and 7 CDB length in the group 6/7 command length setting register.

5.8 INTERRUPT CODES AND COMMAND STEPS

The SPC reports interrupts with an interrupt code and command step which indicates the SPC operating state when the interrupt occurs.
The command step value differs depending on the SPC operating state. There are four different operating states.

- **Execution of discrete/sequential command**
- **User program operation**
- **Automatic selection response**
- **SPC in Ready state (before command reception)**

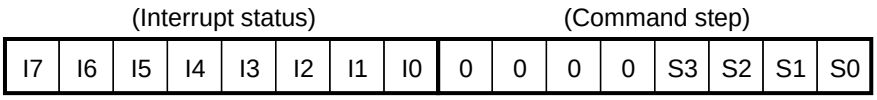
Interrupt analysis should involve reference to both the interrupt status register (address 04) and the command step register (address 05).

■ **Interrupt Code and Command Step**

The command step value differs depending on the operating state.

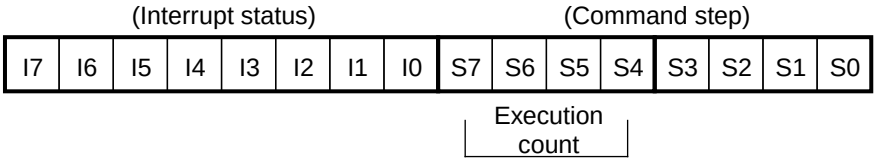
(1) Execution of discrete/sequential command

The command step defined for each command is shown in BIT 3 to 0 of the command step register.



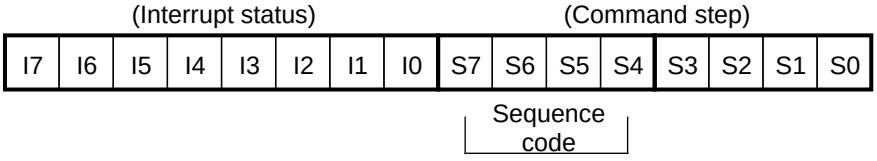
(2) User program operation

The number of discrete commands executed (number of completed commands) is shown in BIT 7 to 4 and the command step of the command where the interrupt occurs is shown in BIT 3 to 0.



(3) Automatic selection response

The response sequence type is shown in BIT 7-4 of the command step register and the sequence step where the interrupt occurs is shown in BIT 3-0.



(4) SPC in Ready state

The command step is 00_H.

(Interrupt status)								(Command step)							
I7	I6	I5	I4	I3	I2	I1	I0	0	0	0	0	0	0	0	0

Relevant interrupts are listed below.

- DISCONNECTED (70_H)
- RESET CONDITION DETECTED (01_H)
- REQ ASSERTED (71_H)
- SELECTED (80_H)
- SELECTED WITH ATN (81_H)
- RESELECTED (90_H)

See Appendix G for a list of interrupt codes and command steps.

5.9 SPC OPERATION DURING ERROR DETECTION

SPC operation when various errors are detected is described below.

■ **When an Offset Error is Detected**

When an offset error is detected, SPC operation differs depending on whether it is functioning as the initiator or the target.

See 5.9.1 for more information.

■ **When a Short Transfer Period Error is Detected**

When a Short Transfer Period error is detected, SPC operation differs depending on whether it is functioning as the initiator or the target.

See 5.9.2 for more information.

■ **When a REQ/ACK Timeout Error is Detected**

a. Initiator

When a REQ/ACK timeout error is detected, the transfer is suspended and a REQ/ACK timeout error interrupt is reported.

b. Target

When a REQ/ACK timeout error is detected, the transfer is suspended and a REQ/ACK timeout error interrupt is reported.

■ **When a SCSI Parity Error is Detected**

When a SCSI parity error is detected, SPC operation differs depending on the detected phase.

See 5.9.4 for more information.

■ **When a MPU Parity Error is Detected**

When a MPU parity error is detected, SPC operation differs depending on the detected phase.

See 5.9.5 for more information.

■ **When a DMA Parity Error is Detected**

When a DMA parity error is detected, SPC operation differs depending on whether it is functioning as the initiator or the target.

See 5.9.6 for more information.

MEMO

5.9.1 When an Offset Error is Detected

The following SPC operations take place when an offset error is detected.

a. Initiator

As soon as the offset error is detected, an ATN signal is asserted and after the transfer is completed, an offset error interrupt is reported. When the target moves to a phase other than the message out phase before the transfer is completed, an offset error and Phase-error in transfer Progress interrupts are reported.

b. Target

As soon as the offset error is detected, the transfer is suspended and after the terminate procedure an offset error interrupt is reported.

■ When an Offset Error is Detected

Table 5.9.1a and Table 5.9.1b list SPC operations when an offset error is detected (initiator, target).

Table 5.9.1a SPC Operation When an Offset Error is Detected (Initiator)

Detected phase	SPC operation
Data in/out	As soon as the offset error is detected, an ATN signal is asserted. The transfer continues but normal operation is not guaranteed. The completion report is performed as follows. • After the transfer is completed or if the target moves to a message out phase before the transfer ends, an offset error interrupt is reported. • When the target moves to a phase other than the message out phase before the transfer ends, an offset error and Phase error in transfer Progress interrupt are reported.

Table 5.9.1b SPC Operation When an Offset Error is Detected (Target)

Detected phase	SPC operation
Data in/out	When the offset error is detected, a terminate procedure is conducted and an offset error interrupt is reported (see 5.9.6 for more on the terminate procedure).

■ Points to Watch when an Offset Error is Detected

When the SPC is functioning as the initiator in the data in phase and the following phenomenon occurs, the transfer is not terminated.

(Phenomenon)

When data is received even though the target has sent a REQ signal which exceeds the specified offset value and the SPC data register is Full.

At this time, the SPC detects an offset error and an ATN signal is asserted.

The occurrence of this phenomenon can be acknowledged and dealt with by the following procedure.

(Response)

(1) When the SPC status register (address 02) Data-Request flag (Bit 2) does not come up (during program transfer) or the DREQ signal is no longer asserted (during DMA transfer) even after waiting the amount of time calculated by the following equation, realize that this phenomenon has occurred.

Equation for calculating waiting time

T_{WAIT} = transfer cycles $\times$ block length + internal operation cycle $\times$ 60

Calculation example

Data transfer speed: 10 Mbyte/sec (1 cycle = 100 nsec), block length: 512 bytes

Internal operating frequency: 5 MHz (1 cycle = 200 nsec), number of blocks: 10

T_{WAIT} = $100 \times 512 + 200 \times 10 \times 60$
= 171700 (nsec) = 171.2 (μsec)

(2) When it is realized that this phenomenon has occurred, use the `Command Pause` command to suspend the transfer.

(3) At this point, there are cases when the Data Request flag or the DREQ signal returns to 1. Initiate a MPU, DMA side transfer.

(4) Read the interrupts looking at the Command Pause (67_H) and then the offset error (29_H) interrupt.

(5) Repeat the **`SET ACK/RESET ACK`** command until the target changes phases.

5.9.2 When a Short Transfer Period Error is Detected

When a Short Transfer Period error is detected, the SPC performs the following operations.

a. Initiator

As soon as the Short Transfer Period error is detected, an ATN signal is asserted, and after the transfer is completed, a Short Transfer Period error interrupt is reported.

When the target moves to a phase other than the message out phase before the transfer ends, a Short Transfer Period and Phase error in transfer Progress interrupts are reported.

b. Target

As soon as a Short Transfer Period error is detected, the transfer is suspended and after the terminate procedure, a Short Transfer Period error interrupt is reported.

■ When a Short Transfer Period Error is Detected

Table 5.9.2a and Table 5.9.2b list SPC operations when a Short Transfer Period error (initiator, target) is detected.

Table 5.9.2a SPC Operation When a Short Transfer Period Error is Detected (Initiator)

Detected phase	SPC operation
Data in/out	As soon as the Short Transfer Period error is detected, an ATN signal is asserted. The transfer continues but normal operation is not guaranteed. The completion report is performed in the following manner. • After the transfer is completed or if the target moves to a message out phase before the transfer ends, a Short Transfer Period error interrupt is reported. • When the target moves to a phase other than the message out phase before the transfer ends, a Short Transfer Period error and Phase error in transfer Progress interrupt are reported.

Table 5.9.2b SPC Operation When a Short Transfer Period Error is Detected (Target)

Detected phase	SPC operation
Data in/out	When the Short Transfer Period error is detected, a terminate procedure is conducted and a Short Transfer Period error interrupt is reported (see 5.9.6 for more on the terminate procedure).

MEMO

5.9.3 When a SCSI Parity Error is Detected

When a SCSI parity error is detected, the SPC performs the following operations.

a. Initiator

- **When the parity error is detected in the message in phase, an ATN signal is immediately asserted. However, the transfer process differs depending on where the parity error is detected.**
- **When the parity error is detected in the status phase, an ATN signal is immediately asserted and after the transfer ends, a SCSI parity error interrupt is reported.**
- **When the parity error is detected in the data in/out phase, an ATN signal is immediately asserted and after the transfer ends, a SCSI parity error interrupt is reported. When the target moves to a phase other than the message out phase before the transfer ends, a SCSI parity error and Phase error in transfer progress interrupts are reported.**

b. Target

- **When the parity error is detected in the message in phase, the transfer is immediately suspended and a SCSI parity error interrupt is reported.**
- **When the parity error is detected in the command phase, the transfer is immediately suspended and a SCSI parity error interrupt is reported.**
- **When the parity error is detected in the data in/out phase, the SPC operation differs depending on whether it is set to byte unit stop or block unit stop.**

■ **When a SCSI Parity Error is Detected**

Table 5.9.3a and Table 5.9.3b list SPC operation when a SCSI parity error is detected (initiator, target).

Table 5.9.3a SPC Operation When a SCSI Parity Error is Detected (Initiator)

Detected phase	SPC operation
Message in	As soon as the parity error is detected, an ATN signal is asserted. However, the transfer process differs depending on the detection location. • When the message being received is a 1 byte or 2 byte message, a SCSI parity error interrupt is reported after the transfer ends. • When the message being received is an extended message, a SCSI parity error interrupt is reported after two bytes are received if the SCSI parity error is detected in the first or second byte or after all bytes are received if it is detected from the third byte on.
Status	As soon as the parity error is detected, an ATN signal is asserted and a SCSI parity error interrupt is reported after the transfer ends.
Data in/out	As soon as the parity error is detected, an ATN signal is asserted and the following reported. • After the transfer is completed or if the target moves to a message out phase before the transfer ends, a SCSI parity error interrupt is reported. • When the target moves to a phase other than the message out phase before the transfer ends, a SCSI parity error and Phase error in transfer Progress interrupt are reported.

Table 5.9.3b SPC Operation When a SCSI Parity Error is Detected (Target)

Detected phase	SPC operation
Message out	As soon as the parity error is detected, the transfer is suspended and a SCSI parity error interrupt is reported.
Command	As soon as the parity error is detected, the transfer is suspended and a SCSI parity error interrupt is reported.
Data in/out	There are two types of operation when a parity error has been detected. • When byte unit stop is specified (*1), the transfer is suspended as soon as the parity error is detected and a SCSI parity error interrupt is reported. • When block unit stop is specified (*), the transfer is suspended when the block where the parity error is detected is finished being transferred and a SCSI parity error interrupt is reported (data register ends Empty).

* See 3.3.3.4 for more on setting byte unit stop or block unit stop.

5.9.4 When a MPU Parity Error is Detected

When a MPU parity error is detected, the SPC performs the following operations.

a. Initiator

- **When the detection phase is not yet executed (during register write), as soon as the parity error is detected, a REG parity error Detected interrupt is reported.**
- **When the parity error is detected in the data in/out phase (during program transfer), an ATN signal is immediately asserted and after the transfer ends, a MPU parity error interrupt is reported. When the target moves to a phase other than the message out phase before the transfer ends, a MPU parity error and Phase error in transfer progress interrupts are reported.**

b. Target

- **When the detection phase is not yet executed (during register write), as soon as the parity error is detected, a REG parity error Detected interrupt is reported.**
- **When the parity error is detected in the data in/out phase (during program transfer), the SPC operation differs depending on whether it is set to byte unit stop or block unit stop.**

■ When a MPU Parity Error is Detected

Table 5.9.4a and Table 5.9.4b list SPC operation when a MPU parity error is detected (initiator, target).

Table 5.9.4a SPC Operation When a MPU Parity Error is Detected (Initiator)

Detected phase	SPC operation
Before execution (register write)	As soon as the parity error is detected, a REG parity error Detected interrupt is reported. This interrupt is produced when SPC operation starts (a command is received and the SPC refers to a register where the MPU has written).
Data in/out (program transfer)	As soon as the parity error is detected, an ATN signal is asserted and the following are reported. • After the transfer is completed or if the target moves to a message out phase before the transfer ends, a MPU parity error interrupt is reported. • When the target moves to a phase other than the message out phase before the transfer ends, a MPU parity error and Phase error in transfer Progress interrupt are reported.

Table 5.9.4b SPC Operation When a MPU Parity Error is Detected (Target)

Detected phase	SPC operation
Before execution (register write)	As soon as the parity error is detected, a REG parity error Detected interrupt is reported. This interrupt is produced when SPC operation starts (a command is received and the SPC refers to a register where the MPU has written).
Data in/out (program transfer)	There are two types of operation when a parity error has been detected. • When byte unit stop is specified, the transfer is suspended as soon as the parity error is detected and a MPU parity error interrupt is reported. • When block unit stop is specified, the transfer is suspended when the block where the parity error is detected is finished being transferred and a MPU parity error interrupt is reported (data register ends Empty).

* See 3.3.3.4 for more on setting byte unit stop or block unit stop.

5.9.5 When a DMA Parity Error is Detected

When a DMA parity error is detected, the SPC performs the following operations.

a. Initiator

- **As soon as the parity error is detected, an ATN signal is asserted and after the transfer ends, a DMA parity error interrupt is reported. When the target moves to a phase other than the message out phase before the transfer ends, a DMA parity error and Phase error in transfer progress interrupts are reported.**

b. Target

- **The SPC operation differs depending on whether it is set to byte unit stop or block unit stop.**

■ When a DMA Parity Error is Detected

Table 5.9.5a and Table 5.9.5b list SPC operation when a DMA parity error is detected (initiator, target).

Table 5.9.5a SPC Operation When a DMA Parity Error is Detected (Initiator)

Detected phase	SPC operation
Data in/out (DMA transfer)	As soon as the parity error is detected, an ATN signal is asserted and the following are reported. • After the transfer is completed or if the target moves to a message out phase before the transfer ends, a DMA parity error interrupt is reported. • When the target moves to a phase other than the message out phase before the transfer ends, a DMA parity error and Phase error in transfer Progress interrupt are reported.

Table 5.9.5b SPC Operation When a DMA Parity Error is Detected (Target)

Detected phase	SPC operation
Data in/out (DMA transfer)	There are two types of operation when a parity error has been detected. • When byte unit stop is specified (*1), the transfer is suspended as soon as the parity error is detected and a DMA parity error interrupt is reported. • When block unit stop is specified (*), the transfer is suspended when the block where the parity error is detected is finished being transferred and a DMA parity error interrupt is reported (data register ends Empty).

* See 3.3.3.4 for more on setting byte unit stop or block unit stop.

MEMO

5.9.6 Terminate Procedure

When an error occurs during SCSI INPUT or SCSI OUTPUT, the following terminate procedures are performed.

- **SCSI Input**

Initiator: After the error is detected and the ACK signal assert is suspended, the MPU, DMA side transfer is conducted until the data register is Empty and completed.

Target: After the error is detected and the REQ signal assert is suspended, the MPU, DMA side transfer is conducted until the data register is Empty and completed.

- **SCSI Output**

Initiator: After the error is detected and the ACK signal assert is suspended, the MPU, DMA side transfer is halted regardless of the data register state (stops with data in the data register).

Target: After the error is detected and the REQ signal assert is suspended, the MPU, DMA side transfer is halted regardless of the data register state (stops with data in the data register).

■ SCSI Input Terminate Procedure

When an error occurs during SCSI INPUT, the following terminate procedures are conducted.

a. Initiator

After the error is detected and the ACK signal assert is suspended, the MPU, DMA side transfer is conducted until the data register is Empty and completed.

For program transfers, 1 is shown in BIT 2 (DATA TRANS REQ) of the SPC status register (1), and for DMA transfers, the DREQ signal is H.

b. Target

After the error is detected and the REQ signal assert is suspended, the MPU, DMA side transfer is conducted until the data register is Empty.

For program transfers, 1 is shown in BIT 2 (DATA TRANS REQ) of the SPC status register (1), and for DMA transfers, the DREQ signal is H.

For a terminate procedure when the MPU/DMA bus width is 16 bits and an odd number of data bytes remains in the data register, a single byte of invalid data is output at the final word access.

At this time, the relationship between the data register value and the modified byte register value is as follows.

(data byte register value) + 1 = (modified byte register value)

■ SCSI OUTPUT Terminate Procedure

When an error occurs during SCSI OUTPUT, the following terminate procedures are conducted.

a. Initiator

After the error is detected and the ACK signal assert is suspended, the MPU, DMA side transfer is halted regardless of the data register state.

In this case, the transfer stops with data in the data register.

b. Target

After the error is detected and the REQ signal assert is suspended, the MPU, DMA side transfer is halted regardless of the data register state.

In this case, the transfer stops with data in the data register.

When the transfer is terminated with data remaining in the data register, this data becomes invalid.

Therefore, when resuming the transfer, count the number of bytes remaining in the register and move the data pointer back this amount.

■ Calculating the Remaining Data Register Bytes

There are cases when an error occurs that a transfer is stopped with data remaining in the data register.

In such cases, the number of bytes remaining in the data register can be calculated in the following manner.

- SCSI INPUT

  Nin = (lower 6 bit values of modified byte register) - (lower 6 bit values of data register)

  When Nin >= 0, the remaining number of bytes is N
  When Nin < 0, the remaining number of bytes is equal to N + 64

- SCSI OUTPUT

  Nout = (lower 6 bit values of data register) - (lower 6 bit values of modified byte register)

  When Nout >= 0, the remaining number of bytes is N
  When Nout < 0, the remaining number of bytes is equal to N + 64

5.10 DIAGNOSIS MODE

By issuing the DIAG START command, the SPC enters the Diagnosis mode.

In the Diagnosis mode, the SPC is cut off from the SCSI bus so that signals cannot be sent to the SCSI nor can control signals be received from the SCSI. Internal control operations can be executed and interface operations between the SPC and the SCSI can be simulated.

■ SCSI Control Signals

In the Diagnosis mode, the values in the Diagnostic register (address 11) are treated as input signals to the SPC.

Also, output signals from the SPC are shown in the SCSI control signal status register (address 11).

The SCSI control signal status register displays both artificial input from the SCSI and artificial output to the SCSI.

■ SCSI Data Bus Signals

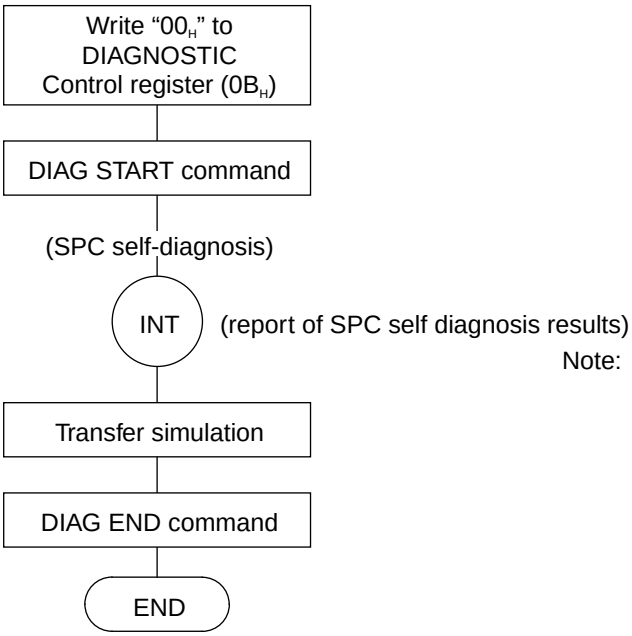
In the Diagnosis mode, SCSI data bus artificial input and output data can be established.

The artificial input and output data settings vary depending on the phase being executed.

In the Diagnosis mode, the SCSI data parity bit is fixed as 0.

■ Diagnosis Mode Execution

Figure 5.10 shows the Diagnosis mode flowchart.



Note: Execute the DIAG START command after the power is turned on and BIT 6, 7 (selectionresponse/reselection response) of the Response mode setting register is set to Disable by the SET UP REG command. Do not execute the DIAG START or the DIAG END commands when BIT 6, 7 are set to Enable or a nexus is established with another device.

Figure 5.10 Diagnosis Mode Execution

5.10.1 SCSI Control Signals

In the Diagnosis mode, the values in the Diagnostic register (address 11) are treated as input signals to the SPC.

Also, output signals from the SPC are shown in the SCSI control signal status register (address 11).

The SCSI control signal status register displays both artificial input from the SCSI and artificial output to the SCSI.

■ Flow of SCSI Control Signals During Diagnosis Mode

Figure 5.10.1 shows the flow of SCSI control signals during the diagnosis mode.

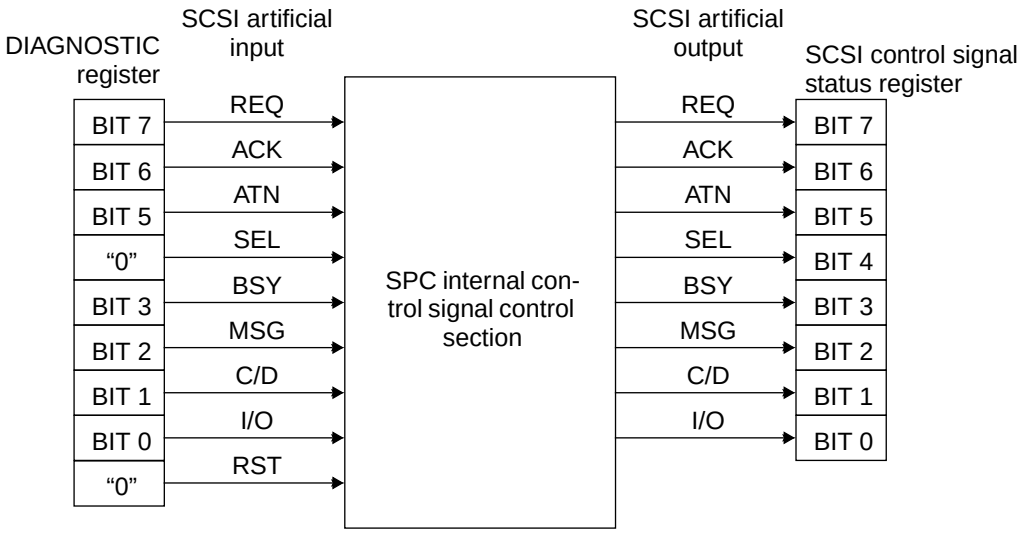


Figure 5.10.1 Flow of SCSI Control Signals During Diagnosis Mode

5.10.2 SCSI Data Bus Signal

In the Diagnosis mode, SCSI data bus artificial input/output data can be set.

The artificial input/output data setting differs depending on the execution phase.

- **In the Diagnosis mode, parity bit input is fixed at 0.**

■ **Sending Messages, Commands, and Statuses**

Send the command after writing the data to be sent in the SEND MCS buffer as in standard modes.

Artificial output data for the SCSI bus is stored in the RECEIVE MCS buffer. The sending process is diagrammed in Figure 5.10.2a.

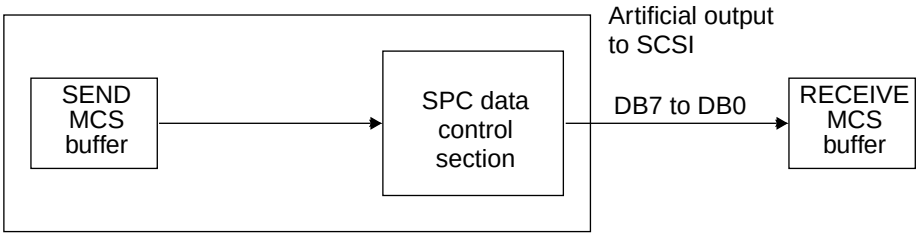


Figure 5.10.2a Sending Messages, Commands, and Statuses in Diagnosis Mode

■ **Receiving Messages, Commands, and Statuses**

Send the command after writing the artificial input data from the SCSI bus in the SEND MCS buffer.

Data received is stored in the RECEIVE MCS buffer as in standard modes. The receiving process is diagrammed in Figure 5.10.2b.

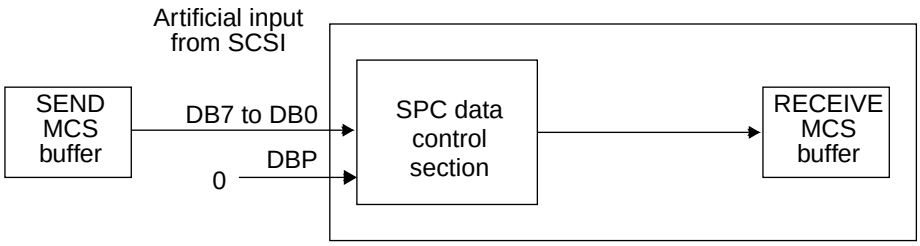


Figure 5.10.2b Receiving Messages, Commands, and Statuses in Diagnosis Mode

■ **Sending Data**

Artificial output data is stored in the RECEIVE MCS buffer.

When the transfer data is 33 bytes or longer, data storage to the RECEIVE MCS buffer is conducted repeatedly.

The synchronous transfer mode can be executed during the data phase, and at this time, the transfer parameters are the values set in the self ID number. After setting the self ID number in the SEL/RESEL ID register, establish the transfer parameters. This sending process is diagrammed in Figure 5.10.2c.

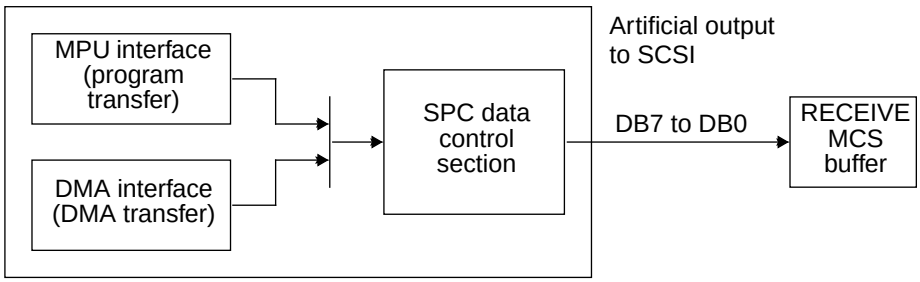


Figure 5.10.2c Sending Data in Diagnosis Mode

■ **Receiving Data**

Send the command after writing SCSI artificial input data in the SEND MCS buffer.

When the transfer data is 33 bytes or longer, the SEND MCS buffer data is used repeatedly as the artificial input data.

The synchronous transfer mode can be executed during the data phase, and at this time, the transfer parameters are the values set in the self ID number. After setting the self ID number in the SEL/RESEL ID register, establish the transfer parameters. This receiving process is diagrammed in Figure 5.10.2d.

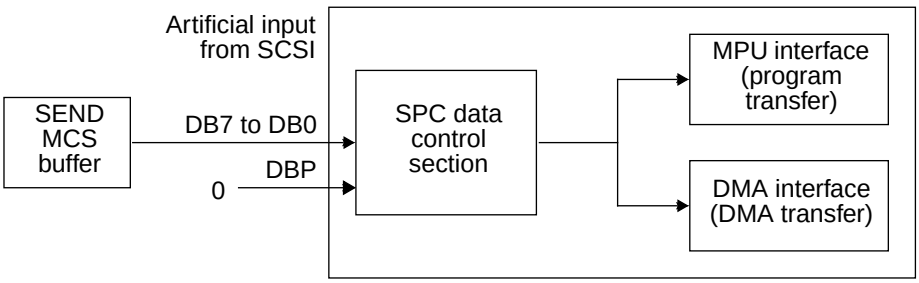


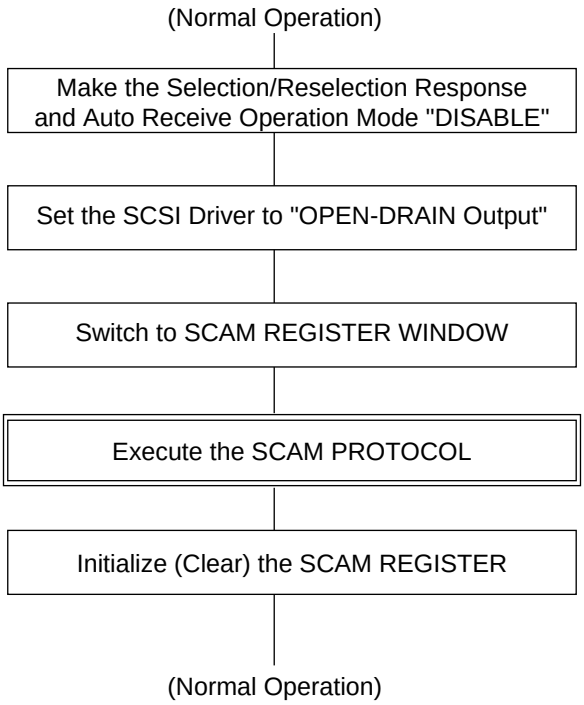
Figure 5.10.2d Receiving Data in Diagnosis Mode

5.11 SCAM FUNCTION (only for MB86604L)

The MB86604L supports the SCAM function that allows the SCAM Level-1 protocol for SCSI Plug & Play. It can directly control the SCSI bus signals from the system bus. The controlled signals are defined and set in the SCAM Register Window.

SCAM Protocol

To execute the SCAM protocol, take the following steps and set the MB86604L.



How to Access SCSI Bus

1) Write: Set the SCSI bus to Open-drain mode for the SCAM protocol. In the open-drain mode, the SPC outputs Low level at the signal assertion (writing 1 to the register).

2) Read: First, initialize the SCAM register (write 0 adn make Hi-Z state) due to the open-drain mode.

CHAPTER 6 SYSTEM CONFIGURATION EXAMPLE

This shows an SPC system configuration example where the SPC is used to develop a product.

6.1 TP SIGNAL CONTROL

The TP (transfer permission) signal is used to suspend DMA transfers temporarily.

The SPC manages transfers by individual blocks, checking the TP signal for each block and determining whether or not to send the next block.

When the DMA transfer is temporarily stopped, input 0 during the transfer of the suspended block.

■ TP Signal Control

Figure 6.1 shows the operation flowchart for cases where the DMA transfer is temporarily stopped after the transfer of block (N + 1) is completed.

During this state in which the transfer is temporarily stopped, the SPC functions in the following manner.

Initiator:REQ signals are received from the target, but ACK signals are not asserted. The DREQ signal outputs L. During the data in phase, the amount of data received is equivalent to the number of REQ received.

Target:The REQ signal is not asserted. The DREQ signal outputs L (signifying that the data register is EMPTY).

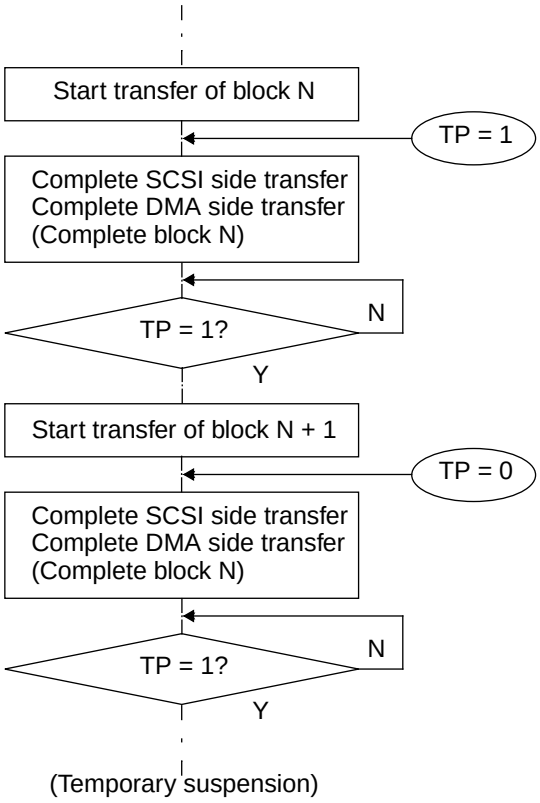


Figure 6.1 Flowchart for TP Signal Control Example

6.2 DMA MISALIGNMENT PROCESSING

For DMA misalignment processing, set “1” in BIT 7 of the SEL/RESEL ID register.

As a result, input and output data can be byte accessed from an odd number memory address.

■ DMA Misalignment Processing

Figure 6.2 provides an example of DMA misalignment processing when the first transfer is an odd number of bytes and a memory misalignment occurs for the second transfer.

In this case, set 1 in BIT 7 of the SEL/RESEL ID register for the second transfer.

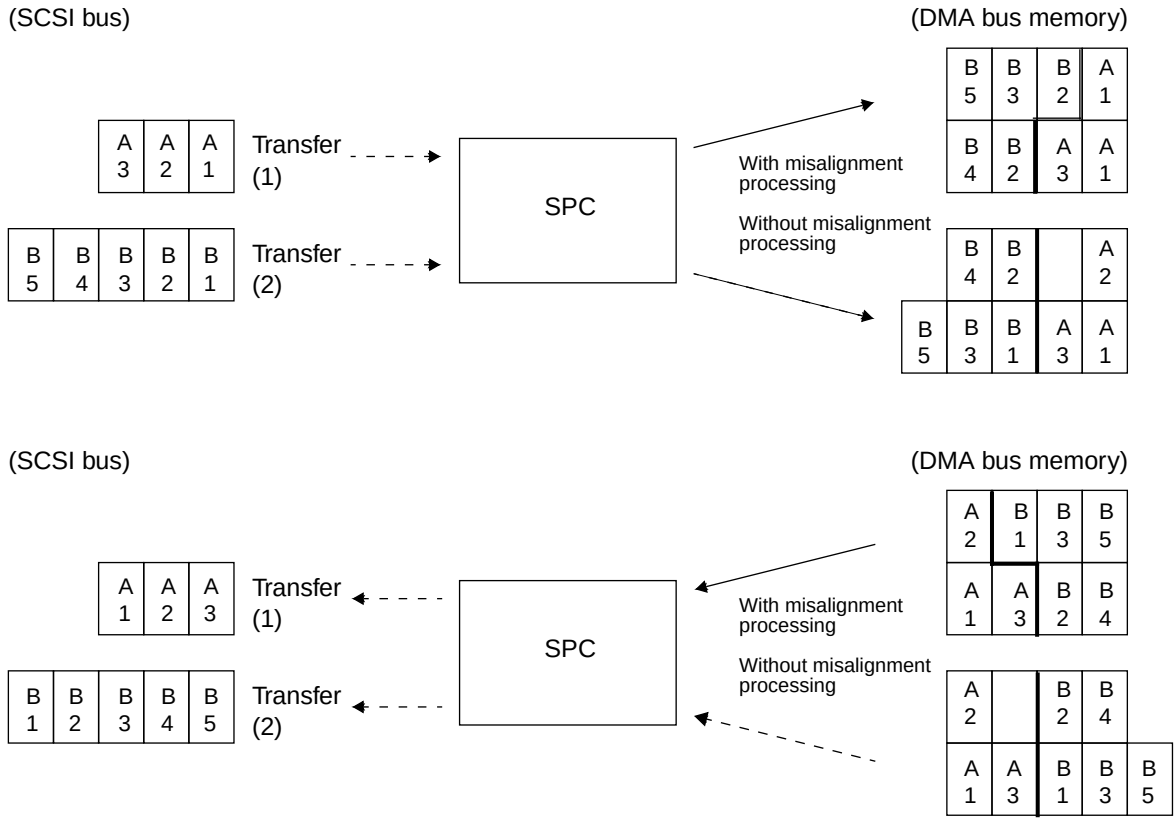


Figure 6.2 DMA Misalignment Processing Example

6.3 SYSTEM CONFIGURATION EXAMPLE 1 (80 SERIES, SEPARATE BUS TYPE)

Figure 6.3 shows a system configuration example for a SCSI-2 80 series, separate bus type system.

■ **System Configuration Example 1 (80 Series, Separate Bus Type)**

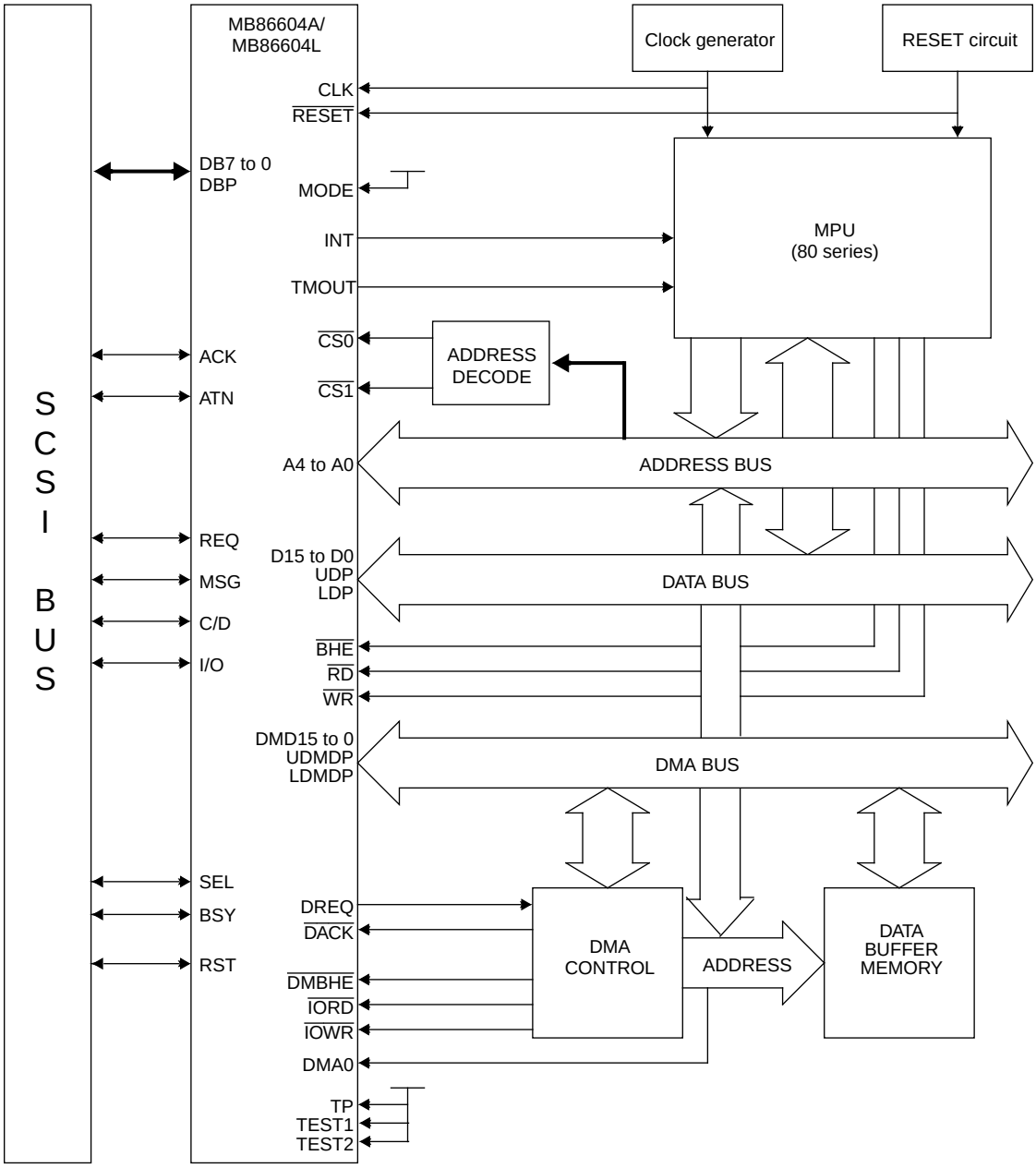


Figure 6.3 System Configuration Example 1 (80 Series, Separate Bus Type)

6.4 SYSTEM CONFIGURATION EXAMPLE 2 (68 SERIES, SEPARATE BUS TYPE)

Figure 6.4 shows a system configuration example for a SCSI-2 68 series, separate bus type system.

■ **System Configuration Example 2 (68 Series, Separate Bus Type)**

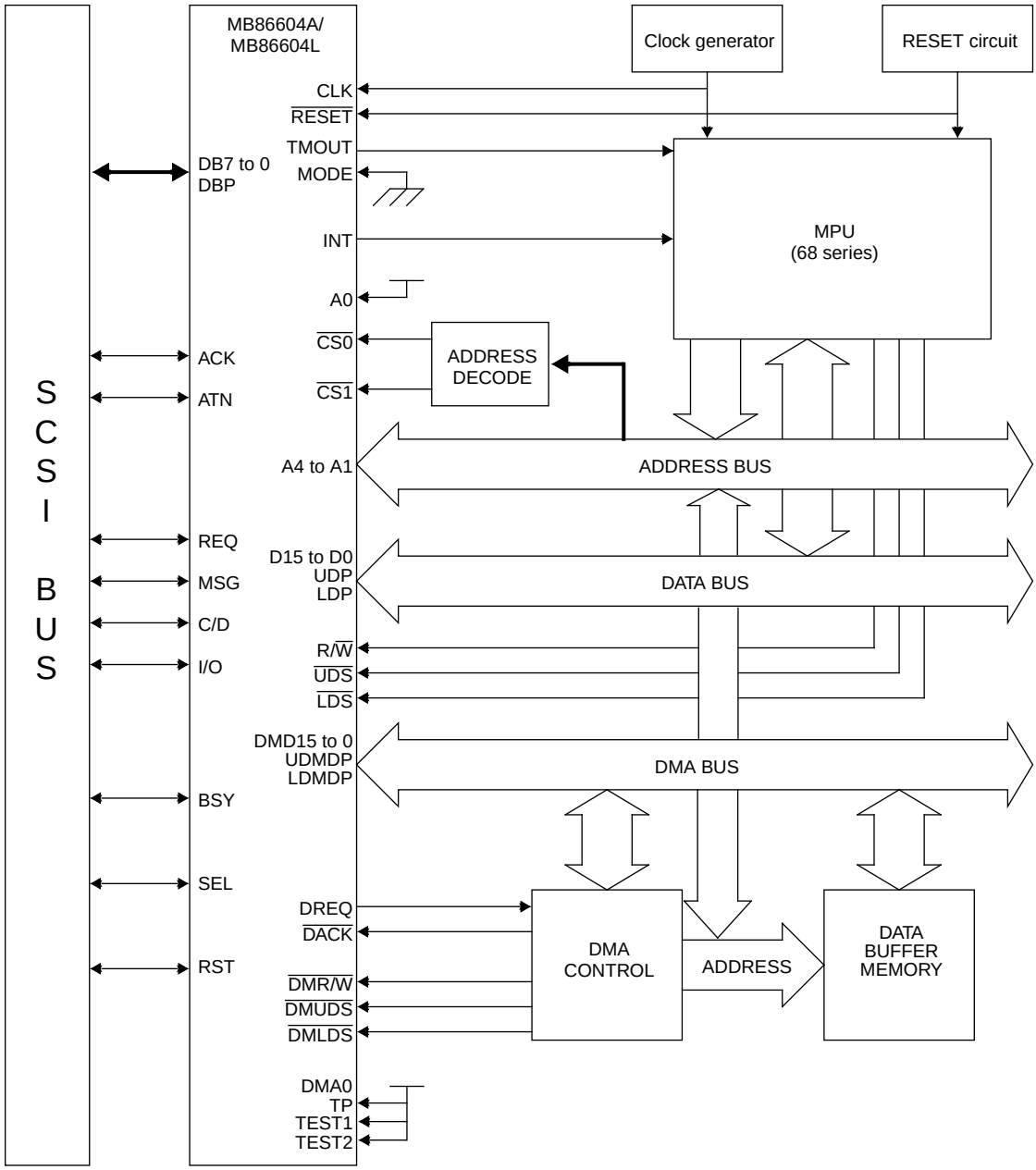


Figure 6.4 System Configuration Example 2 (68 Series, Separate Bus Type)

6.5 SYSTEM CONFIGURATION EXAMPLE 3 (80 SERIES, COMMON BUS TYPE)

Figure 6.5 shows a system configuration example for a SCSI-2 80 series, common bus type system.

■ **System Configuration Example 3 (80 Series, Common Bus Type)**

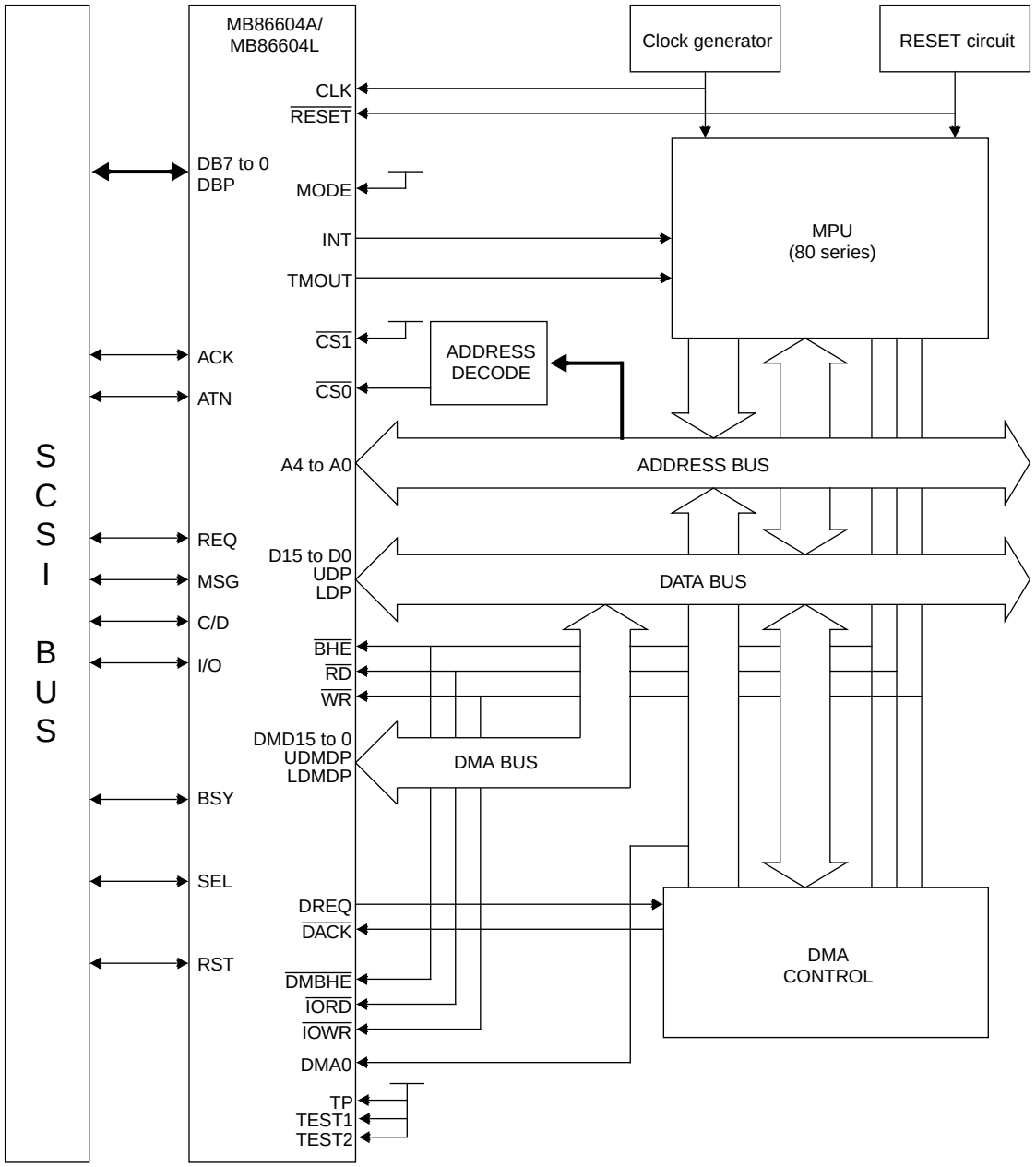


Figure 6.5 System Configuration Example 3 (80 Series, Common Bus Type)

6.6 SYSTEM CONFIGURATION EXAMPLE 4 (68 SERIES, COMMON BUS TYPE)

Figure 6.6 shows a system configuration example for a SCSI-2 68 series, common bus type system.

■ **System Configuration Example 4 (68 Series, Common Bus Type)**

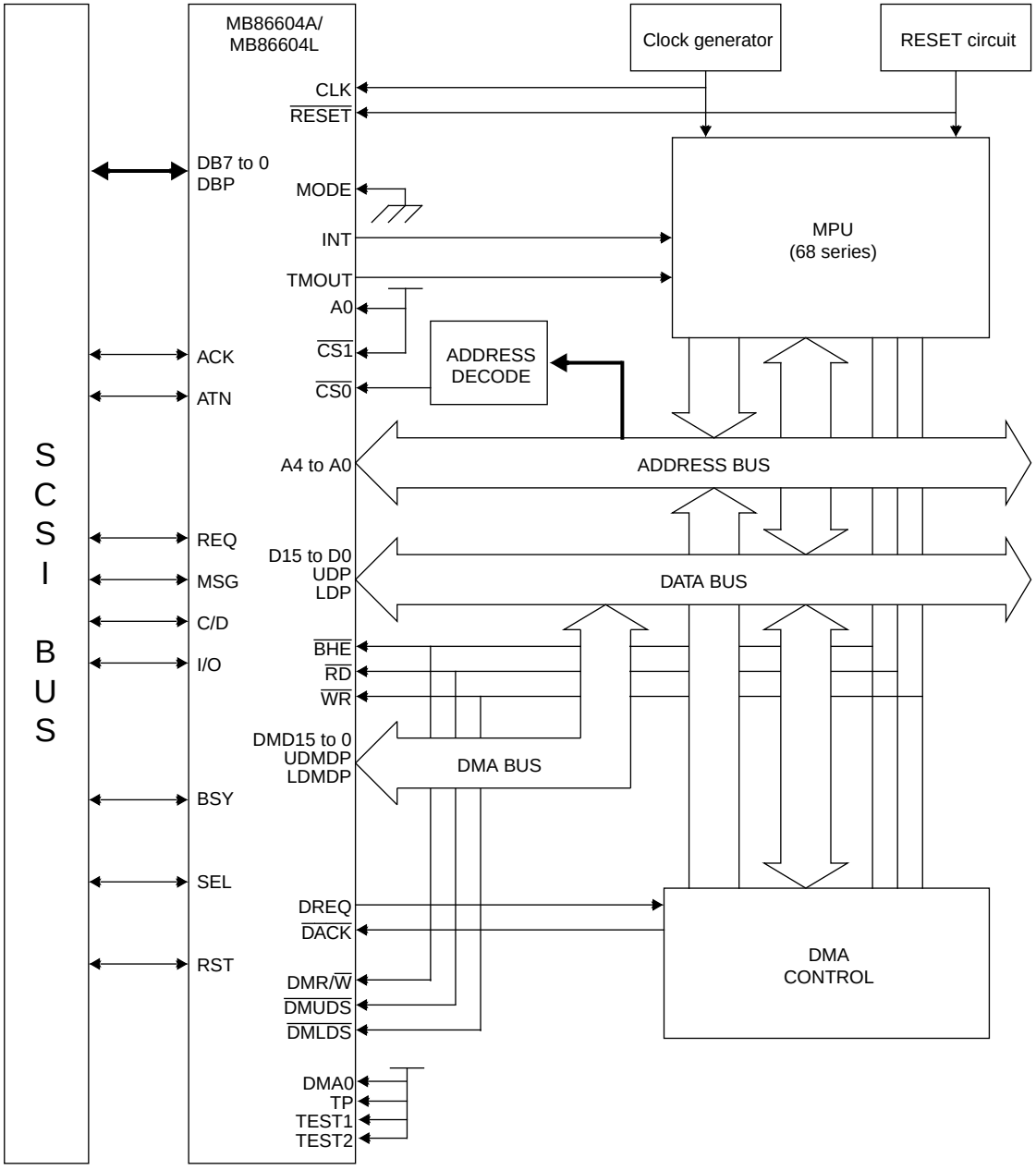


Figure 6.6 System Configuration Example 4 (68 Series, Common Bus Type)

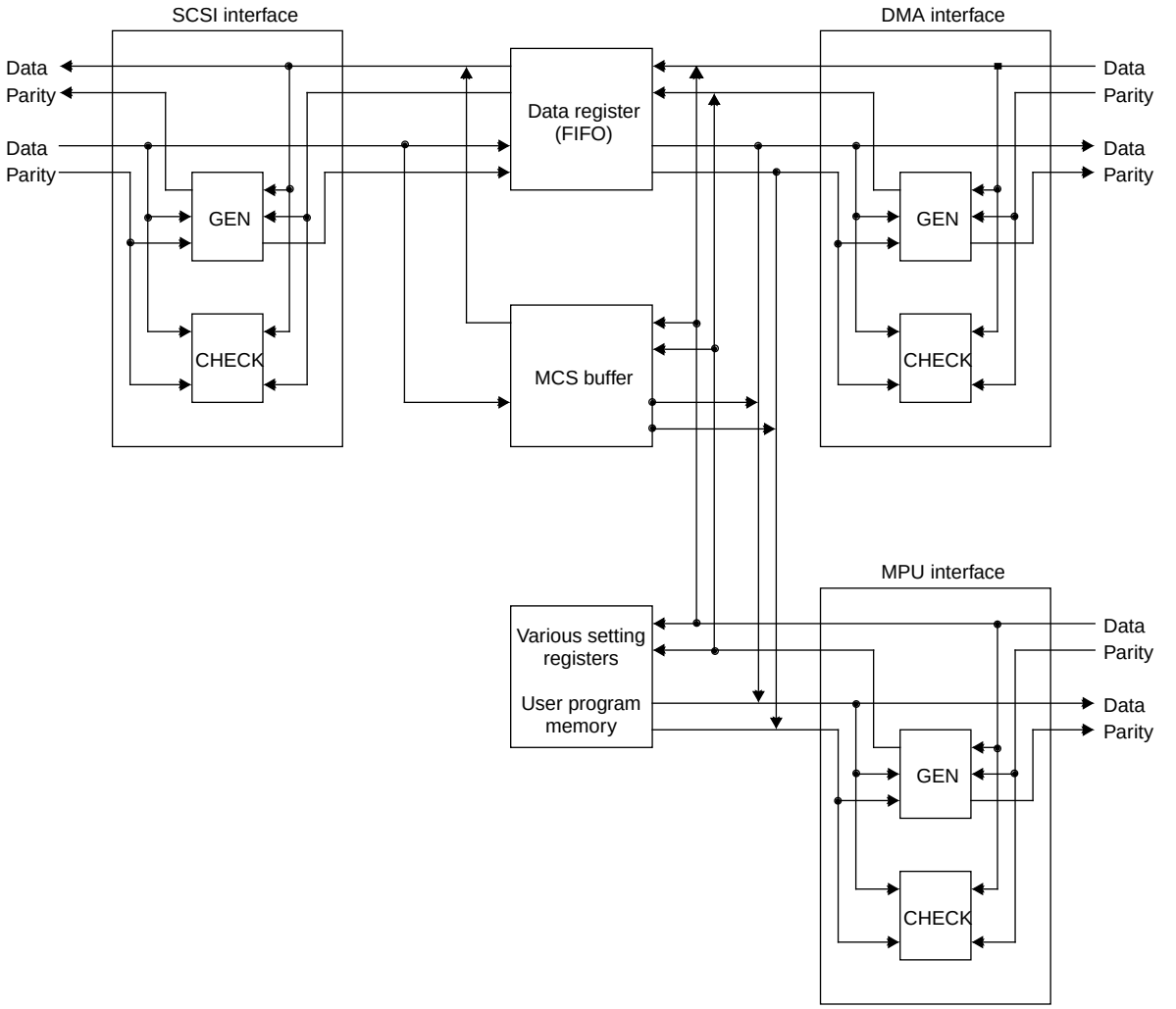
APPENDIX

APPENDIX A PACKAGE DIMENSIONS

■ MB86604A Package Dimensions (QFP100)

■ MB86604L Package Dimensions (SQFP 100)

APPENDIX B PARITY CHECK AND PARITY GENERATE FLOWS



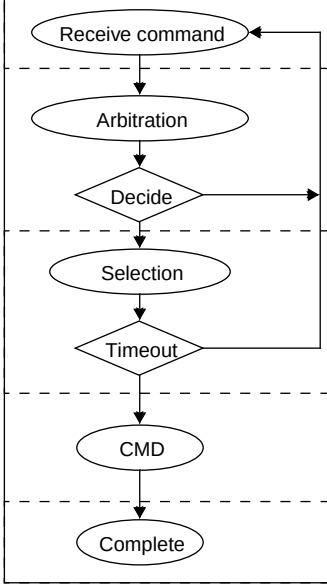
Note: For systems which do not have parity bits on the data bus, respectively pull up or pull down the UDP, LDP pins or the UDMP, LDMP pins with about 10 kiloohms of resistance.

When SCSI data bus parity checks are performed, in addition to the pin processing indicated above, enable MPU and DMA parity generate in the parity error detect setting register.

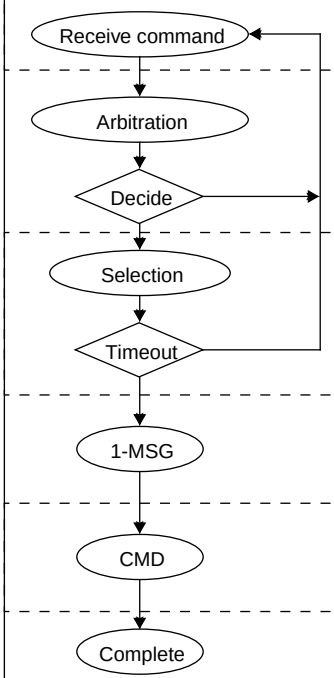
Appendix Figure B Parity Check and Parity Generate Flows

APPENDIX C INITIATOR COMMAND STEP CODES

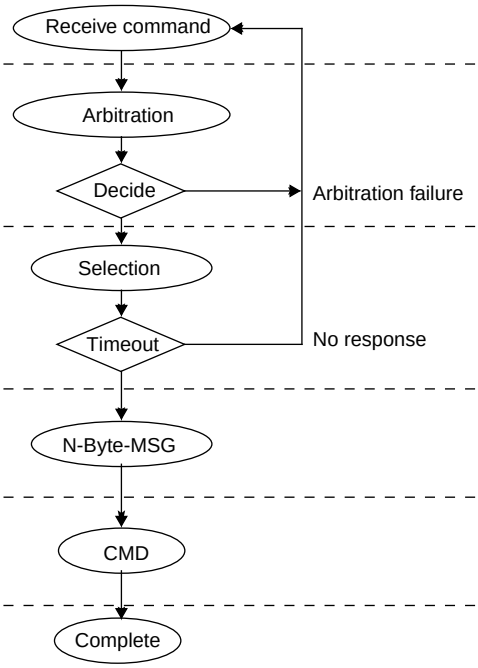
- No.1 SELECT & CMD

Flowchart	Step	Description
	0	* Receive command Wait for BUS FREE phase
	1	* Detect BUS FREE phase During Arbitration
	2	* Acquire bus usage rights During Selection
	3	* Establish a nexus Wait for CMD phase Executing CMD phase
	4	* Complete sending all CDB bytes Command complete

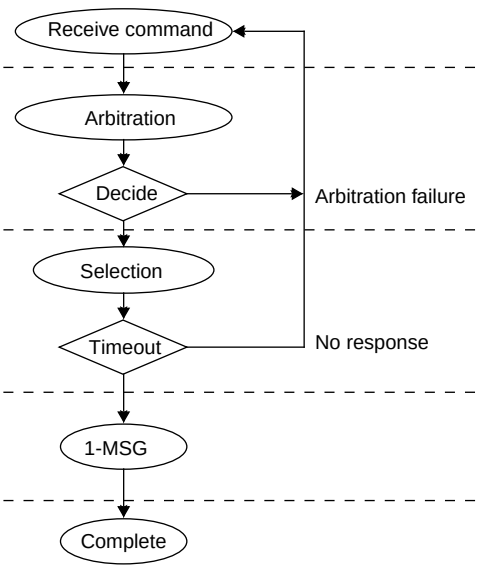
- No.2 SELECT & 1-MSG & CMD

Flowchart	Step	Description
	0	* Receive command Wait for BUS FREE phase
	1	* Detect BUS FREE phase During Arbitration
	2	* Acquire bus usage rights During Selection
	3	* Establish a nexus Wait for MSG OUT phase Executing MSG OUT phase
	4	* Complete sending all MSG bytes Wait for CMD phase Executing CMD phase
	5	* Complete sending all CDB bytes Command complete

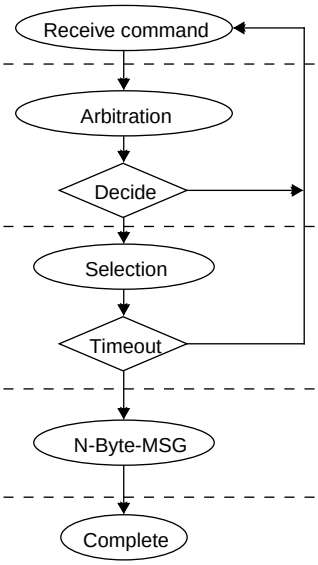
- No.3 SELECT & N-Byte-MSG & CMD

Flowchart	Step	Description
	0	* Receive command Wait for BUS FREE phase
	1	* Detect BUS FREE phase During Arbitration
	2	* Acquire bus usage rights During Selection
	3	* Establish a nexus Wait for MSG OUT phase Executing MSG OUT phase
	4	* Complete sending all MSG bytes Wait for CMD phase Executing CMD phase
	5	* Complete sending all CDB bytes Command complete

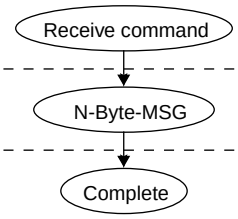
- No.4 SELECT & 1-MSG

Flowchart	Step	Description
	0	* Receive command Wait for BUS FREE phase
	1	* Detect BUS FREE phase During Arbitration
	2	* Acquire bus usage rights During Selection
	3	* Establish a nexus Wait for MSG OUT phase Executing MSG OUT phase
	4	* Complete sending all MSG bytes Command complete

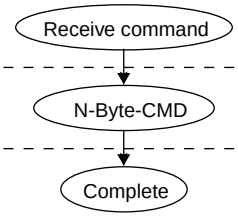
- No.5 SELECT & N-Byte-MSG

Flowchart	Step	Description
	0	* Receive command Wait for BUS FREE phase
	1	* Detect BUS FREE phase During Arbitration
	2	* Acquire bus usage rights During Selection
	3	* Establish a nexus Wait for MSG OUT phase Executing MSG OUT phase
	4	* Complete sending all MSG bytes Command complete

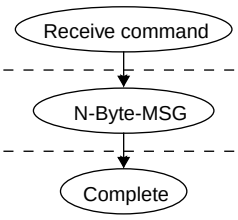
- No.6 SEND N-Byte-MSG

Flowchart	Step	Description
	0	* Receive command
	1	* Wait for MSG OUT phase Executing MSG OUT phase
	2	* Complete sending all MSG bytes Command complete

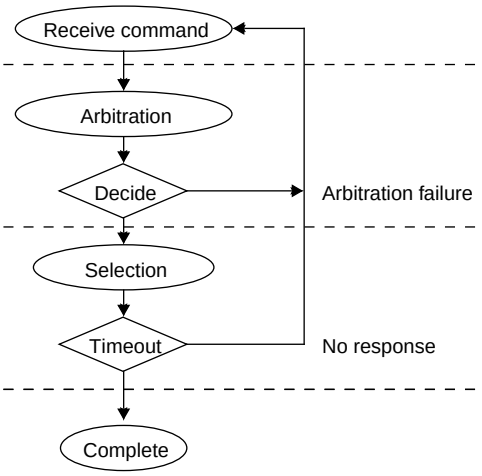
- No.7 SEND N-Byte-CMD

Flowchart	Step	Description
	0	* Receive command
	1	* Wait for CMD phase Executing CMD phase
	2	* Complete sending all CBD bytes Command complete

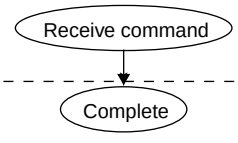
- No.8 Receive & N-Byte-MSG

Flowchart	Step	Description
	0	* Receive command
	1	* Wait for MSG IN phase Executing MSG IN phase
	2	* Complete reception of all MSG bytes Command complete

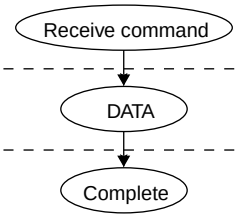
- No.9 SELECT
- No.10 SELECT with ATN

Flowchart	Step	Description
	0	* Receive command Wait for BUS FREE phase
	1	* Detect BUS FREE phase During Arbitration
	2	* Acquire bus usage rights During Selection
	3	* Establish a nexus Command complete

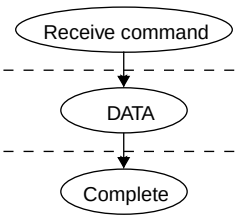
- No.11 SET ATN
- No.12 RESET ATN
- No.13 SET ACK
- No.14 RESET ACK

Flowchart	Step	Description
	0	* Receive command
	1	Command complete

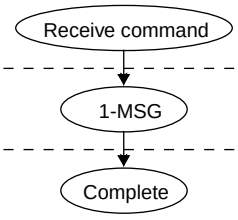
- No.15 SEND DATA from MPU
- No.16 SEND DATA from DMA
- No.19 SEND DATA from MPU (PADDING)
- No.20 SEND DATA from DMA (PADDING)

Flowchart	Step	Description
	0 1 2	* Receive command * Wait for DATA OUT phase Executing DATA OUT phase * Complete sending all DATA bytes Command complete

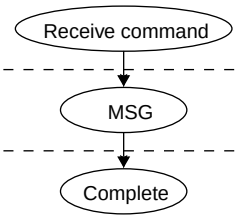
- No.17 RECEIVE DATA to MPU
- No.18 RECEIVE DATA to DMA
- No.21 RECEIVE DATA to MPU (PADDING)
- No.22 RECEIVE DATA to DMA (PADDING)

Flowchart	Step	Description
	0 1 2	* Receive command * Wait for DATA IN phase Executing DATA IN phase * Complete reception of all DATA byte Command complete

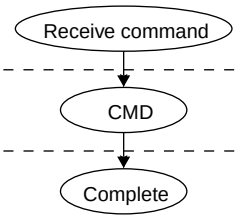
- No.23 SEND 1-MSG
- No.24 SEND 1-MSG with ATN

Flowchart	Step	Description
	0 1 2	* Receive command * Wait for MSG OUT phase Executing MSG OUT phase * Complete sending all MSG bytes Command complete

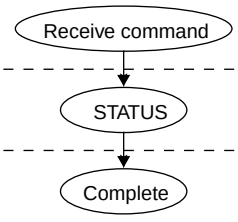
- No.25 RECEIVE MSG

Flowchart	Step	Description
	0	* Receive command
	1	* Wait for MSG IN phase Executing MSG IN phase
	2	* Complete reception of all MSG bytes Command complete

- No.26 SEND CMD

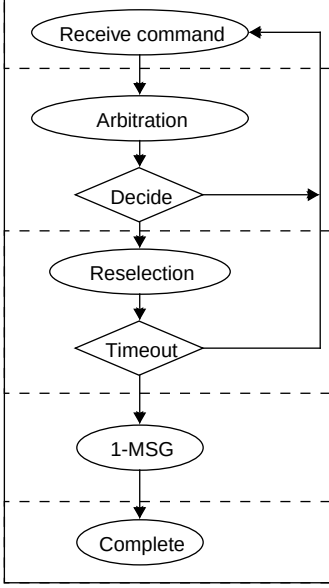
Flowchart	Step	Description
	0	* Receive command
	1	* Wait for CMD phase Executing CMD phase
	2	* Complete sending all CDB bytes Command complete

- No.27 RECEIVE STATUS

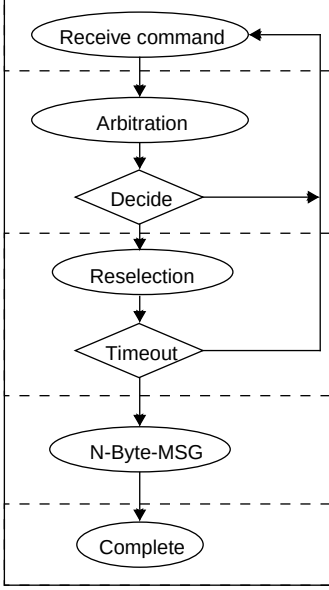
Flowchart	Step	Description
	0	* Receive command
	1	* Wait for STATUS phase Executing STATUS phase
	2	* Complete reception of STATUS Command complete

APPENDIX D TARGET COMMAND STEP CODES

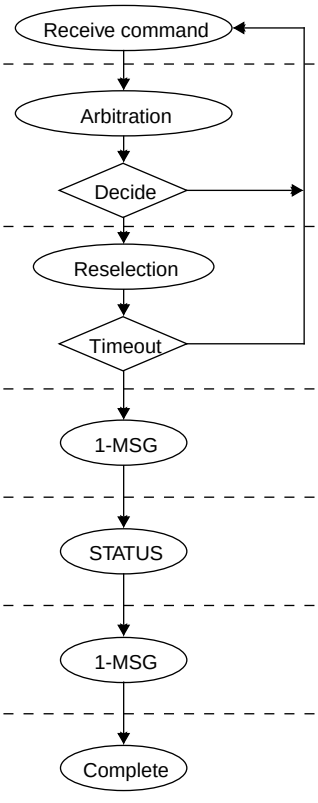
- No.1 RESELECT & 1-MSG

Flowchart	Step	Description
	0	* Receive command Wait for BUS FREE phase
	1	* Detect BUS FREE phase During Arbitration
	2	* Acquire bus usage rights During Reselection
	3	* Establish a nexus Move to MSG IN phase Executing MSG IN phase
	4	* Complete sending all MSG bytes Command complete

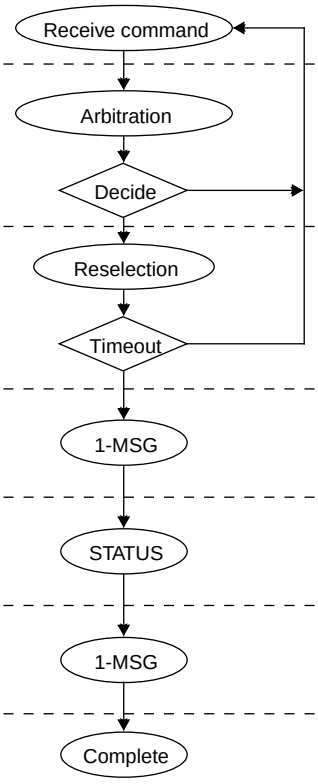
- No.2 RESELECT & N-Byte-MSG

Flowchart	Step	Description
	0	* Receive command Wait for BUS FREE phase
	1	* Detect BUS FREE phase During Arbitration
	2	* Acquire bus usage rights During Reselection
	3	* Establish a nexus Move to MSG IN phase Executing MSG IN phase
	4	* Complete sending all MSG bytes Command complete

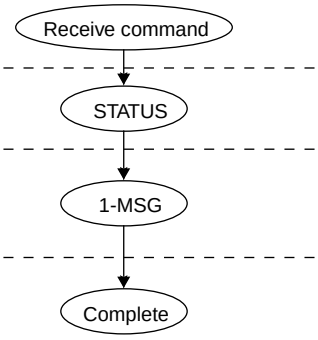
- No.3 RESELECT & 1-MSG & TERMINATE

Flowchart	Step	Description
	0	* Receive command Wait for BUS FREE phase
	1	* Detect BUS FREE phase During Arbitration
	2	* Acquire bus usage rights During Reselection
	3	* Establish a nexus Move to MSG IN phase Executing MSG IN phase
	4	* Complete sending MSG Move to STATUS phase Executing STATUS phase
	5	* Complete sending STATUS Move to MSG IN phase Executing MSG IN phase
	6	* Complete sending MSG Disconnect Command complete

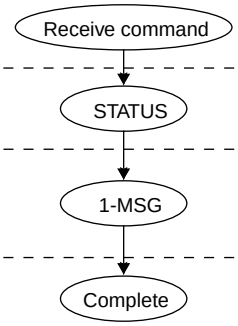
- No.4 RESELECT & 1-MSG & LINK TERMINATE

Flowchart	Step	Description
	0	* Receive command Wait for BUS FREE phase
	1	* Detect BUS FREE phase During Arbitration
	2	* Acquire bus usage rights During Reselection
	3	* Establish a nexus Move to MSG IN phase Executing MSG IN phase
	4	* Complete sending MSG Move to STATUS phase Executing STATUS phase
	5	* Complete sending STATUS Move to MSG IN phase Executing MSG IN phase
	6	* Complete sending MSG Command complete

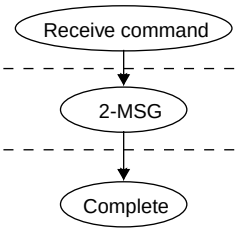
- No.5 TERMINATE

Flowchart	Step	Description
	0	* Receive command
	1	* Move to STATUS phase Executing STATUS phase
	2	* Complete sending STATUS Move to MSG IN phase Executing MSG IN phase
	3	* Complete sending MSG Disconnect Command complete

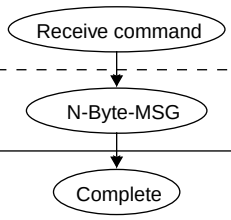
- No.6 LINK TERMINATE

Flowchart	Step	Description
	0	* Receive command
	1	* Move to STATUS phase Executing STATUS phase
	2	* Complete sending STATUS Move to MSG IN phase Executing MSG IN phase
	3	* Complete sending MSG Command complete

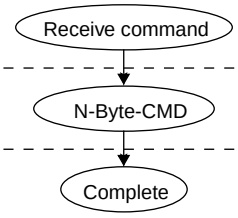
- No.7 DISCONNECT SEQUENCE

Flowchart	Step	Description
	0	* Receive command
	1	* Move to MSG IN phase Executing MSG IN phase
	2	* Complete sending 2-byte MSG Disconnect Command complete

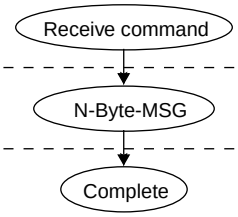
- No.8 SEND N-Byte-MSG

Flowchart	Step	Description
	0	* Receive command
	1	* Move to MSG IN phase Executing MSG IN phase
	2	* Complete sending all MSG bytes Command complete

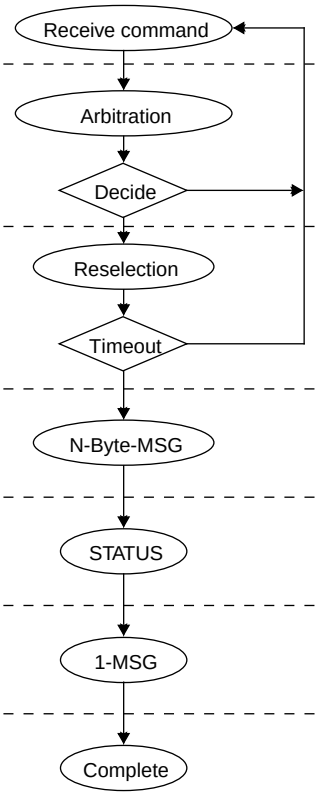
- No.9 RECEIVE N-Byte-CMD

Flowchart	Step	Description
	0	* Receive command
	1	* Move to CMD phase Executing CMD phase
	2	* Complete reception of all CDB bytes Command complete

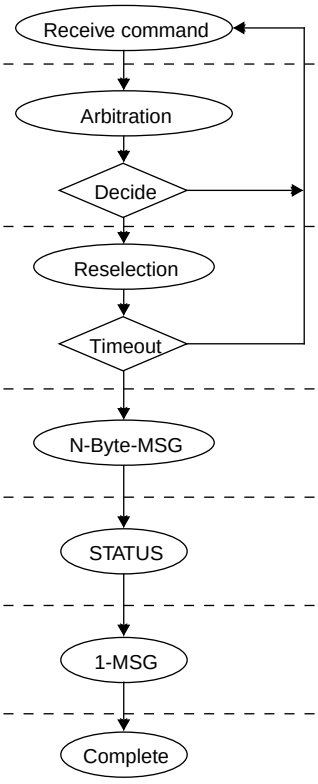
- No.10 RECEIVE N-Byte-MSG

Flowchart	Step	Description
	0	* Receive command
	1	* Move to MSG OUT phase Executing MSG OUT phase
	2	* Complete reception of all MSG bytes Command complete

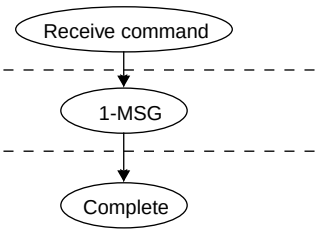
- No.11 RESELECT & N-Byte-MSG & TERMINATE

Flowchart	Step	Description
	0	* Receive command Wait for BUS FREE phase
	1	* Detect BUS FREE phase During Arbitration
	2	* Acquire bus usage rights During Reselection
	3	* Establish a nexus Move to MSG IN phase Executing MSG IN phase
	4	* Complete sending MSG Move to STATUS phase Executing STATUS phase
	5	* Complete sending STATUS Move to MSG IN phase Executing MSG IN phase
	6	* Complete sending MSG Disconnect Command complete

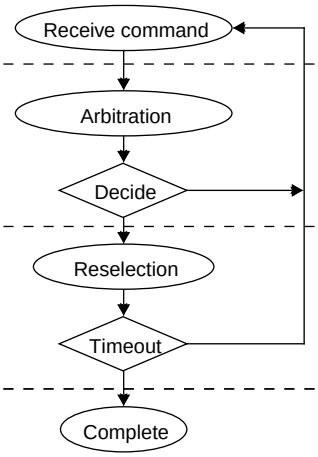
- No.12 RESELECT & N-Byte-MSG & LINK TERMINATE

Flowchart	Step	Description
	0	* Receive command Wait for BUS FREE phase
	1	* Detect BUS FREE phase During Arbitration
	2	* Acquire bus usage rights During Reselection
	3	* Establish a nexus Move to MSG IN phase Executing MSG IN phase
	4	* Complete sending MSG Move to STATUS phase Executing STATUS phase
	5	* Complete sending STATUS Move to MSG IN phase Executing MSG IN phase
	6	* Complete sending MSG Command complete

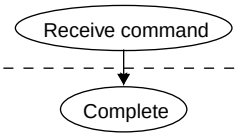
- No.13 DISCONNECT SEQUENCE 2

Flowchart	Step	Description
	0	* Receive command
	1	* Move to MSG IN phase Executing MSG IN phase
	2	* Complete sending 1-byte MSG Disconnect Command complete

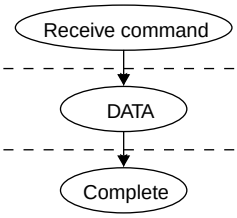
- No.14 RESELECT

Flowchart	Step	Description
	0	* Receive command Wait for BUS FREE phase
	1	* Detect BUS FREE phase During Arbitration
	2	* Acquire bus usage rights During Reselection
	3	* Establish a nexus Command complete

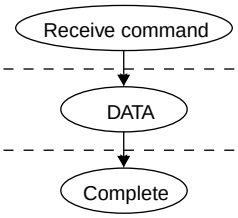
- No.15 SET REQ
- No.16 RESET REQ
- No.17 DISCONNECT

Flowchart	Step	Description
	0	* Receive command
	1	C. Command complete

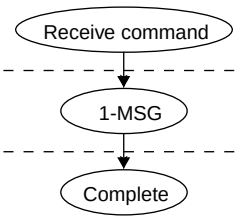
- No.18 SEND DATA from MPU
- No.19 SEND DATA from DMA

Flowchart	Step	Description
	0	* Receive command
	1	* Move to DATA IN phase Executing DATA IN phase
	2	* Complete sending all DATA bytes Command complete

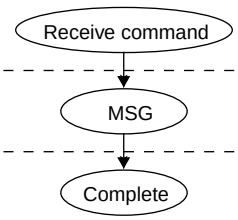
- No.20 RECEIVE DATA to MPU
- No.21 RECEIVE DATA to DMA

Flowchart	Step	Description
	0	* Receive command
	1	* Move to DATA OUT phase Executing DATA OUT phase
	2	* Complete sending all DATA bytes Command complete

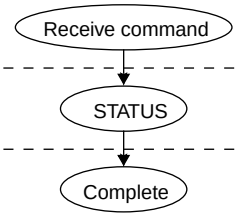
- No.22 SEND 1-MSG

Flowchart	Step	Description
	0	* Receive command
	1	* Move to MSG IN phase Executing MSG IN phase
	2	* Complete sending all MSG bytes Command complete

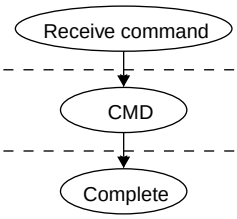
- No.23 RECEIVE MSG

Flowchart	Step	Description
	0	* Receive command
	1	* Move to MSG IN phase Executing MSG IN phase
	2	* Complete receiving all MSG bytes Command complete

- No.24 SEND STATUS

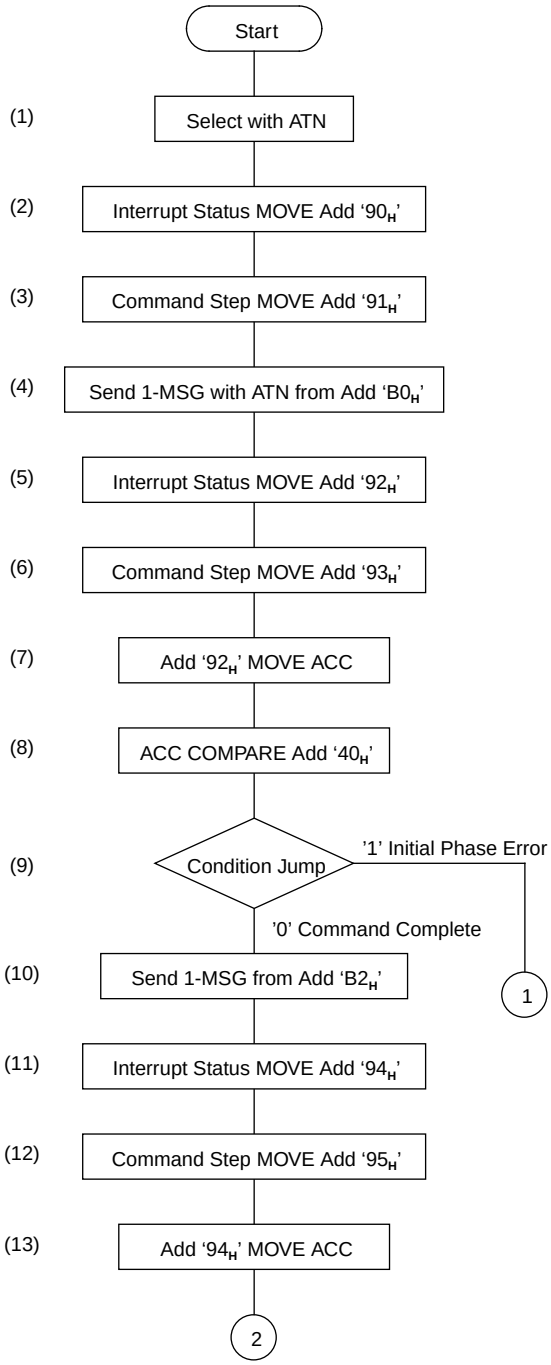
Flowchart	Step	Description
	0	* Receive command
	1	* Move to STATUS phase Executing STATUS phase
	2	* Complete sending STATUS Command complete

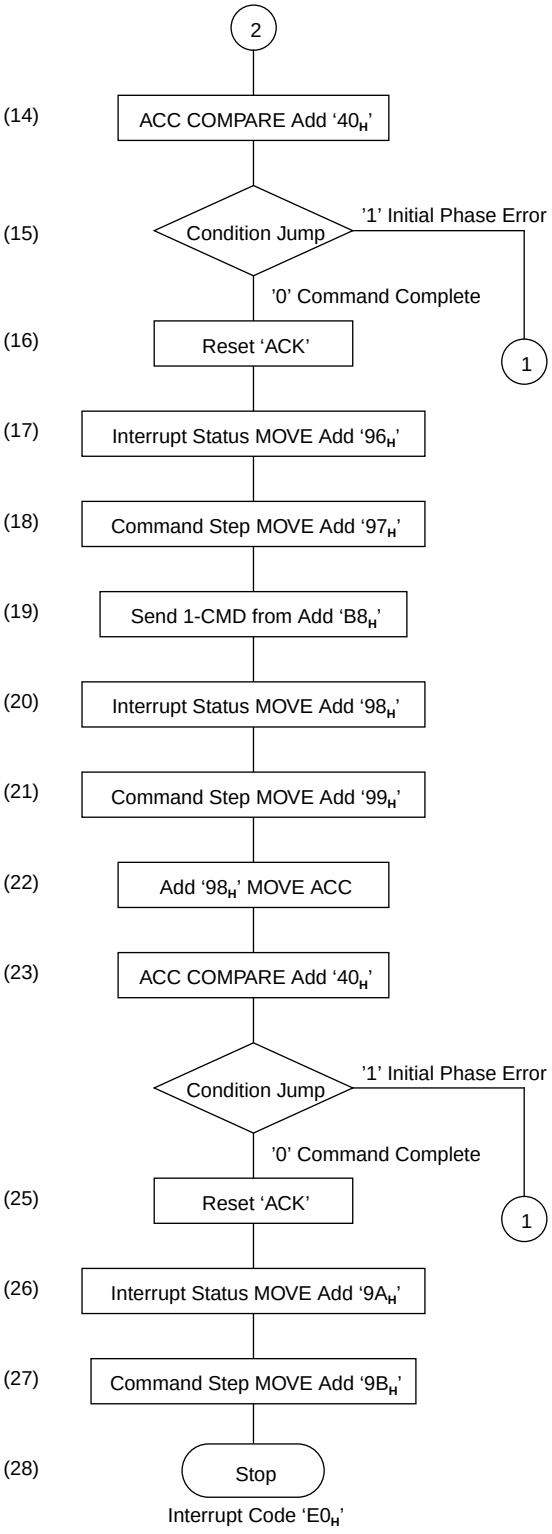
- No.25 RECEIVE CMD

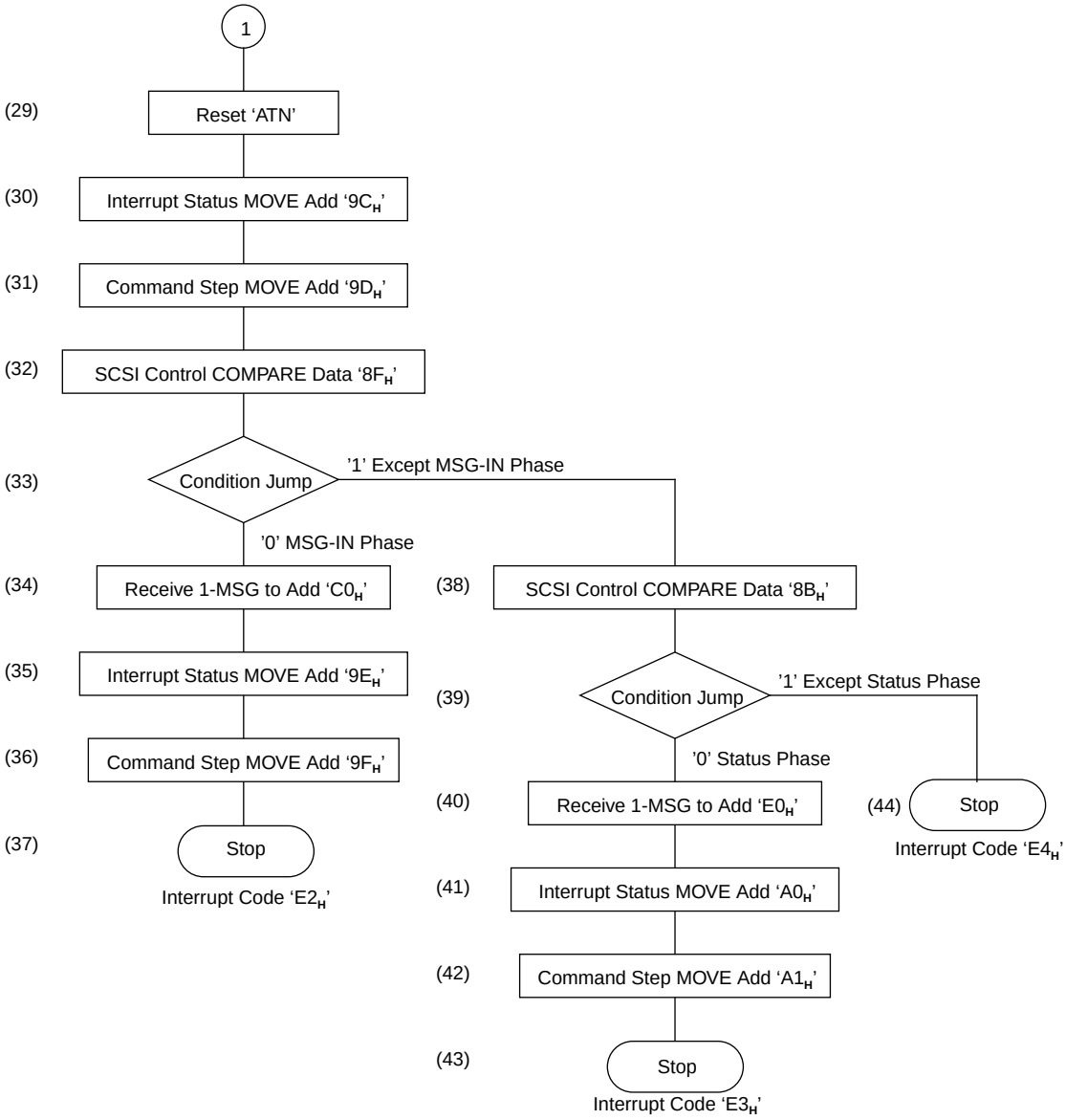
Flowchart	Step	Description
	0	* Receive command
	1	* Move to CMD phase Executing CMD phase
	2	* Complete receiving all CDB bytes Command complete

APPENDIX E FLOWCHART FOR INITIATOR USER PROGRAM EXAMPLES

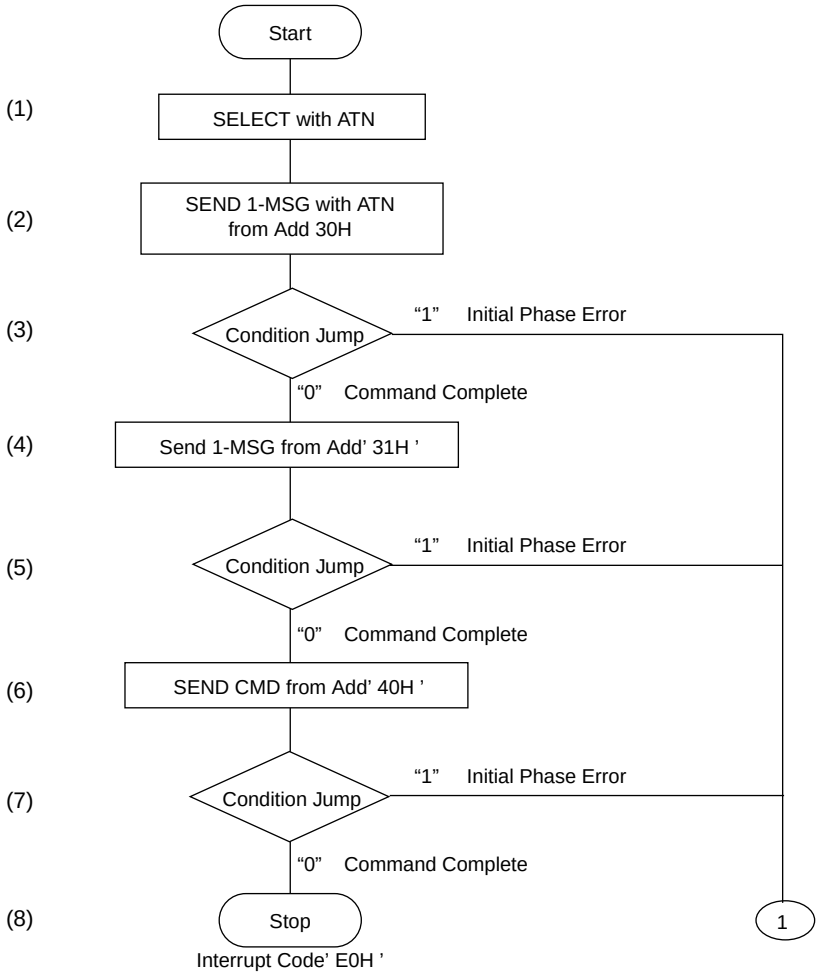
EX 1:

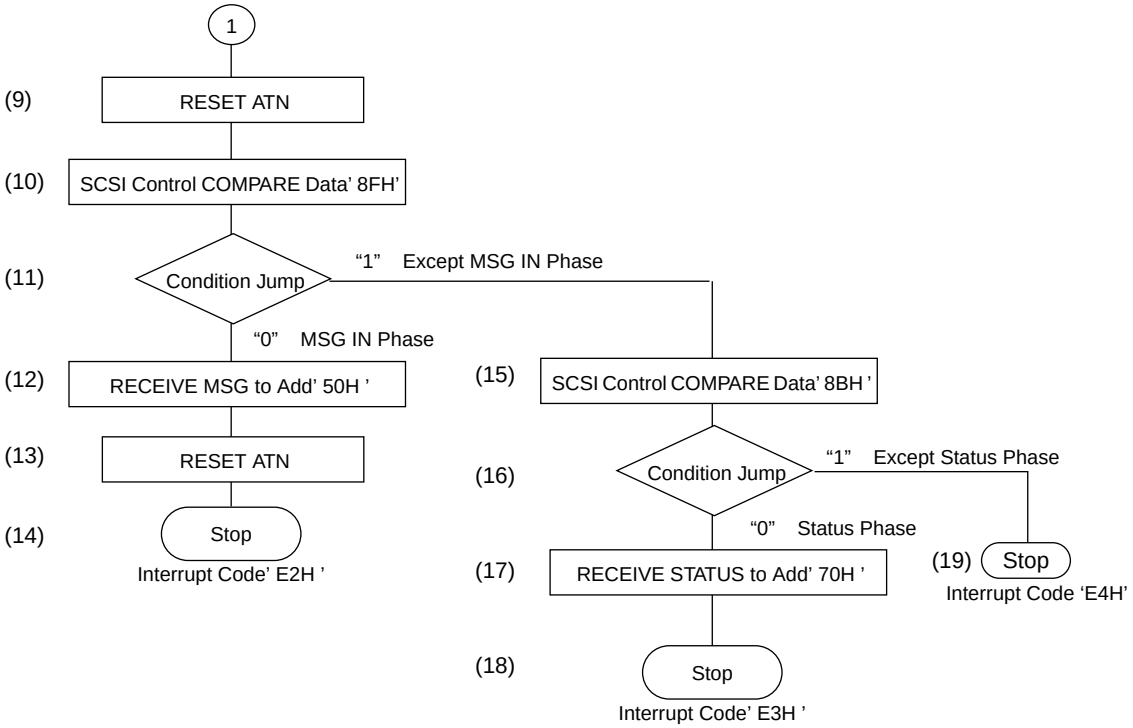



2
(14)
ACC COMPARE Add '40H'
(15)
Condition Jump
'1' Initial Phase Error
'0' Command Complete
(16)
Reset 'ACK'
1
(17)
Interrupt Status MOVE Add '96H'
(18)
Command Step MOVE Add '97H'
(19)
Send 1-CMD from Add 'B8H'
(20)
Interrupt Status MOVE Add '98H'
(21)
Command Step MOVE Add '99H'
(22)
Add '98H' MOVE ACC
(23)
ACC COMPARE Add '40H'
Condition Jump
'1' Initial Phase Error
'0' Command Complete
(25)
Reset 'ACK'
1
(26)
Interrupt Status MOVE Add '9AH'
(27)
Command Step MOVE Add '9BH'
(28)
Stop
Interrupt Code 'E0H'


1
(29)
Reset 'ATN'
(30)
Interrupt Status MOVE Add '9C$_H$'
(31)
Command Step MOVE Add '9D$_H$'
(32)
SCSI Control COMPARE Data '8F$_H$'
(33)
Condition Jump
'1' Except MSG-IN Phase
'0' MSG-IN Phase
(34)
Receive 1-MSG to Add 'C0$_H$'
(35)
Interrupt Status MOVE Add '9E$_H$'
(36)
Command Step MOVE Add '9F$_H$'
(37)
Stop
Interrupt Code 'E2$_H$'
(38)
SCSI Control COMPARE Data '8B$_H$'
(39)
Condition Jump
'1' Except Status Phase
'0' Status Phase
(40)
Receive 1-MSG to Add 'E0$_H$'
(41)
Interrupt Status MOVE Add 'A0$_H$'
(42)
Command Step MOVE Add 'A1$_H$'
(43)
Stop
Interrupt Code 'E3$_H$'
(44)
Stop
Interrupt Code 'E4$_H$'

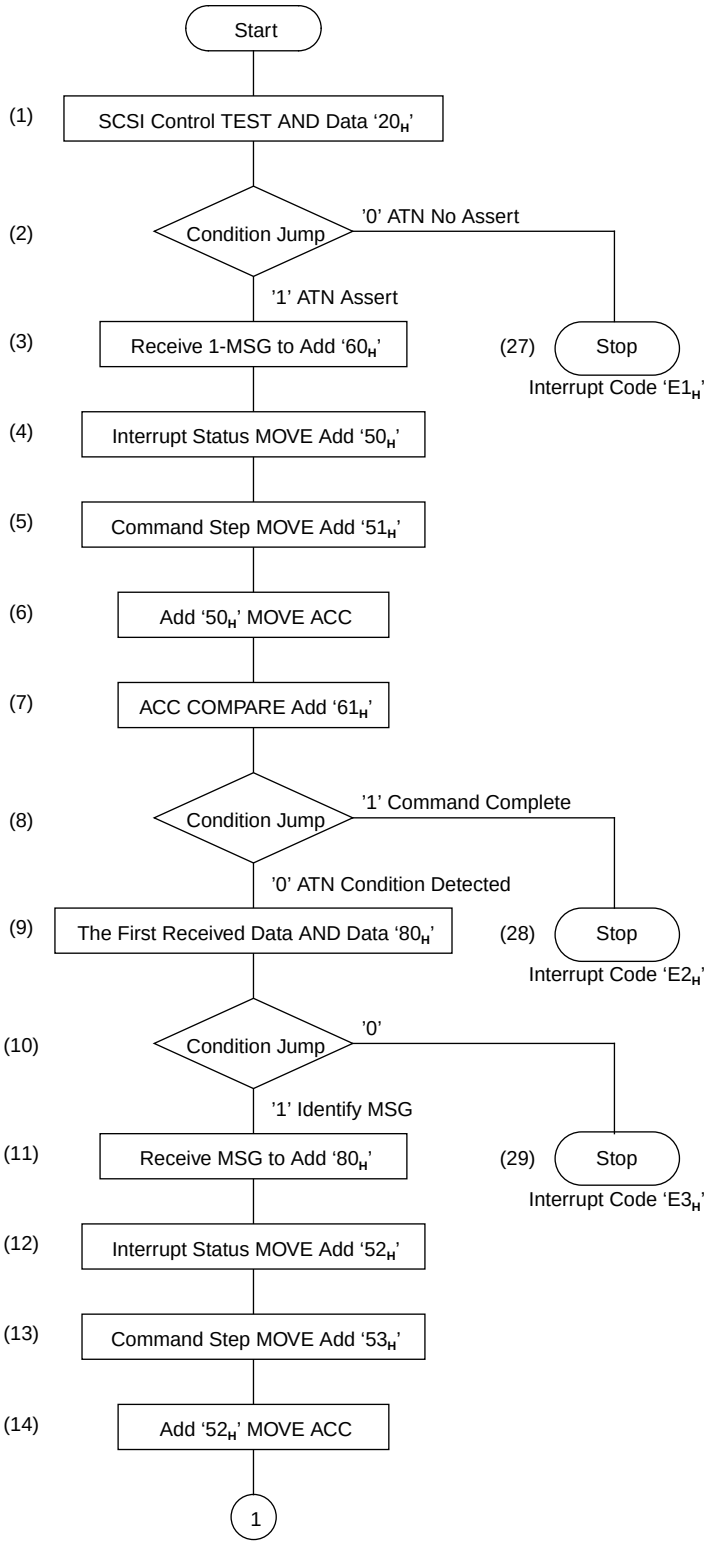
EX 2:

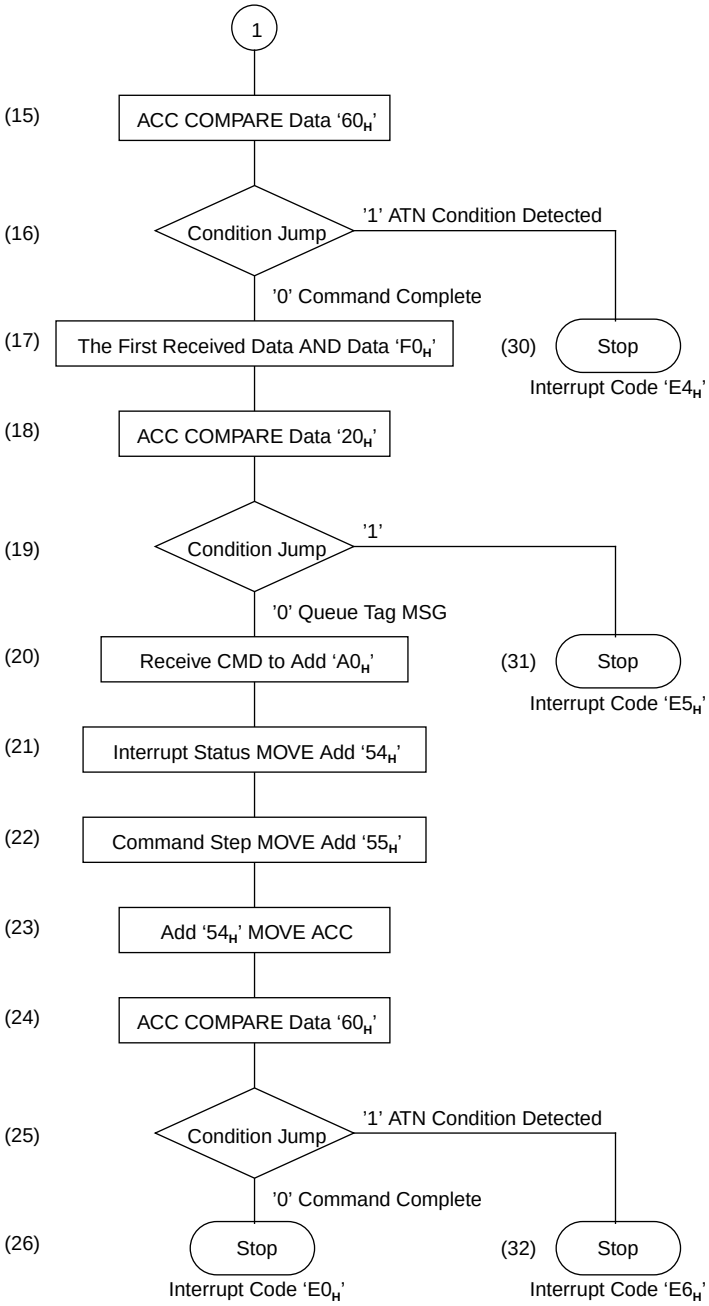

Start
(1)
SELECT with ATN
(2)
SEND 1-MSG with ATN
from Add 30H
(3)
Condition Jump
"1" Initial Phase Error
"0" Command Complete
(4)
Send 1-MSG from Add' 31H '
(5)
Condition Jump
"1" Initial Phase Error
"0" Command Complete
(6)
SEND CMD from Add' 40H '
(7)
Condition Jump
"1" Initial Phase Error
"0" Command Complete
(8)
Stop
Interrupt Code' E0H '
1


1
(9)
RESET ATN
(10)
SCSI Control COMPARE Data' 8FH'
(11)
Condition Jump
"1" Except MSG IN Phase
"0" MSG IN Phase
(12)
RECEIVE MSG to Add' 50H '
(13)
RESET ATN
(14)
Stop
Interrupt Code' E2H '
(15)
SCSI Control COMPARE Data' 8BH '
(16)
Condition Jump
"1" Except Status Phase
"0" Status Phase
(17)
RECEIVE STATUS to Add' 70H '
(18)
Stop
Interrupt Code' E3H '
(19)
Stop
Interrupt Code 'E4H'

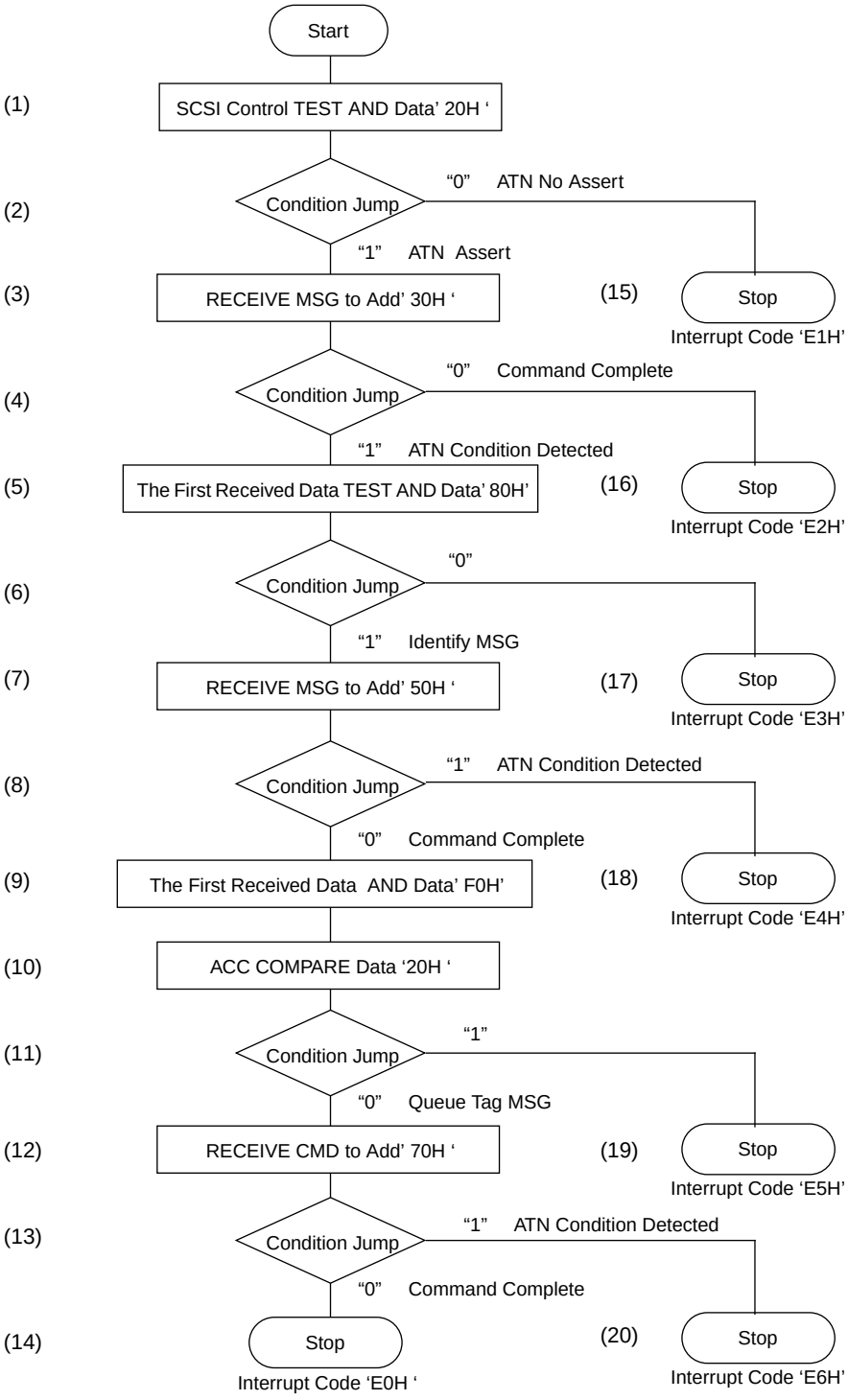
APPENDIX F FLOWCHART FOR TARGET USER PROGRAM EXAMPLES

EX 1:




1
(15)
ACC COMPARE Data '60H'
(16)
Condition Jump
'1' ATN Condition Detected
'0' Command Complete
(17)
The First Received Data AND Data 'F0H'
(30)
Stop
Interrupt Code 'E4H'
(18)
ACC COMPARE Data '20H'
(19)
Condition Jump
'1'
'0' Queue Tag MSG
(20)
Receive CMD to Add 'A0H'
(31)
Stop
Interrupt Code 'E5H'
(21)
Interrupt Status MOVE Add '54H'
(22)
Command Step MOVE Add '55H'
(23)
Add '54H' MOVE ACC
(24)
ACC COMPARE Data '60H'
(25)
Condition Jump
'1' ATN Condition Detected
'0' Command Complete
(26)
Stop
Interrupt Code 'E0H'
(32)
Stop
Interrupt Code 'E6H'

EX 2:


Start
(1)
SCSI Control TEST AND Data' 20H '
(2)
Condition Jump
"0" ATN No Assert
"1" ATN Assert
(3)
RECEIVE MSG to Add' 30H '
(15)
Stop
Interrupt Code 'E1H'
(4)
Condition Jump
"0" Command Complete
"1" ATN Condition Detected
(5)
The First Received Data TEST AND Data' 80H'
(16)
Stop
Interrupt Code 'E2H'
(6)
Condition Jump
"0"
"1" Identify MSG
(7)
RECEIVE MSG to Add' 50H '
(17)
Stop
Interrupt Code 'E3H'
(8)
Condition Jump
"1" ATN Condition Detected
"0" Command Complete
(9)
The First Received Data AND Data' F0H'
(18)
Stop
Interrupt Code 'E4H'
(10)
ACC COMPARE Data '20H '
(11)
Condition Jump
"1"
"0" Queue Tag MSG
(12)
RECEIVE CMD to Add' 70H '
(19)
Stop
Interrupt Code 'E5H'
(13)
Condition Jump
"1" ATN Condition Detected
"0" Command Complete
(14)
Stop
Interrupt Code 'E0H '
(20)
Stop
Interrupt Code 'E6H'

APPENDIX G LIST OF INTERRUPT CODES AND COMMANDS

This section provides a list of the interrupt codes explained in 5.8 and the command steps displayed when these interrupts are produced, their content, and SPC operation during the interrupt.

The * and Note symbols used in this list have the following meaning.

***: Indicates an SPC operating state**

Note: Indicates points to watch in using the SPC.

G.1 Initiator Commands

■ Initiator Sequential Commands

- All commands

CODE	STEP	SPC Operation
31_H	00_H	A bus free state is detected keeping ACK signal asserted (at the manual RESET ACK mode), after transfer completion.
64_H	00_H	• A new command is received while a command is being executed (double reception). After the execution results for the command received first are reported, this interrupt is reported. • When a command is received and the SPC has already detected an interrupt cause produced during the Ready state, this interrupt cause report is given priority. After the interrupt cause reported during the SPC Ready state is reported, this interrupt is reported.
65_H	00_H	• A target command is received while the SPC is functioning as the initiator. • An undefined command is received. • Commands which start with a transfer are sent even though a nexus has not been established. Or commands which start with a selection are sent even though a nexus has been established.
66_H	00_H	A parity error is detected in a register set by the host MPU.

- SELECT & CMD

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected while waiting for a bus free state.
	01_H	A reset condition is detected during arbitration phase execution.
	02_H	A reset condition is detected during selection phase execution.
	03_H	A reset condition is detected during command phase execution.*1
	04_H	A reset condition is detected after command phase execution.
$2C_H$	03_H	The specified REQ/ACK timeout time is exceeded during command phase execution.*1
31_H	03_H	A bus free state is detected during command phase execution.*1
32_H	03_H	A target phase change occurs during command phase execution.*1
54_H	03_H	The phase when the REQ signal is first received is not the command phase.
55_H	03_H	The phase when the REQ signal is first received is not the command phase and a single message is received in response to the target message reception request.*2
56_H	03_H	The phase when the REQ signal is first received is not the command phase and a single status is received in response to the target status reception request.*2
60_H	04_H	Operation is completed normally.*3
	$0C_H$	Operation is completed normally and negates the final ACK signal.*4
64_H	00_H	• The SPC is reselected after failing at arbitration and waiting for a bus free state. After this interrupt is reported, the results of the operation established after reselection (Automatic reselection response mode or user program operation) are reported. • The SPC has already been reselected when a command is received and the operation established after reselection is being executed. After the results of the operation being executed are reported. this interrupt is reported.
65_H	03_H	• The group 6/7 CDB length setting is not defined. • The data sent is group 3/4 CDB. A report is made without performing a CDB transfer.
67_H	01_H	A Command Pause is received during the arbitration phase.
70_H	02_H	A BSY signal response is received from the target and a nexus is established. After that, however, a bus free state is detected.
	04_H	A bus free state is detected after the command phase is executed.
82_H	02_H	The selection/reselection timeout time is exceeded. After the specified number of retries is attempted, this interrupt is reported.

*1 Completed at the same time as this state is detected.
*2 Reported when the Automatic receive mode is established.
*3 Applicable when the Manual RESET ACK mode is set.
*4 Applicable when the Automatic RESET ACK mode is set.

• SELECT & 1-MSG & CMD, SELECT & N-Byte-MSG & CMD

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected while waiting for a bus free state.
	01_H	A reset condition is detected during arbitration phase execution.
	02_H	A reset condition is detected during selection phase execution.
	03_H	A reset condition is detected during message phase execution.*1
	04_H	A reset condition is detected during command phase execution.*1
	05_H	A reset condition is detedted after command phase execution.
$2C_H$	03_H	The specified REQ/ACK timeout time is exceeded during message phase execution.*1
	04_H	The specified REQ/ACK timeout time is exceeded during command phase execution.*1
31_H	03_H	A bus free state is detected during message phase execution.*1
	04_H	A bus free state is detected during command phase execution.*1
32_H	03_H	A target phase change occurs during message phase execution.*1
	04_H	A target phase change occurs during command phase execution.*1
54_H	03_H	The phase when the REQ signal is first received is not the message out phase. This interrupt is reported without negating the ATN signal.
	04_H	The phase when the REQ signal is first received is not the command phase.
55_H	03_H	The phase when the REQ signal is first received is not the message out phase and a single message is received in response to the target message reception request. (the ATN signal is negated).*2
	04_H	The phase when the REQ signal is first received is not the command phase and a single message is received in response to the target message reception request.*2
56_H	03_H	The phase when the REQ signal is first received is not the message out phase and a single status is received in response to the target status reception request (the ATN signal is negated).*2
	04_H	The phase when the REQ signal is first received is not the command phase and a single status is received in response to the target status reception request.*2
60_H	05_H	Operation is completed normally.*3
	$0D_H$	Operation is completed normally and negates the final ACK signal.*4
64_H	00_H	• The SPC is reselected after failing at arbitration and waiting for the next bus free state. After this interrupt is reported, the results of the operation set after reselection (Automatic reselection response mode or user program operation) are reported. • The SPC has already been reselected when a command is received and the operation set after reselection is being executed. After the results of the operation being executed are reported, this interrupt is reported.
	03_H	The message length is 33 bytes or longer (SELECT & CMD only)
65_H	03_H	The MC byte register is undefiend (for SELECT & N-Byte-MSG & CMD only)
	04_H	• The group 6/7 CDB length setting is not defined. • The data sent is group 3/4 CDB. A report is made without performing a CDB transfer.
67_H	01_H	A Command Pause is received during the arbitration phase.
70_H	02_H	A BSY signal response is received from the target and a nexus is established. After that, however, a bus free state is detected.
	05_H	A bus free state is detected after the command phase is executed.
82_H	02_H	The selection/reselection timeout time is exceeded. After the specified number of retries is attempted, this interrupt is reported.

*1 Completed at the same time as this state is detected.
*2 Reported when the Automatic receive mode is established.
*3 Applicable when the Manual RESET ACK mode is set.
*4 Applicable when the Automatic RESET ACK mode is set.

- SELECT & 1-MSG, SELECT & N-Byte-MSG

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected while waiting for a bus free state.
	01_H	A reset condition is detected during arbitration phase execution.
	02_H	A reset condition is detected during selection phase execution.
	03_H	A reset condition is detected during message phase execution.*1
	04_H	A reset condition is detected after message phase execution.
$2C_H$	03_H	The specified REQ/ACK timeout time is exceeded.*1
31_H	03_H	A bus free state is detected during message phase execution.*1
32_H	03_H	A target phase change occurs during message phase execution.*1
54_H	03_H	The phase when the REQ signal is first received is not the message out phase This interrupt is reported without negating the ATN signal.
55_H	03_H	The phase when the REQ signal is first received is not themessage out phase and a single message is received in response to the target message reception request (the ATN is negated).*2
56_H	03_H	The phase when the REQ signal is first received is not the message out phase and a single status is received in response to the target status reception request (the ATN is negated).*2
60_H	04_H	Operation is completed normally.*3
	$0C_H$	Operation is completed normally and negates the final ACK signal.*4
64_H	00_H	• The SPC is reselected after failing at arbitration and waiting for a bus free state. After this interrupt is reported, the results of the operation established after reselection (Automatic reselection response mode or user program operation) are reported. • The SPC has already been reselected when a command is received and the operation established after reselection is being executed. After the results of the operation being executed are reported, this interrupt is reported.
	03_H	The message length is 33 bytes or longer (SELECT & 1-MSG only)
65_H	03_H	The MC byte register is undefined (00_H) (SELECT & N-Byte-MSG only)
67_H	01_H	A Command Pause is received during the arbitration phase.
70_H	02_H	A BSY signal response is received from the target and a nexus is established, after which a bus free state is detected.
	04_H	A bus free state is detected after the command phase is executed.
82_H	02_H	The selection/reselection timeout time is exceeded. After the specified number of retries is attempted, this interrupt is reported.

*1 Completed at the same time as this state is detected.
*2 Reported when the Automatic receive mode is established.
*3 Applicable when the Manual RESET ACK mode is set.
*4 Applicable when the Automatic RESET ACK mode is set.

- SEND N-Byte-MSG

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected before the transfer is executed.
	01_H	A reset condition is detected while the transfer is being executed.*1
	02_H	A reset condition is detected after the transfer is executed.
$2C_H$	01_H	The specified REQ/ACK timeout time is exceeded.*1
31_H	01_H	A bus free state is detected while the transfer is being executed.*1
32_H	01_H	A target phase change occurs while the transfer is being executed.*1
54_H	01_H	The phase when the REQ signal is first received is not the message out phase. This interrupt is reported without negating the ATN signal.
55_H	01_H	The phase when the REQ signal is first received is not the message out phase and a single message is received in response to the target message reception request (the ATN signal is negated).*2
56_H	01_H	The phase when the REQ signal is first received is not the message out phase and a single status is received in response to the target status reception request (the ATN signal is negated).*2
60_H	02_H	Operation is completed normally.*3
	$0A_H$	Operation is completed normally and negates the final ACK signal.*4
65_H	01_H	The MC byte register is undefined (00_H).
70_H	00_H	A bus free state is detected before the transfer is executed.
	02_H	A bus free state is detected after the transfer is executed.

*1 Completed at the same time as this state is detected.
*2 Reported when the Automatic receive mode is established.
*3 Applicable when the Manual RESET ACK mode is set.
*4 Applicable when the Automatic RESET ACK mode is set.

- SEND N-Byte-CMD

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected before the transfer is executed.
	01_H	A reset condition is detected while the transfer is being executed.*1
	02_H	A reset condition is detected after the transfer is executed.
$2C_H$	01_H	The specified REQ/ACK timeout time is exceeded.*1
31_H	01_H	A bus free state is detected while the transfer is being executed.*1
32_H	01_H	A target phase change occurs while the transfer is being executed.*1
54_H	01_H	The phase when the REQ signal is first received is not the command phase.
55_H	01_H	The phase when the REQ signal is first received is not the command phase and a single message is received in response to the target message reception request.*2
56_H	01_H	The phase when the REQ signal is first received is not the command phase and a single status is received in response to the target status reception request.*2
60_H	02_H	Operation is completed normally.*3
	$0A_H$	Operation is completed normally and negates the final ACK signal.*4
65_H	01_H	The MC byte register is undefined (00_H).
70_H	00_H	A bus free state is detected before the transfer is executed.
	02_H	A bus free state is detected after the transfer is executed.

*1 Completed at the same time as this state is detected.
*2 Reported when the Automatic receive mode is established.
*3 Applicable when the Manual RESET ACK mode is set.
*4 Applicable when the Automatic RESET ACK mode is set.

- RECEIVE N-Byte-MSG

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected before the transfer is executed.
	01_H	A reset condition is detected while the transfer is being executed.*
	02_H	A reset condition is detected after the transfer is executed.
24_H	01_H	A parity error is detected in input data at the SCSI interface.
$2C_H$	01_H	The specified REQ/ACK timeout time is exceeded.*1
31_H	01_H	A bus free state is detected while the transfer is being executed.*1
32_H	01_H	A target phase change occurs while the transfer is being executed.*1
54_H	01_H	The phase when the REQ signal is first received is not the message in phase.
56_H	01_H	The phase when the REQ signal is first received is not the message in phase and a single status is received in response to the target status reception request.*2 Reported when the Automatic receive mode is established.
60_H	02_H	Operation is completed normally.
65_H	01_H	The MC byte register is undefined (00_H).
70_H	00_H	A bus free state is detected before the transfer is executed.
	02_H	A bus free state is detected after the transfer is executed.

*1 Completed at the same time as this state is detected.
*2 Reported when the Automatic receive mode is established.

■ Initiator Discrete Commands

- All commands

CODE	STEP	SPC Operation
31_H	00_H	A bus free state is detected keeping ACK signal asserted (at Manual RESET ACK mode), after the transfer completion.
64_H	00_H	• A new command is received while a command is being executed (double reception). After the execution results for the command received first are reported, this interrupt is reported. • When a command is received and the SPC has already detected an interrupt cause produced during the Ready state, this interrupt cause report is given priority. After the interrupt cause reported during the SPC Ready state is reported, this interrupt is reported.
65_H	00_H	• A target command is received while the SPC is functioning as the initiator. • An undefined command is received. • Commands which start with a transfer are sent even though a nexus has not been established. Or commands which start with a selection are sent even though a nexus has been established.
66_H	00_H	A parity error is detected in a register set by the host MPU.

- SELECT, SELECT with ATN

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected while waiting for a bus free state.
	01_H	A reset condition is detected during arbitration phase execution.
	02_H	A reset condition is detected during selection phase execution.
60_H	03_H	Operation is completed normally.
64_H	00_H	• The SPC is reselected after failing at arbitration and waiting for the next bus free state. After this interrupt is reported, the results of the operation set after reselection (Automatic reselection response mode or user program operation) are reported. •The SPC has already been reselected when a command is received and the operation set after reselection is being executed. After the results of the operation being executed are reported, this interrupt is reported.
67_H	01_H	A Command Pause is received during the arbitration phase.
70_H	02_H	A BSY signal response is received from the target and a nexus is established. After that, however, a bus free state is detected.
82_H	02_H	The selection/reselection timeout time is exceeded. After the specified number of retries is attempted, this interrupt is reported.

- SET ATN, RESET ATN, SET ACK, RESET ACK

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected before the command is executed.
	01_H	A reset condition is detected after the command is executed.
31_H	00_H	A bus free state is detected before the command is executed (when this command is sent after the transfer command is executed).*1
60_H	01_H	Operation is completed normally.
	09_H	Operation is completed normally and negates the ACK signal.*2
70_H	00_H	A bus free state is detected before the command is executed (when this command is sent after a command which does not include a transfer is executed).
	01_H	A bus free state is detected after the command is executed.

*1 Reported when the Manual RESET ACK mode is set.

*2 Reported when the Automatic RESET ACK mode is set. Also, applicable to SET ATN command only.

- SEND DATA from MPU/DMA (WITH PADDING)

CODE	STEP	Interrupt Source and SPC Operation
01_H	00_H	Reset condition detected before transfer
	01_H	Reset condition detected during transfer *2
	02_H	• Reset condition detected after transfer. • Reset condition detected during padding transfer.
21_H	01_H	Parity error detected in input/output data over DMA interface during DMA transfer *1
	09_H	Parity error detected in input/output data over DMA interface during DMA transfer, and the last ACK signal negated after the transfer completed *1, *6
22_H	01_H	Parity error detected when accessing SCSI data register during program transfer *1
	09_H	Parity error detected in input/output data over MPU interface during program transfer, and the last ACK signal negated after the transfer completed *1, *6
24_H	01_H	Parity error detected in input/output data over SCSI interface *1
	09_H	Parity error detected in input/output data over SCSI interface, and the last ACK signal negated after the transfer completed *1, *6
25_H	01_H	Parity error detected in input/output data over DMA and SCSI interfaces during DMA transfer *1
	09_H	Parity error detected in input/output data over DMA and SCSI interfaces during DMA transfer, and the last ACK signal negated after the transfer completed *1, *6
26_H	01_H	Parity error detected in input/output data over MPU and SCSI interfaces during program transfer *1
	09_H	Parity error detected in input/output data over MPU and SCSI interfaces during program transfer, and the last ACK signal negated after the transfer completed *1,*6
29_H	01_H	REQ signal greater than specified offset value received in synchronous transfer *1
	09_H	REQ signal greater than specified offset value received in synchronous transfer, and the last ACK signal negated after the transfer completed *1, *6
$2A_H$	01_H	SPC could not follow because period of received REQ signal short *1 This interrupt may be reported even in asynchronous transfer.
	09_H	SPC could not follow because period of received REQ signal short, and the last ACK signal negated after the transfer completed *1, *6 This interrupt may be reported even in asynchronous transfer.
$2B_H$	01_H	REQ signal greater than specified offset value received in synchronous transfer and SPC could not follow because period of received REQ signal short *1
	09_H	REQ signal greater than specified offset value received in synchronous transfer and SPC could not follow because period of received REQ signal short, and the last ACK signal negated after the transfer completed *1, *6
$2C_H$	01_H	More than specifed REQ/ACK timeout time elapsed (only in asynchronous mode) *2
31_H	01_H	Bus free state detected during data transfer *2
32_H	01_H	Target changed phase during data transfer *2
54_H	01_H	Phase when REQ signal first received not data-out phase
55_H	01_H	Phase when REQ signal first received not data-out phase and one message received in response to message receive request from target*3
56_H	01_H	Phase when REQ signal first received not data-out phase and one status received in response to status receive request from target*3
60_H	02_H	Terminated normally*5
	$0A_H$	Terminated normally and negated the last ACK signal*6
65_H	01_H	Value of data block register or data byte register specified by host MPU was 0 Padding transfer is performed from the beginning without reporting this interrupt.
67_H	01_H	**Command Pause** accepted during data transfer*2
70_H	01_H	Bus free state detected before transfer*3
	02_H	• Bus free state detected after transfer • Bus free state detected during padding transfer*4

*1 The ATN signal is asserted to continue tranfer.
*2 This interrupt is reported after the exit processing (for exit processing, see section 5.9.6).
*3 Reported when the Automatic receive mode is set.
*4 Reported at the same time as this state is detected.
*5 Reported only when the Manual RESET ACK mode is set.
*6 Reported only when the Automatic RESET ACK mode is set (except for padding transfer being executed).

- RECEIVE DATA to MPU/DMA (WITH PADDING)

CODE	STEP	Interrupt Source and SPC Operation
01_H	00_H	Reset condition is detected before transfer
	01_H	Reset condition is detected during transfer*2
	02_H	• Reset condition detected after transfer. • Reset condition detected during padding transfer.*4
21_H	01_H	Parity error detected in input/output data over DMA interface during DMA transfer*1
	09_H	Parity error detected in input/output data over DMA interface during DMA transfer, and the last ACK signal negated after the transfer completed*1, *6
22_H	01_H	Parity error detected in input/output data over MPU interface*1
	09_H	Parity error detected in input/output data over MPU interface during program transfer, and the last ACK signal negated after the transfer completed*1, *6
24_H	01_H	Parity error detected in input/output data over SCSI interface*1
	09_H	Parity error detected in input/output data over SCSI interface, and the last ACK signal negated after the transfer completed*1, *6
25_H	01_H	Parity error detected in input/output data over DMA and SCSI interfaces during DMA transfer*1
	09_H	Parity error detected in input/output data over DMA and SCSI interfaces during DMA transfer, and the last ACK signal negated after the transfer completed*1, *6
26_H	01_H	Parity error detected in input/output data over MPU and SCSI interfaces during program transfer*1
	09_H	Parity error detected in input/output data over MPU and SCSI interfaces during program transfer, and the last ACK signal negated after the transfer completed*1, *6
29_H	01_H	REQ signal greater than specified offset value received in synchronous transfer*1
	09_H	REQ signal greater than specified offset value received in synchronous transfer, and the last ACK signal negated after the transfer completed*1, *6
$2A_H$	01_H	SPC could not follow because period of received REQ signal short*1 This interrupt may be reported even in asynchronous transfer.
	09_H	SPC could not follow because period of received REQ signal short, and the last ACK signal negated after the transfer completed*1, *6 This interrupt may be reported even in asynchronous transfer.
$2B_H$	01_H	REQ signal greater than specified offset value received in synchronous transfer and SPC could not follow because period of received REQ signal short*1
	09_H	REQ signa greater than specified offset value received in synchronous transfer and SPC ould not follow because period of received REQ signal short, and the last ACK signal negated after the transfer completed*1, *6
$2C_H$	01_H	More than specifed REQ/ACK timeout time elapsed (only in asynchronous mode)*4
31_H	01_H	Bus free state detected during data transfer*2
32_H	01_H	Target changed phase during data transfer*2
54_H	01_H	Phase when REQ signal first received not data-out phase
55_H	01_H	Phase when REQ signal first received not data-out phase and one message received in response to message receive request from target*3
56_H	01_H	Phase when REQ signal first received not data-out phase and one status received in response to status receive request from target*3
60_H	02_H	Terminated normally*5
	$0A_H$	Terminated normally and negated the last ACK signal*6
65_H	01_H	Value of data block register or data byte register set by host MPU was 0 Padding transfer is performed from the beginning without reporting this interrupt.
67_H	01_H	**Command Pause** accepted during data phase*2
70_H	01_H	Bus free state detected before transfer*3
	02_H	• Bus free state detected after transfer • Bus free state detected during padding transfer*4

*1 The ATN signal is asserted to continue tranfer.
*2 This interrupt is reported after the exit processing (for exit processing, see section 5.9.6).
*3 Reported when the Automatic receive mode is set.
*4 Reported at the same time as this state is detected.
*5 Reported only when the Manual RESET ACK mode is set.
*6 Reported only when the Automatic RESET ACK mode is set (except for padding transfer being executed).

- SEND 1-MSG, SEND 1-MSG with ATN

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected before the transfer is executed.
	01_H	A reset condition is detected while the transfer is being executed.*1
	02_H	A reset condition is detected after the transfer is executed.
$2C_H$	01_H	The specified REQ/ACK timeout time is exceeded.*1
31_H	01_H	A bus free state is detected while the transfer is being executed.*1
32_H	01_H	A target phase change occurs while the transfer is being executed.*1
54_H	01_H	The phase when the REQ signal is first received is not the message out phase. This interrupt is reported without negating the ATN signal.
55_H	01_H	The phase when the REQ signal is first received is not the message out phase and a single message is received in response to the target message reception request (the ATN signal is negated).*2
56_H	01_H	The phase when the REQ signal is first received is not the message out phase and a single status is received in response to the target status reception request (the ATN signal is negated).*2
60_H	02_H	Operation is completed normally.*3
	$0A_H$	Operation is completed normally and negates the final ACK signal*4
64_H	01_H	The MSG length is 33 bytes or longer. MSGs are not sent.
70_H	00_H	A bus free state is detected before the transfer is executed.
	02_H	A bus free state is detected after the transfer is executed.

*1 Completed at the same time as this state is detected.
*2 Reported when the Automatic receive mode is set.
*3 Applicable when the Manual RESET ACK mode is set.
*4 Applicable when the Automatic RESET ACK mode is set.

- RECEIVE MSG

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected before the transfer is executed.
	01_H	A reset condition is detected while the transfer is being executed.*1
	02_H	A reset condition is detected after the transfer is executed.
24_H	01_H	A parity error is detected in the received MSG bytes. • When the message being received is a 1 byte message or a 2 byte message, this is reported after all bytes are received. • When the message being received is an extended message, if the parity error is detected in the first or second byte, this is reported after 2 bytes are received but if the parity error is detected in the third byte or after, this is reported after all bytes are received.
$2C_H$	01_H	The specified REQ/ACK timeout time is exceeded.*1
31_H	01_H	A bus free state is detected while the transfer is being executed.*1
32_H	01_H	A target phase change occurs while the transfer is being executed.*1
54_H	01_H	The phase when the REQ signal is first received is not the message in phase.
56_H	01_H	The phase when the REQ signal is first received is not the message in phase and a single status is received in response to the target status reception request.*2 Reported when the Automatic receive mode is set.
60_H	02_H	Operation is completed normally.
67_H	01_H	The message received is an extended message and has a message length of 33 bytes or longer. The operation completes receiving up to second byte of message.
70_H	00_H	A bus free state is detected before the transfer is executed.
	02_H	A bus free state is detected after the transfer is executed.

*1 Completed at the same time as this state is detected.
*2 Reported when the Automatic receive mode is set.

- SEND CMD

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected before the transfer is executed.
	01_H	A reset condition is detected while the transfer is being executed.*1
	02_H	A reset condition is detected after the transfer is executed.
$2C_H$	01_H	The specified REQ/ACK timeout time is exceeded.*1
31_H	01_H	A bus free state is detected while the transfer is being executed.*1
32_H	01_H	A target phase change occurs while the transfer is being executed.*1
54_H	01_H	The phase when the REQ signal is first received is not the command phase.
55_H	01_H	The phase when the REQ signal is first received is not the command phase and a single message is received in response to the target message reception request.*2
56_H	01_H	The phase when the REQ signal is first received is not the command phase and a single status is received in response to the target status reception request.*2
60_H	02_H	Operation is completed normally.*3
	$0A_H$	Operation is completed normally and negates the final ACK signal*4
65_H	01_H	The group 6/7 CDB length setting is not defined. • The data sent is group 3/4 CDB. A report is made without performing a command phase transfer.
70_H	00_H	A bus free state is detected before the transfer is executed.
	02_H	A bus free state is detected after the transfer is executed.

*1 Completed at the same time as this state is detected.
*2 Reported when the Automatic receive mode is set.
*3 Applicable when the Manual RESET ACK mode is set.
*4 Applicable when the Automatic RESET ACK mode is set.

- RECEIVE STATUS

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected before the transfer is executed.
	01_H	A reset condition is detected while the transfer is being executed.*1
	02_H	A reset condition is detected after the transfer is executed.
24_H	01_H	A parity error is detected in input data at the SCSI interface (end after 1 byte is transferred).*3
	09_H	A parity error is detected in input data at the SCSI interface and the ACK signal negated after transfer completion.*4
$2C_H$	01_H	The specified REQ/ACK timeout time is exceeded.*1
54_H	01_H	The phase when the REQ signal is first received is not the status phase.
55_H	01_H	The phase when the REQ signal is first received is not the status phase and a single message is received in response to the target message reception request. Reported when the Automatic receive mode is set.
60_H	02_H	Operation is completed normally.*3
	$0A_H$	Operation is completed normally and negates the final ACK signal*4
70_H	00_H	A bus free state is detected before the transfer is executed.
	02_H	A bus free state is detected after the transfer is executed.

*1 Completed at the same time as this state is detected.
*2 Reported when the Automatic receive mode is set.
*3 Applicable when the Manual RESET ACK mode is set.
*4 Applicable when the Automatic RESET ACK mode is set.

G.2 Target Commands

■ Target Sequential Commands

- All commands

CODE	STEP	SPC Operation
64_H	00_H	• A new command is received while a command is being executed (double reception). After the execution results for the command received first are reported, this interrupt is reported. • When a command is received, SPC has already detected an interrupt source which is generated only in SPC ready state, and the priority of interrupt report is higher than for the received command. This interrupt is reported after reporting the interrupt which has been generated.
65_H	00_H	• An initiator command is received while the SPC is functioning as the target. • An undefined command is received. • Commands which start with a transfer are sent even though a nexus has not been established. Or commands which start with a reselection are sent even though a nexus has been established.
66_H	00_H	A parity error is detected in a register set by the host MPU.

- RESELECT & 1-MSG, RESELECT & N-Byte-MSG

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected while waiting for a bus free state.
	01_H	A reset condition is detected during arbitration.
	02_H	A reset condition is detected during reselection.
	03_H	A reset condition is detected during message phase execution.*
	04_H	A reset condition is detected after message phase execution.
$2C_H$	03_H	The specified REQ/ACK timeout time is exceeded.*
60_H	04_H	Operation is completed normally.
61_H	04_H	The command is completed normally and the attention condition generated by the initiator is detected. Reported when the Automatic receive mode is not set.
62_H	04_H	The command is completed normally, the attention condition generated by the initiator is detected, and a single message is received. Reported when the Automatic receive mode is set.
64_H	00_H	• The SPC is selected after failing at arbitration and waiting for the next bus free state. After this interrupt is reported, the results of the operation set after selection (Automatic selection response mode or user program operation) are reported. • The SPC has already been selected when a command is received and the operation set after reselection is being executed. After the results of the operation being executed are reported, this interrupt is reported.
	03_H	The message length is 33 bytes or longer (RESELECT & 1-MSG only). MSG not transferred.
65_H	03_H	The MC byte register is undefined (00_H) (RESELECT & N-Byte-MSG only).
67_H	01_H	A Command Pause is received during the arbitration phase.
82_H	02_H	The selection/reselection timeout time is exceeded. After the specified number of retries is attempted, this interrupt is reported.

* Completed at the same time as this state is detected.

- RESELECT & 1-MSG & TERMINATE, RESELECT & N-Byte-MSG & TERMINATE

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected while waiting for a bus free state.
	01_H	A reset condition is detected during arbitration.
	02_H	A reset condition is detected during reselection.
	03_H	A reset condition is detected during message in phase execution.*1
	04_H	A reset condition is detected during status phase execution.*1
	05_H	A reset condition is detected during message in phase execution.
	06_H	A reset condition is detected after message in phase execution.
$2C_H$	03_H	The specified REQ/ACK timeout time is exceeded during message in phase execution.*1
	04_H	The specified REQ/ACK timeout time is exceeded during status phase execution.*1
	05_H	The specified REQ/ACK timeout time is exceeded during message in phase execution.*1
42_H	03_H	After the message in phase is completed normally, the attention condition generated by the initiator is detected.*2
	04_H	After the status phase is completed normally, the attention condition generated by the initiator is detected.*2
	05_H	After the message in phase is completed normally, the attention condition generated by the initiator is detected.*2
43_H	03_H	After the message in phase is completed normally, the attention condition generated by the initiator is detected and a single message is received.*3
	04_H	After the status phase is completed normally, the attention condition generated by the initiator is detected and a single message is received.*3
	05_H	After the message in phase is completed normally, the attention condition generated by the initiator is detected and a single message is received.*3
60_H	06_H	Operation is completed normally.
64_H	00_H	• The SPC is selected after failing at arbitration and waiting for the next bus free state. After this interrupt is reported, the results of the operation set after selection (Automatic selection response mode or user program operation) are reported. • The SPC has already been selected when a command is received and the operation set after reselection is being executed. After the results of the operation being executed are reported, this interrupt is reported.
	03_H	The message length is 33 bytes or longer (RESELECT & 1-MSG & TERMINATE only).
65_H	03_H	The MC byte register is undefined (00_H) (RESELECT & N-Byte-MSG & TERMINATE only).
67_H	01_H	A Command Pause is received during the arbitration phase.
82_H	02_H	The selection/reselection timeout time is exceeded. After the specified number of retries is attempted, this interrupt is reported.

*1 Completed at the same time as this state is detected.
*2 Reported when the Automatic receive mode is not set.
*3 Reported when the Automatic receive mode is set.

- RESELECT & 1-MSG & LINK-TERMINATE, RESELECT & N-Byte-MSG & LINK TERMINATE

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected while waiting for a bus free state.
	01_H	A reset condition is detected during arbitration.
	02_H	A reset condition is detected during reselection.
	03_H	A reset condition is detected during message in phase execution.*1
	04_H	A reset condition is detected during status phase execution.*1
	05_H	A reset condition is detected during message in phase execution.
	06_H	A reset condition is detected after message in phase execution.
$2C_H$	03_H	The specified REQ/ACK timeout time is exceeded during message in phase execution.*1
	04_H	The specified REQ/ACK timeout time is exceeded during status phase execution.*1
	05_H	The specified REQ/ACK timeout time is exceeded during message in phase execution.*1
42_H	03_H	After the message in phase is completed normally, the attention condition generated by the initiator is detected.*2
	04_H	After the status phase is completed normally, the attention condition generated by the initiator is detected.*2
43_H	03_H	After the message in phase is completed normally, the attention condition generated by the initiator is detected and a single message is received.*3
	04_H	After the status phase is completed normally, the attention condition generated by the initiator is detected and a single message is received.*3
60_H	06_H	Operation is completed normally.
61_H	06_H	After the command is completed normally (after the message in phase is completed normally), the attention condition generated by the initiator is detected.*2
62_H	06_H	After the command is completed normally (after the message in phase is completed normally) the attention condition generated by the initiator is detected and a single message is received.*3
64_H	00_H	• The SPC is selected after failing at arbitration and waiting for the next bus free state. After this interrupt is reported, the results of the operation set after selection (Automatic selection response mode or user program operation) are reported. • The SPC has already been selected when a command is received and the operation set after reselection is being executed. After the results of the operation being executed are reported, this interrupt is reported.
	03_H	The message length is 33 bytes or longer (RESELECT & 1-MSG & TERMINATE only).
65_H	03_H	The MC byte register is undefined (00_H) (RESELECT & N-Byte-MSG & TERMINATE only).
67_H	01_H	A Command Pause is received during the arbitration phase.
82_H	02_H	The selection/reselection timeout time is exceeded. After the specified number of retries is attempted, this interrupt is reported.

*1 Completed at the same time as this state is detected.
*2 Reported when the Automatic receive mode is not set.
*3 Reported when the Automatic receive mode is set.

- TERMINATE

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected before the transfer.
	01_H	A reset condition is detected during status phase execution.*1
	02_H	A reset condition is detected during message in phase execution.*1
	03_H	A reset condition is detected after message in phase execution.
$2C_H$	01_H	The specified REQ/ACK timeout time is exceeded during status phase execution.*1
	02_H	The specified REQ/ACK timeout time is exceeded during message in phase execution.*1
42_H	01_H	After the status phase is completed normally, the attention condition generated by the initiator is detected.*2
	02_H	After the message in phase is completed normally, the attention condition generated by the initiator is detected.*2
43_H	01_H	After the status phase is completed normally, the attention condition generated by the initiator is detected and a single message is received.*3
	02_H	After the message in phase is completed normally, the attention condition generated by the initiator is detected and a single message is received.*3
60_H	03_H	Operation is completed normally.

*1 Completed at the same time as this state is detected.
*2 Reported when the Automatic receive mode is not set.
*3 Reported when the Automatic receive mode is set.

- LINK-TERMINATE

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected before the transfer.
	01_H	A reset condition is detected during status phase execution.*1
	02_H	A reset condition is detected during message in phase execution.*1
	03_H	A reset condition is detected after message in phase execution.
$2C_H$	01_H	The specified REQ/ACK timeout time is exceeded during status phase execution.*1
	02_H	The specified REQ/ACK timeout time is exceeded during message in phase execution.*1
42_H	01_H	After the status phase is completed normally, the attention condition generated by the initiator is detected.*2
43_H	01_H	After the status phase is completed normally, the attention condition generated by the initiator is detected and a single message is received.*3
60_H	03_H	Operation is completed normally.
61_H	03_H	After the command is completed normally (after the message in phase is completed normally), the attention condition generated by the initiator is detected.*2
62_H	03_H	After the command is completed normally (after the message in phase is completed normally), the attention condition generated by the initiator is detected and a single message is received.*3

*1 Completed at the same time as this state is detected.
*2 Reported when the Automatic receive mode is not set.
*3 Reported when the Automatic receive mode is set.

- DISCONNECT SEQUENCE, DISCONNECT SEQUENCE 2

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected before the transfer.
	01_H	A reset condition is detected during the transfer.*1
	02_H	A reset condition is detected after the transfer is completed.
$2C_H$	01_H	The specified REQ/ACK timeout time is exceeded.*1
42_H	01_H	After the message in phase is completed normally, the attention condition generated by the initiator is detected.*2
43_H	01_H	After the message in phase is completed normally, the attention condition generated by the initiator is detected and a single message is received.*3
60_H	02_H	Operation is completed normally.

*1 Completed at the same time as this state is detected.
*2 Reported when the Automatic receive mode is not set.
*3 Reported when the Automatic receive mode is set.

- SEND N-Byte-MSG

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected before the transfer.
	01_H	A reset condition is detected during the transfer.*
	02_H	A reset condition is detected after the transfer is completed.
$2C_H$	01_H	The specified REQ/ACK timeout time is exceeded.*
60_H	02_H	The operation is completed normally.*
61_H	02_H	After the transfer is completed, the attention condition generated by the initiator is detected. Reported when the Automatic receive mode is not set.
62_H	02_H	After the transfer is completed, the attention condition generated by the initiator is detected and a single message is received. Reported when the Automatic receive mode is set.
65_H	01_H	The MC byte register is undefined (00_H).

* Completed at the same time as this state is detected.

- RECEIVE N-Byte-CMD

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected before the transfer.
	01_H	A reset condition is detected during the transfer.*
	02_H	A reset condition is detected after the transfer is completed.
$2C_H$	01_H	The specified REQ/ACK timeout time is exceeded.*
24_H	01_H	A parity error is detected for input data at the SCSI interface.*
60_H	02_H	The operation is completed normally.
61_H	02_H	After the command is completed, the attention condition generated by the initiator is detected. Reported when the Automatic receive mode is not set.
62_H	02_H	After the command is completed, the attention condition generated by the initiator is detected and a single message is received. Reported when the Automatic receive mode is set.
65_H	01_H	The MC byte register is undefined (00_H).

* Completed at the same time as this state is detected.

- RECEIVE N-Byte-MSG

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected before the transfer.
	01_H	A reset condition is detected during the transfer.*
	02_H	A reset condition is detected after the transfer is completed.
$2C_H$	01_H	The specified REQ/ACK timeout time is exceeded.*
24_H	01_H	A parity error is detected in input data at the SCSI interface.*
60_H	02_H	The operation is completed normally.
61_H	02_H	After the command is completed, the attention condition generated by the initiator is detected. Reported regardless of the Automatic receive mode is set or not.
65_H	01_H	The MC byte register is undefined (00_H).

* Completed at the same time as this state is detected.

■ Target Discrete Commands

- All commands

CODE	STEP	SPC Operation
64_H	00_H	• A new command is received while a command is being executed (double reception). After the execution results for the command received first are reported, this interrupt is reported. • When a command is received and the SPC has already detected an interrupt cause produced during the Ready state, this interrupt cause report is given priority. After the interrupt cause reported during the SPC Ready state is reported, this interrupt is reported.
65_H	00_H	• An initiator command is received while the SPC is functioning as the target. • An undefined command is received. • Commands which start with a transfer are sent even though a nexus has not been established. Or commands which start with a reselection are sent even though a nexus has been established.
66_H	00_H	• A parity error is detected in a register set by the host MPU.

- RESELECT

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected while waiting for a bus free state.
	01_H	A reset condition is detected during arbitration.
	02_H	A reset condition is detected during reselection.
60_H	03_H	Operation is completed normally.
64_H	00_H	• The SPC is selected after failing at arbitration and waiting for the next bus free state. After this interrupt is reported, the results of the operation set after selection (Automatic selection response mode or user program operation) are reported. • The SPC has already been selected when a command is received and the operation set after reselection is being executed. After the results of the operation being executed are reported, this interrupt is reported.
67_H	01_H	A Command Pause is received during the arbitration phase.
82_H	02_H	The selection/reselection timeout time is exceeded. After the specified number of retries is attempted, this interrupt is reported.

- SET REQ, RESET REQ, DISCONNECT

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected before the command is executed.
	01_H	A reset condition is detected after the command is executed.
60_H	01_H	Operation is completed normally.

- SEND DATA from MPU/DMA
  RECEIVE DATA to MPU/DMA

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected before the transfer is executed.
	01_H	A reset condition is detected while the transfer is being executed. This interrupt is reported after the completion process.*1
	02_H	A reset condition is detected after the transfer is completed.
21_H	01_H	During a DMA transfer, a parity error is detected in input or output data at the DMA interface.*2
22_H	01_H	During a program transfer, a parity error is detected in input or output data at the MPU interface.*2
24_H	01_H	A parity error is detected in input data at the SCSI interface.*2
25_H	01_H	During a DMA transfer, a parity error is detected in input or output data at the DMA interface and the SCSI interface.*2
26_H	01_H	During a program transfer, a parity error is detected in input or output data at the MPU interface and the SCSI interface.*2
29_H	01_H	During a synchronous transfer, more ACK signals are received than the number of REQ signals sent out.*1
$2A_H$	01_H	Because the period of the ACK signals received is short, the SPC cannot keep up.*1 This interrupt can also be reported for asynchronous transfers.
$2B_H$	01_H	During a synchronous transfer, more ACK signals are received than the number of REQ signals sent out and because the period of the ACK signals received is short, the SPC cannot keep up.*1
$2C_H$	01_H	The specified REQ/ACK timeout time is exceeded.*1
42_H	01_H	The generation of an initiator attention condition causes a completion of the transfer at a transfer block boundary.*3
43_H	01_H	The generation of an initiator attention condition causes the suspension of the transfer at a transfer block boundary and then a single message is received.*4
60_H	02_H	The operation is completed normally.
61_H	02_H	After the command is completed normally, an attention condition generated by the initiator is detected.*3
62_H	02_H	After the command is completed normally, an attention condition generated by the initiator is detected and then a single message is received.*4
65_H	01_H	The data block register or data byte register setting values are 0.
67_H	01_H	A **COMMAND PAUSE** command is received.*1

*1 See 5.9.6 for more on the exit process.

*2 In the Byte stop mode, the process is completed at the same time as this state is detected. In the Block stop mode, the transfer is completed after the block where the parity error is detected.

*3 Reported when the Automatic receive mode is not set.

*4 Reported when the Automatic receive mode is set.

- SEND 1-MSG

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected before the transfer.
	01_H	A reset condition is detected during the transfer.*
	02_H	A reset condition is detected after the transfer is completed.
$2C_H$	01_H	The specified REQ/ACK timeout time is exceeded.*
60_H	02_H	The operation is completed normally.
61_H	02_H	After the command is completed, the attention condition generated by the initiator is detected. Reported when the Automatic receive mode is not set.
62_H	02_H	After the command is completed, the attention condition generated by the initiator is detected and a single message is received. Reported when the Automatic receive mode is set.
64_H	01_H	The message length is 33 bytes or longer. MSG not sent.

* Completed at the same time as this state is detected.

- RECEIVE MSG

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected before the transfer.
	01_H	A reset condition is detected during the transfer.*
	02_H	A reset condition is detected after the transfer is completed.
24_H	01_H	A parity error is detected for input data at the SCSI interface.*
$2C_H$	01_H	The specified REQ/ACK timeout time is exceeded.*
60_H	02_H	The operation is completed normally.
61_H	02_H	After the command is completed, the attention condition generated by the initiator is detected. Reported when the Automatic receive mode is not established.
67_H	01_H	The message received is 33 bytes or longer. The message is received up to the second byte and then completed.

* Completed at the same time as this state is detected.

- SEND STATUS

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected before the transfer.
	01_H	A reset condition is detected during the transfer.*
	02_H	A reset condition is detected after the transfer is completed.
$2C_H$	01_H	The specified REQ/ACK timeout time is exceeded.*
60_H	02_H	The operation is completed normally.
61_H	02_H	After the command is completed, the attention condition generated by the initiator is detected. Reported when the Automatic receive mode is not set.
62_H	02_H	After the command is completed, the attention condition generated by the initiator is detected and a single message is received. Reported when the Automatic receive mode is set.

* Completed at the same time as this state is detected.

- RECEIVE CMD

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected before the transfer.
	01_H	A reset condition is detected during the transfer.*
	02_H	A reset condition is detected after the transfer is completed.
24_H	01_H	A parity error is detected for input data at the SCSI interface.*
$2C_H$	01_H	The specified REQ/ACK timeout time is exceeded.*
60_H	02_H	The operation is completed normally.
61_H	02_H	After the command is completed, the attention condition generated by the initiator is detected. Reported when the Automatic receive mode is not set.
62_H	02_H	After the command is completed, the attention condition generated by the initiator is detected and a single message is received. Reported when the Automatic receive mode is set.
65_H	01_H	• The group 6/7 command is not defined. • The group 3/4 command is received. The first byte is received and then the process completed.

* Completed at the same time as this state is detected.

G.3 Common Commands

- All commands

CODE	STEP	SPC Operation
64_H	00_H	• A new command is received while a command is being executed (double reception). After the execution results for the command received first are reported, this interrupt is reported. • When an interrupt cause occurs before this command is executed, the execution cannot be carried out. When an interrupt cause except “REQ Asserted” occurs that is reported during the SPC Ready state, this interrupt is reported after that interrupt cause is reported.
65_H	00_H	An undefined command is received.
66_H	00_H	A parity error is detected in a register established by the host MPU.

- TRANSFER RESET, SET UP REG

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected before the command is executed.
	00_H	A reset condition is detected after the command is executed.
60_H	01_H	The operation is completed normally.
70_H	00_H	A bus free state is detected before the command is executed.
	01_H	A bus free state is detected after the command is executed.

- SCSI RESET

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected before the command is executed.
60_H	01_H	The operation is completed normally.
70_H	00_H	A bus free state is detected before the command is executed.

- INIT DIAG START, TARG DIAG START

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected before the command is executed.
68_H	01_H	No abnormality found in self diagnosis results.
69_H	00_H	Abnormality found in self diagnosis results.

- DIAG END

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected after the command is executed.
60_H	01_H	The operation is completed normally.

- COMMAND PAUSE

CODE	STEP	SPC Operation
67_H	01_H	Command Pause is received.

- SET RST, RESET RST

CODE	STEP	SPC Operation
60_H	01_H	Operation is normally completed.

G.4 Automatic Selection/Reselection Response

- Automatic reselection response

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected before the nexus is established.
	01_H	After the reselection phase, the SPC moves to the message in phase, and during a transfer, a reset condition is detected.*1
24_H	01_H	After the reselection phase, the SPC moves to the message in phase, and a parity error is detected in input data at the SCSI interface.*2 • When the message being received is a 1 byte message or a 2 byte message, this is reported after all of the bytes are received. • When the message being received is an extended message, if the error is detected in the first byte or second byte, this is reported after 2 bytes are received, but if the error is detected in the third byte or thereafter, this is reported after all of the bytes are received.
$2C_H$	01_H	After the reselection phase, the SPC moves to the message in phase but the specified REQ/ACK timeout time is exceeded during the transfer.*1
31_H	01_H	After the reselection phase, the SPC moves to the message in phase but a bus free state is detected during the transfer.*1
32_H	01_H	After the reselection phase, the SPC moves to the message in phase but the target changes phase during the transfer.*1
64_H	00_H	• The SPC is reselected after failing at arbitration and waiting for the next bus free state. After this interrupt is reported, the results of the operation set after reselection (Automatic reselection response mode or user program operation) are reported. • The SPC has already been reselected when a command is received and the operation set after reselection is being executed. After the results of the operation being executed are reported, this interrupt is reported.
67_H	01_H	After the reselection phase, the SPC moves to the message in phase, and the message received is an extended message with a length of 33 bytes or longer. Received through the second byte.*2
70_H	00_H	After a nexus is established, a bus free state is detected before executing a transfer.
90_H	00_H	The SPC is reselected by the target as the initiator. This is reported when the Automatic reselection response mode is not set.
$B0_H$	01_H	The SPC is reselected by the target as the initiator and after the reselection phase, the SPC moves to a phase other than the message in phase.
$B1_H$	02_H	The SPC is reselected by the target as the initiator and after the reselection phase, the SPC moves to the message in phase and a single message is received.

*1 Completed at the same time as this state is detected.

*2 After this command is reported, an interrupt code = $B1_H$ and sequence step = 02_H are also reported.

- Automatic selection response

CODE	STEP	SPC Operation
01_H	00_H	A reset condition is detected before a nexus is established.
	81_H	After the selection phase, the SPC moves to the message out phase but a reset condition is detected during the transfer.*2
	82_H	After the selection phase, an Identify message is received and an attempt is made to receive another message but a reset condition is detected during this transfer.*2
	92_H	After the selection phase, an Identify message is received and then an attempt is made to receive a single CMD but a reset condition is detected during this transfer.*2
	93_H	After the selection phase, an Identify message and then a single CMD are received. After that, an ATN signal is detected from the initiator and the SPC moves to the message out phase but a reset condition is detected during the transfer.*2
	$B1_H$	After the selection phase, the SPC moves to the CMD phase but a reset condition is detected during the transfer.*1
	$B2_H$	After the selection phase, a single CMD is received, the ATN signal from the initiator is detected, and the SPC moves to the message out phase but a reset condition is received during the transfer.*1, *2, *4
24_H	81_H	After the selection phase, the SPC moves to the message out phase but a parity error is detected in input data at the SCSI interface.*2
	82_H	After the selection phase, an Identify message is received and an attempt is made to receive another message but a parity error is detected in input data at the SCSI interface.*2
	92_H	After the selection phase, an Identify message is received and then an attempt is made to receive a single CMD but a parity error is detected in input data at the SCSI interface.*2
	93_H	After the selection phase, an Identify message and then a single CMD are received. After that, an ATN signal is received from the initiator and the SPC moves to the message out phase but a parity error is detected in input data at the SCSI interface.*2
	$B1_H$	After the selection phase, the SPC moves to the CMD phase, and a parity error is detected in input data at the SCSI interface.*1, *2
	$B2_H$	After the selection phase, a single CMD is received, the ATN signal from the initiator is detected, and the SPC moves to the message out phase but a parity error is detected in input data at the SCSI interface.*1, *2, *4
$2C_H$	81_H	After the selection phase, the SPC moves to the message out phase but a parity error is detected in input data at the SCSI interface.*2
	82_H	After the selection phase, an Identify message is received and an attempt is made to receive another message but the specified REQ/ACK timeout time is exceeded.*2
	92_H	After the selection phase, an Identify message is received and then an attempt is made to receive a single CMD but the specified REQ/ACK timeout time is exceeded.*2
	93_H	After the selection phase, an Identify message and then a single CMD are received. After that, an ATN signal is received from the initiator and the SPC moves to the message out phase but the specified REQ/ACK timeout time is exceeded.*2
	$B1_H$	After the selection phase, the SPC moves to the CMD phase but the specified REQ/ACK timeout time is exceeded.*1, *2
	$B2_H$	After the selection phase, a single CMD is received, the ATN signal from the initiator is detected, and the SPC moves to the message out phase but the specified REQ/ACK timeout time is exceeded.*1, *2
64_H	00_H	• The SPC is selected after failing at arbitration and waiting for the next bus free state. After this interrupt is reported, the results of the operation established after selection (Automatic selection response mode or user program operation) are reported. • The SPC has already been selected when a command is received and the operation established after selection is being executed. After the results of the operation being executed are reported, this interrupt is reported.

*1 Either message or CMD phase is assigned to the SEL/RESEL operation mode setting register (WRITE/READ) after the selection phase.
*2 Completed at the same time as the state is detected.
*3 Only the message phase is assigned to the SEL/RESEL operation mode setting register (WRITE/READ) after the selection phase.
*4 When the Automatic Receive mode is set.
*5 When the Automatic Receive mode is not set.
*6 When Automatic Selection Response mode is not set.

(Continued)

CODE	STEP	SPC Operation
67_H	81_H	After the selection phase and the SPC moves to the message output phase, the message received has a length of 33 bytes or longer. 2 bytes of the message are received and then this process is completed.
	82_H	After the selection phase, an Identify message is received, an ATN signal from the initiator is detected, and the SPC moves to the message out phase. The message received has a length of 33 bytes or longer.2 bytes of the message are received and then this process is completed.
	93_H	After the selection phase, an Identify message is received and then a single CMD is received, an ATN signal from the initiator is detected, and the SPC moves to the message out phase. The message received has a length of 33 bytes or longer. 2 bytes of the message are received and then this process is completed.
	$B2_H$	After the selection phase, a single CMD is received, an ATN signal from the initiator is detected, and the SPC moves to the message out phase. The message received has a length of 33 bytes or longer.*1, *4 2 bytes of the message are received and then this process is completed.
80_H	00_H	The SPC is selected by the initiator as the target. *6
81_H	00_H	The SPC is selected by the initiator as the target and the ATN signal produced by the initiator is detected. *6
$A0_H$	81_H	The SPC is selected by the initiator as the target, but the ATN signal is not asserted in the selection phase.*3
$A1_H$	$A2_H$	The SPC is selected by the initiator as the target and after the selection phase, a single message other than the Identify message is received.
	$A3_H$	The SPC is selected by the initiator as the target and after the selection phase, a single message other than the Identify message is received, but the ATN signal remains asserted.
$A2_H$	82_H	The SPC is selected by the initiator as the target and after the selection phase, the Illegal Identify message is received, but the ATN signal remains asserted.
	92_H	The SPC is selected by the initiator as the target and after the selection phase, the Illegal Identify message is received.
$A3_H$	83_H	The SPC is selected by the initiator as the target and after the selection phase, the Identify message is received and then another single message is received.
	84_H	The SPC is selected by the initiator as the target and after the selection phase, the Identify message is received and then another single message is received, but the ATN signal remains asserted.
$A4_H$	$B1_H$	The SPC is selected by the initiator as the target and after the selection phase, an attempt is made to receive a single CMD but since it is an undefined CDB, only 1 byte is received.*1
$A5_H$	92_H	The SPC is selected by the initiator as the target and after the selection phase, the Identify message is received and an attempt is made to receive a single CMD but since it is an undefined CDB, only 1 byte is received.
$A6_H$	$B2_H$	The SPC is selected by the initiator as the target and after the selection phase, a single CMD is received.*1
$A7_H$	93_H	The SPC is selected by the initiator as the target and after the selection phase, the Identify message is received and then a single CMD is received.
$A8_H$	$B1_H$	The SPC is selected by the initiator as the target and after the selection phase, a single CMD is received and then the ATN signal from the initiator is detected and a single message is received.*1, *2, *4
	$B4_H$	The SPC is selected by the initiator as the target and after the selection phase, a single CMD is received and then the ATN signal from the initiator is detected and a single message is received but the ATN signal asserted again.*1, *4

*1 Either the message or CMD phase is assigned to the SEL/RESEL operation mode setting register (WRITE/READ) after the selection phase.
*3 Only the message is assigned to the SEL/RESEL operation mode setting register (WRITE/READ) after the selection phase.
*4 When the Automatic Receive mode is set.
*6 When the Automatic Selection Response mode is not set.

(Continued)

CODE	STEP	SPC Operation
$A9_H$	94_H	The SPC is selected by the initiator as the target and after the selection phase, the Identify message is received and a single CMD is received and then the ATN signal from the initiator is detected and a single message is received.
	95_H	The SPC is selected by the initiator as the target and after the selection phase, the Identify message is received and a single CMD is received and then the ATN signal from the initiator is detected and a single message is received but the ATN signal remains asserted.
AA_H	$B2_H$	The SPC is selected by the initiator as the target and after the selection phase, a single CMD is received and then the ATN signal from the initiator is detected.*1, *5

*1 Either the message or CMD phase is assigned to the SEL/RESEL operation mode setting register (WRITE/READ) after the selection phase.
*5 When the Automatic Receive mode is not set.

- Other interrupt codes and sequence steps

Take the following steps if interrupt codes and sequence steps other than those described in this section are reported.

(1) When a nexus is not established.....Issue Software Reset command

(2) When a nexus is established.....After executing **`Disconnect`** or **`SCSI Reset`** command, Issue Software Reset command.

Also, SPC may report an interrupt code "60_H" and command step code "00_H" repeatedly when error that SPC can not handle occur.

In this case the SPC normal operation is not guaranteed and so, the following process should be taken:

(1) When a nexus is not established..... Issue Software Reset command.

(2) When a nexus is established..... a. After executing Software Reset command, issue SCSI Reset command (for Initiator).

b. Issue Software Reset command (for Target).

Worldwide Headquarters

Japan

Fujitsu Limited

Tel: +81 44 754 3753
Fax: +81 44 754 3332

1015 Kamiodanaka
Nakaharaku
Kawasaki 211
Japan

http://www.fujitsu.co.jp/

Asia

Fujitsu Microelectronics Asia PTE Limited

Tel: +65 336 1600
Fax: +65 336 1609

No. 51 Bras Basah Road
Plaza by the Park
#06-04/07
Singapore 0718

http://www.fsl.com.sg/

USA

Tel: +1 408 922 9000
Fax: +1 408 922 9179

Fujitsu Microelectronics Inc
3545 North First Street
San José CA 95134-1804
USA

Tel: +1 800 866 8608
Fax: +1 408 922 9179

Customer Response Center
Mon-Fri: 7am-5pm (PST)

http://www.fujitsumicro.com/

Europe

Tel: +49 6103 6900
Fax: +49 6103 690122

Fujitsu Mikroelektronik GmbH
Am Siebenstein 6-10
D-63303 Dreieich-Buchschlag
Germany

http://www.fujitsu.ede.com/

FJ ID Number AM15-10410-1E

CP-SCSI-DM-20378-10/96