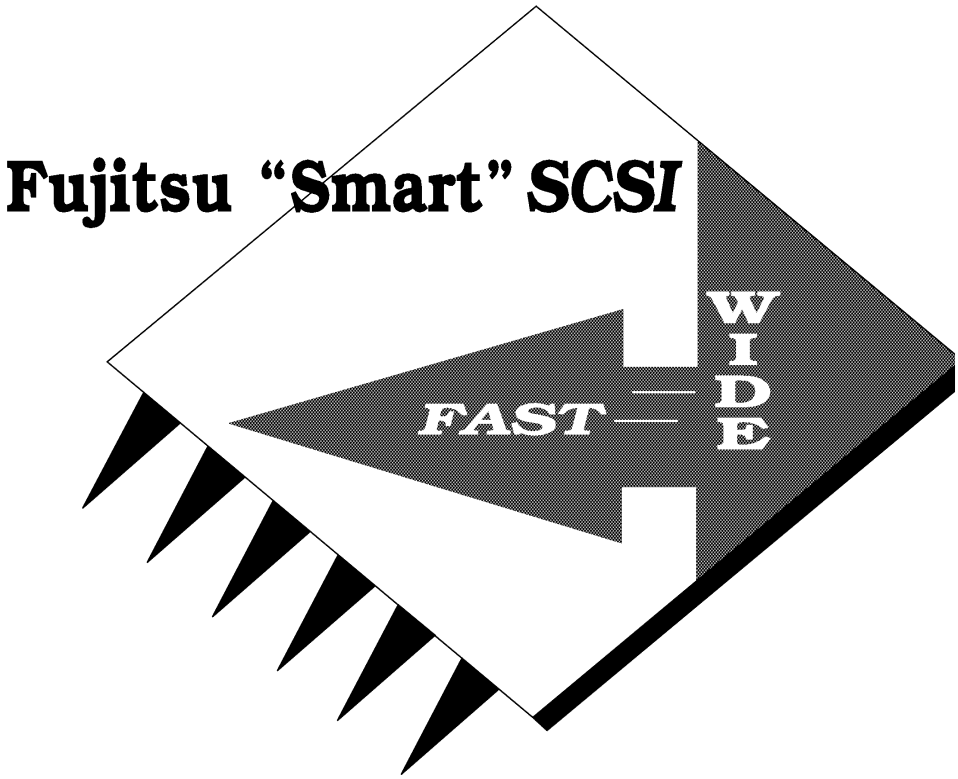
FUJITSU

MB86605

User's Manual

SCSI 2 Protocol Controller with PCI Bus Interface and 16–bit Wide SCSI Interface



Edition 1.1
Preliminary

FUJITSU MICROELECTRONICS

MB86605

SCSI PROTOCOL CONTROLLER WITH PCI LOCAL BUS INTERFACE

USER'S MANUAL

PRELIMINARY

Edition 1.1 January 1996

Table of Contents

1. GENERAL

The MB86605 is an intelligent SCSI protocol controller (SPC) conforming to the ANSI (SCSI-2) standard and integrating a PCI local bus interface function. The specification of SCSI controller block is based on the MB86603's one which is another SCSI-2 protocol controller realizes the wide SCSI transfer, but the internal operation speed and the performance are improved on the MB86605. The MB86605 is capable of transferring up to 20Mbyte/sec at the wide high speed synchronous mode. As for the SCSI bus pins, a totem pole type single-ended driver/receiver is incorporated in the device so that it can drive the SCSI bus directly. Furthermore, the MB86605 is capable of connecting the external differential type driver/receiver.

The SCSI bus sequence is controlled by commands issued from a system side. So, it supports sequential commands that perform the phase-to-phase sequences to reduce the overhead of system's sequence operations.

As another key feature to reduce the system overhead, the device has a 2Kbytes user program memory to store the user program with the commands. Due to this, all the SCSI bus sequences including the data transfer can be performed automatically.

As the system interface block, it incorporates a 32-bit PCI local bus interface that easily realizes the SCSI interface on the motherboards of PCI bus based PCs and WSs, in addition to a 16-bit separate MPU and DMA buses. For the on-chip PCI bus interface, the MB86605 also incorporates a 32-bit DMA controller that is capable of supporting the scatter-gather function so that the data transfers can be controlled by both user program and the system sides.

The device is fabricated by the advanced CMOS process and is housed in an 144-pin plastic Shrink Quad Flat Package (Suffix : –PMT).

1.1 FEATURES

SCSI Protocol Controller Block :

- Operable as initiator and target
- WIDE and FAST data transfers
  - – Synchronous transfer (max. 20 Mbytes/s: Up to 32 offset values can be set.)
  - – Asynchronous transfer (max. 10 Mbytes/s)
- 64-byte FIFO register for data phase
- Two types (send-only and receive-only) of 32-byte type data buffers for message, command, and status phases (MCS Buffers)
- On-chip totem pole type SCSI single-ended driver/receiver
- Supports external SCSI differential driver/receiver
- On-chip memory to store transfer parameters (up to 15 connected devices) for each ID
- On-chip 16-bit transfer block counter and 24-bit transfer byte counter
  - – Maximum Transfer Byte : 1Tbyte at fixed length data transfer
    : 16Mbyte at variable length data transfer
- Supports various control commands :

– Sequential Commands : can perform phase-to-phase sequential operations (functions only when issuing from a system side.)
– Discrete Commands : can perform any desired sequence to program in the user program memory
– Data Transfer Commands : can program the transfer data length at the user program operation.

- On-chip direct control register for SCAM (SCSI Configured AutoMagically)
- Supports Multi Selection/Reselection Responses
  – Selection and Reselection responses can be done to plural IDs.
- On-chip 2Kbyte User Program Memory
  – Two Modes : 2Kbyte x 1 bank and 1Kbyte x 2 banks (While 1Kbyte x 2 banks are selected, host system can access another bank even if the user program is executing.)
  – Access to User program : Burst transfer via I/O access port
  : Direct access to 2Kbyte user program memory (only for PCI bus I/F mode)
- User Selectable Interrupt Report
  – Unnecessary interrupt reports can be disabled depending on user's applications to reduce a system ISR overhead.
- Two automatic receive modes
  – Initiator: Can automatically receive information for new phase to which target switched
  – Target: Can automatically receive attention condition generated by initiator
- Automatic selection/reselection
  – For Command issues : automatically performs to receive MSG/CMD to the selection/reselection request from partner device
  – For user program operation : pauses the program currently executed and automatically jumps to the specified selection/reselection routine to the selection/reselection request from partner device.
- Operation Clock
  – System Clock : Max 40MHz
  – Internal Processor Clock : Max 20MHz

<u>System Specifications</u>

- Separate MPU and DMA buses called 16-bit Bus Mode
  – Directly connectable to 68-series or 80-series 16-bit MPU.
  – Two transfer modes (Program transfer and DMA transfer (slave mode))
- PCI Bus Interface Mode
  – Directly connectable to the 32-bit PCI local bus.
  – On-chip 32-bit DMAC to support the scatter-gather function.
  – Supports the PERR&SERR function
  – Supports the INTA# Interrupt Signals
  – Max. 64bytes burst transfer
  – PCI system clock : Max. 33MHz

- Data Bus Parity and Address Bus Parity (only for PCI bus interface mode) generation/check function

Additional Information

- Compact 144-Pin Plastic Shrink Quad Flat Package (SQFP, Package Suffix : –PMT)
- Supply Voltage : 5V +/– 5%

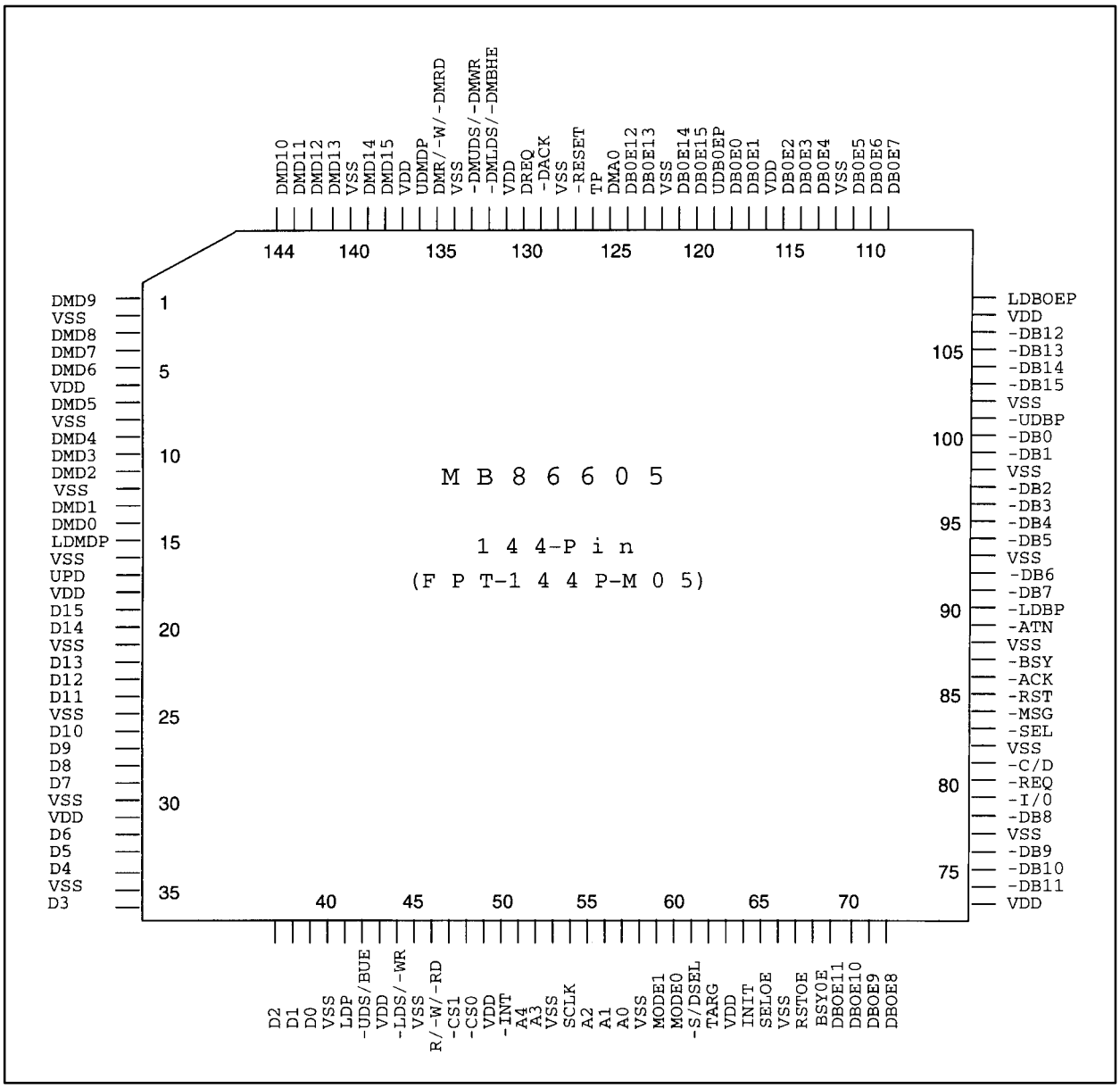
1.2 ABBREVIATIONS

SCSI-OUTPUT: Data output transfer from SPC to SCSI bus
SCSI-INPUT: Data input transfer from SCSI bus to SPC
Command: Internal commands for SPC
Host MPU: Microprocessor controlling SPC
MCS: Message (MSG), Command (CMD), Status
xxH or xxh: Hexadecimal number
FIFO: First-In First-Out register
CDB: Command Descriptor Block

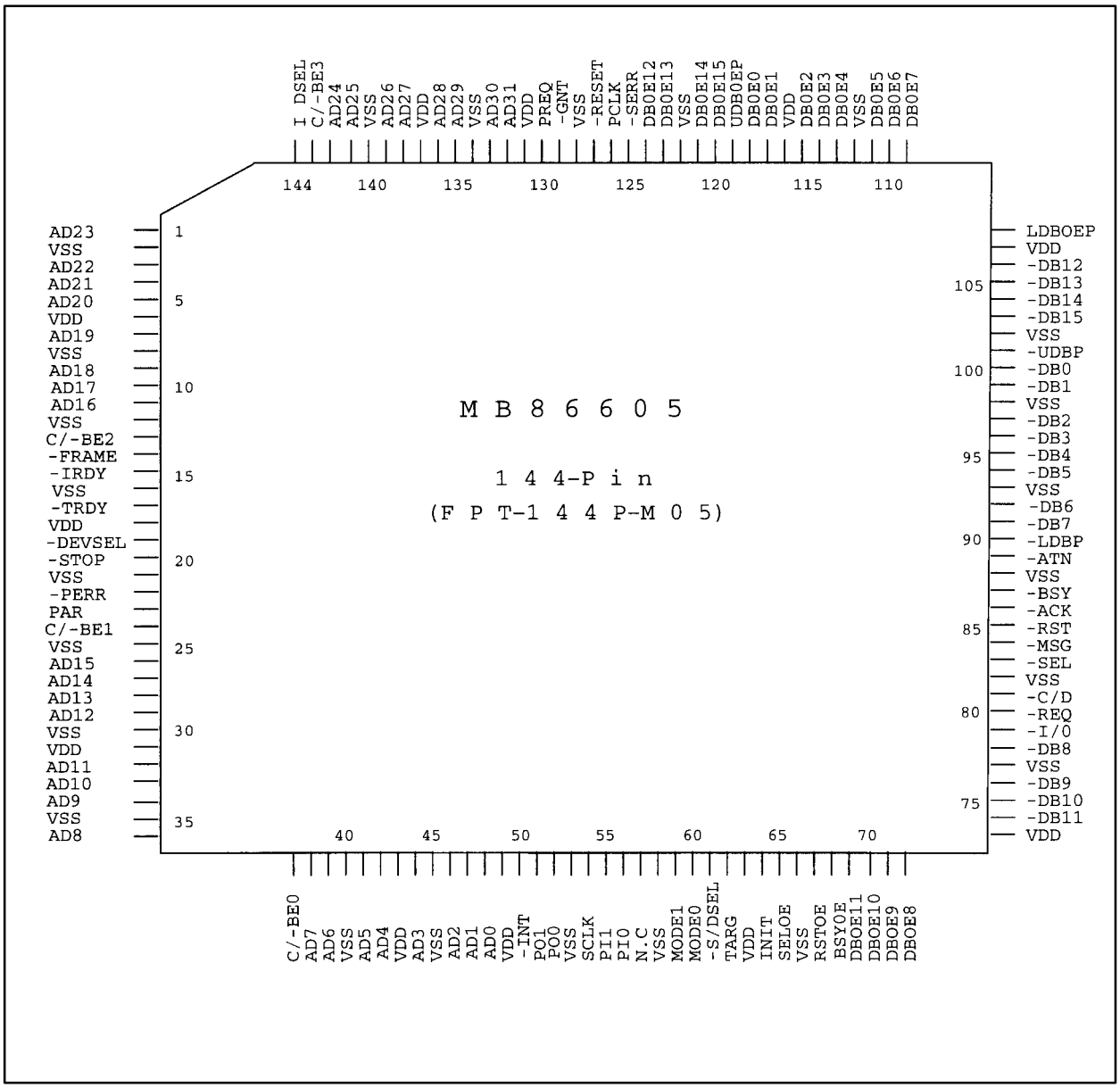
2. PACKAGES AND PIN ASSIGNMENT

2.1. PIN ASSIGNMENT

1) 16-Bit Bus Mode



2) PCI Bus Interface Mode



2.2. PIN LIST

Because the MB86605 has two system interface modes, there is a difference in pin name and function between 16-bit bus (either 68 or 80 series interface) and PCI bus interface modes as listed in the table below.

Pin No.	16-Bit Bus Mode				PCI Bus I/F Mode		Pin No.	16-Bit Bus Mode				PCI Bus I/F Mode	
	Mode 0 (68 I/F)		Mode 1 (80 I/F)		Mode 3 (PCI I/F)			Mode 0 (68 I/F)		Mode 1 (80 I/F)		Mode 3 (PCI I/F)	
	I/O	Pin Name	I/O	Pin Name	I/O	Pin Name		I/O	Pin Name	I/O	Pin Name	I/O	Pin Name
1	I/O	DMD9			I/O	AD23	24	I/O	D11			I/O	C/–BE1
2	–	V_{SS}			–	V_{SS}	25	–	V_{SS}			–	V_{SS}
3	I/O	DMD8			I/O	AD22	26	I/O	D10			I/O	AD15
4	I/O	DMD7			I/O	AD21	27	I/O	D9			I/O	AD14
5	I/O	DMD6			I/O	AD20	28	I/O	D8			I/O	AD13
6	–	V_{DD}			–	V_{DD}	29	I/O	D7			I/O	AD12
7	I/O	DMD5			I/O	AD19	30	–	V_{SS}			–	V_{SS}
8	–	V_{SS}			–	V_{SS}	31	–	V_{DD}			–	V_{DD}
9	I/O	DMD4			I/O	AD18	32	I/O	D6			I/O	AD11
10	I/O	DMD3			I/O	AD17	33	I/O	D5			I/O	AD10
11	I/O	DMD2			I/O	AD16	34	I/O	D4			I/O	AD9
12	–	V_{SS}			–	V_{SS}	35	–	V_{SS}			–	V_{SS}
13	I/O	DMD1			I/O	C/–BE2	36	I/O	D3			I/O	AD8
14	I/O	DMD0			I/O	–FRAME	37	I/O	D2			I/O	C/–BE0
15	I/O	LDMDP			I/O	–IRDY	38	I/O	D1			I/O	AD7
16	–	V_{SS}			–	V_{SS}	39	I/O	D0			I/O	AD6
17	I/O	UDP			I/O	–TRDY	40	–	V_{SS}			–	V_{SS}
18	—	V_{DD}			—	V_{DD}	41	I/O	LDP			I/O	AD5
19	I/O	D15			I/O	–DEVSEL	42	I	–UDS	I	–BHE	I/O	AD4
20	I/O	D14			I/O	–STOP	43	–	V_{DD}			–	V_{DD}
21	—	V_{SS}			—	V_{SS}	44	I	–LDS	I	–WR	I/O	AD3
22	I/O	D13			I/O	–PERR	45	–	V_{SS}			–	V_{SS}
23	I/O	D12			I/O	PAR	46	I	R/–W	I	–RD	I/O	AD2

Note : A short bar "–" added to the pin name in the above table means the active-low signal pin.

PIN LIST **(CONTINUED)**

Pin No.	16-Bit Bus Mode				PCI Bus I/F Mode		Pin No.	16-Bit Bus Mode				PCI Bus I/F Mode	
	Mode 0 (68 I/F)		Mode 1 (80 I/F)		Mode 3 (PCI I/F)			Mode 0 (68 I/F)		Mode 1 (80 I/F)		Mode 3 (PCI I/F)	
	I/O	Pin Name	I/O	Pin Name	I/O	Pin Name		I/O	Pin Name	I/O	Pin Name	I/O	Pin Name
47	I	–CS1			I/O	AD1	72	O	DBOE8				
48	I	–CS0			I/O	AD0	73	–	V_{DD}				
49	–	V_{DD}			–	V_{DD}	74	I/O	–DB11				
50	O/OD	–INT					75	I/O	–DB10				
51	I	A4			O	PO1	76	I/O	–DB9				
52	I	A3			O	PO0	77	–	V_{SS}				
53	–	V_{SS}			–	V_{SS}	78	I/O	–DB8				
54	I	SCLK					79	I/O	–I/O				
55	IU	A2			IU	PI1	80	I/O	–REQ				
56	IU	A1			IU	PI0	81	I/O	–C/D				
57	IU	A0			IU	N.C.	82	–	V_{SS}				
58	–	V_{SS}			–	V_{SS}	83	I/O	–SEL				
59	I	MODE1					84	I/O	–MSG				
60	I	MODE0					85	I/O	–RST				
61	I	–S/DSEL					86	I/O	–ACK				
62	O	TARG					87	I/O	–BSY				
63	–	V_{DD}					88	–	V_{SS}				
64	O	INIT					89	I/O	–ATN				
65	O	SELOE					90	I/O	–LDBP				
66	O	V_{SS}					91	I/O	–DB7				
67	O	RSTOE					92	I/O	–DB6				
68	–	BSYOE					93	–	V_{SS}				
69	O	DBOE11					94	I/O	–DB5				
70	O	DBOE10					95	I/O	–DB4				
71	O	DBOE9					96	I/O	–DB3				

Note : A short bar "–" added to the pin name in the above table means the active-low signal pin.

PIN LIST **(CONTINUED)**

Pin No.	16-Bit Bus Mode				PCI Bus I/F Mode		Pin No.	16-Bit Bus Mode				PCI Bus I/F Mode	
	Mode 0 (68 I/F)		Mode 1 (80 I/F)		Mode 3 (PCI I/F)			Mode 0 (68 I/F)		Mode 1 (80 I/F)		Mode 3 (PCI I/F)	
	I/O	Pin Name	I/O	Pin Name	I/O	Pin Name		I/O	Pin Name	I/O	Pin Name	I/O	Pin Name
97	I/O	–DB2					121	O	DBOE14				
98	–	V_{SS}					122	–	V_{SS}				
99	I/O	–DB1					123	O	DBOE13				
100	I/O	–DB0					124	O	DBOE12				
101	I/O	–UDBP					125	I	DMA0			OD	–SERR
102	I/O	V_{SS}					126	I	TP			I	PCLK
103	–	–DB15					127	I	–RESET				
104	I/O	–DB14					128	–	V_{SS}				
105	I/O	–DB13					129	I	–DACK			I	–GNT
106	I/O	–DB12					130	O	DREQ			O	–PREQ
107	–	V_{DD}					131	–	V_{DD}				
108	O	LDBOEP					132	I	–DMUDS	I	–DMBHE	I/O	AD31
109	O	DBOE7					133	I	–DMLDS	I	–DMWR	I/O	AD30
110	O	DBOE6					134	–	V_{SS}				
111	O	DBOE5					135	I	DMR/–W	I	–DMRD	I/O	AD29
112	–	V_{SS}					136	I/O	UDMDP			I/O	AD28
113	O	DBOE4					137	–	V_{DD}				
114	O	DBOE3					138	I/O	DMD15			I/O	AD27
115	O	DBOE2					139	I/O	DMD14			I/O	AD26
116	O	V_{DD}					140	–	V_{SS}				
117	–	DBOE1					141	I/O	DMD13			I/O	AD25
118	O	DBOE0					142	I/O	DMD12			I/O	AD24
119	O	UDBOEP					143	I/O	DMD11			I/O	C/–BE3
120	O	DBOE15					144	I/O	DMD10			I	IDSEL

Note : A short bar "–" added to the pin name in the above table means the active-low signal pin.

2.3. PIN DESCRIPTION

2.3.1. SCSI INTERFACE

Pin (Signal) Name	I/O	Function
–ATN, –MSG, –C/D, –I/O	I/O	**These are the SCSI control signal input and output pins.** They can be connected directly to a single-ended SCSI connector. Either open-drain or totem pole output can be selected.
–REQ, –ACK	I/O	**These are the SCSI control signal input and output pins**. They can be connected directly to a single-ended SCSI connector. The output buffer is the totem pole type.
BSYOE SELOE RSTOE	O	**These are used for output control of SCSI control signals.** They should be used as control signals for the external differential driver/receiver circuit.
–BSY –SEL –RST	I/O	**These are the SCSI control signal input and output pins.** They can be connected directly to a single-ended SCSI connector. The output buffer is the open-drain type.
DBOE15 to DBOE8, UDBOEP, DBOE7 to DBOE0, LDBOEP	O	**These are used for output control of SCSI data bus signals.** They should be used as control signals for the external differential driver/receiver circuit.
–DB15 to –DB8, –UDBP –DB7 to –DB0, –LDBP	I/O	**These are used to input and output SCSI data bus signals.** They can be connected directly to a single-ended SCSI connector. Either open-drain or totem pole output buffer can be selected.
INIT TRAG	O	**These are used to output signals indicating the chip operating status.** They should be used as control signals for the external differential driver/receiver circuit.
–S/DSEL	I	**These are used to input signals for selecting the chip operation modes.** Single-ended: Input 0. Differential-ended: Input 1. While 0 is input to this pin, all the SCSI control signals, data bus output control signals, INIT, and TARG signals are fixed with L level.
SCLK	I	**This pin is used for a system clock input for SCSI protocol controller block.** (Max. 40MHz)

Note : A short bar "–" added to the pin name in the above table means the active-low signal pin.

2.3.2. 16-BIT BUS MODE – MPU INTERFACE

The MPU interface can connect the MPU input/output signals directly by setting the 68-series or 80-series mode.

Pin (Signal) Name	I/O	Function
–CS0	I	**Input signals for the MPU to select the SPC as the I/O device.**
–CS1	I	**Input select signals (external circuit select signals) for the MPU to input and output the DMA data bus data via the SPC.**
D15 to D8, UDP	I/O	**Upper byte and parity of data bus** When –CS0 input valid: I/O ports for internal registers in SPC When –CS1 input valid: I/O ports for DMA bus data
D7 to D0, LDP	I/O	**Lower byte and parity of data bus** When –CS0 input valid: I/O ports for internal registers in SPC When –CS1 input valid: I/O ports for DMA bus data
A4 to A0	I/IU	**These are used to input addresses for selecting the internal registers.**
–RD (R/–W)	I	In 80-series mode: This is used to input the –IORD or –RD signal for reading data from the SPC to the MPU. In 68-series mode: This is used to input the R/W control signal for reading and writing data from the MPU to the SPC.
–WR (–LDS)	I	In 80-series mode: This is used to input the –IOWR or –WR signal for writing data from the MPU to the SPC. In 68-series mode: This is used to input the –LDS signal output by the MPU when the lower byte of the data bus is valid.
–BHE (–UDS)	I	In 80-series mode: This is used to input the –BHE signal output by the MPU when the upper byte of the data bus is valid. In 68-series mode: This is used to input the –UDS signal output by the MPU when the upper byte of the data bus is valid.

Note : A short bar "–" added to the pin name in the above table means the active-low signal pin.

2.3.3. 16-BIT BUS MODE – DMA INTERFACE

Like the MPU interface, the DMA interface has the input/output signals for the 68 series or 80 series.

Pin (Signal) Name	I/O	Function
DREQ	O	**Output DMA transfer request signals to the DMAC.** DMA data transfer between the SPC and memory is requested.
–DACK	I	**Input DMA-enabling signals from the DMAC.** When the DMA enabling signal is active, DMA reading and writing are executed.
DMD15 to 8 UDMDP	I/O	**Upper byte and parity of DMA data bus** When CS1 input valid: The MPU data bus is directly connected. When 80-series mode: The 2nd data is input/output. When 68-series mode: The 1st data is input/output.
DMD7 to 0 LDMDP	I/O	**Lower byte and parity of DMA data bus** When CS1 input is valid: The MPU data bus is directly connected. When 80-series mode : The 2nd data is input/output. When 68-series mode: The 1st data is input/output.
–IORD (DMR/–W)	I	In 80-series mode: This is used to input the –IORD or –RD signal for outputting data from the SPC to the DMA bus. In 68-series mode: This is used to input the R/–W control signal for outputting and inputting data from the DMAC to the SPC.
–IOWR (–DMLDS)	I	In 80-series mode: This is used to input the –IOWR or –WR signal for inputting data from the DMA bus to the SPC. In 68-series mode: This is used to input the –LDS signal output by the DMAC when the lower byte of the DMA data bus is valid.
–DMBHE (–DMUDS)	I	In 80-series mode: This is used to input the –BHE signal output by the DMAC when the upper byte of the DMA data bus is valid. In 68-series mode: This is used to input the –UDS signal output by the DMAC when the upper byte of the DMA data bus is valid.
DMA0	I	**Input the address data A0 signal output by the DMAC in the 80-series mode.** In 68-series mode: Connect to power supply pin (V_{DD}).
TP (Transfer Permission)	I	**Input DMA-transfer-enabling signals.** When the TP signal is active, the SPC performs the DMA transfer. When this signal becomes inactive during DMA transfer, the transfer stops temporarily at the block boundary.

Note : A short bar "–" added to the pin name in the above table means the active-low signal pin.

2.3.4. PCI BUS INTERFACE MODE

Pin (Signal) Name	I/O	Function
–PREQ	O	**This pin is used to request the bus arbiter for use of the bus.**
–GNT	I	**This is the response signal input pin to the REQ signal from the bus arbiter.**
AD31 to AD0	I/O	**PCI 32-bit address and data multiplexed pins.**
C/–BE3 to C/–BE0	I/O	**Bus command and Byte Enable signals multiplexed pins.**
PAR	I/O	**This is an even parity signal pin for the AD31 to AD0 and C/–BE3 to C/–BE0 signals. This PAR signal becomes valid after one clock.**
–FRAME	I/O	**This is a frame signal pin that indicates data are transferring on the bus**
–TRDY	I/O	**Data Ready signal of Target side.**
–IRDY	I/O	**Data Ready signal of Initiator (Bus master) side.**
–STOP	I/O	**This is a stop request signal to stop the data transfer from target to master.**
–DEVSEL	I/O	**Device select pin. While the device is a target, this pin outputs the select signal that indicates the self device is selected. While the device is a master, this pin functions as an input pin to indicate that a device on the bus is selected.**
IDSEL	I	**This is a chip select signal that indicates the configuration access.**
PCLK	I	**PCI bus clock input pin. The maximum clock frequency is 33MHz.**
–PERR	I/O	**Data parity error input and output pin.**
–SERR	OD	**Address parity error output pin.**

Note : A short bar "–" added to the pin name in the above table means the active-low signal pin.

2.3.5. OTHER SIGNALS

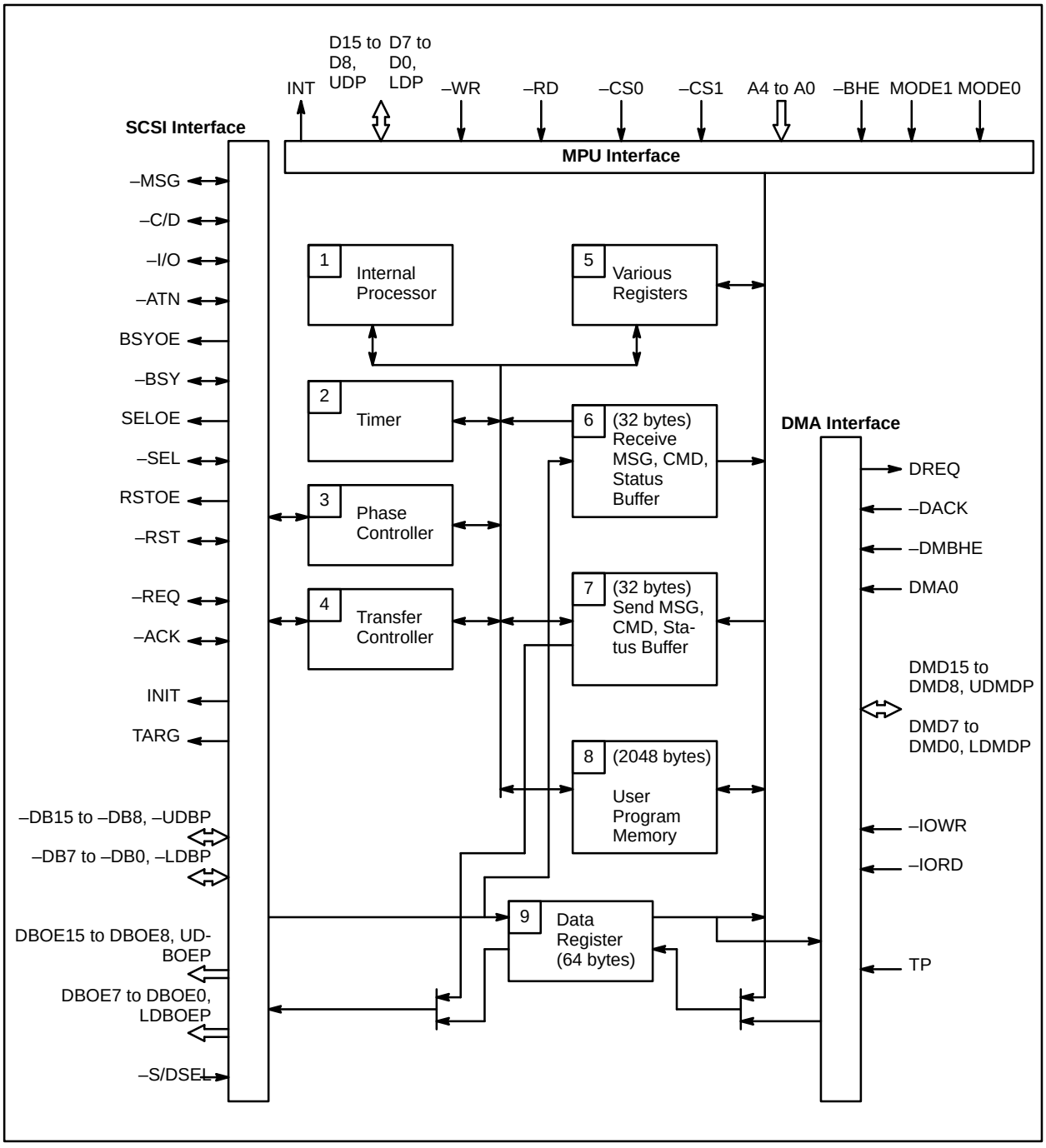
<table>
<tr><th>Pin (Signal) Name</th><th>I/O</th><th>Function</th></tr>
<tr><td>–RESET</td><td>I</td><td>This is used to input system reset signals.</td></tr>
<tr><td>MODE1,
MODE0</td><td>I</td><td>These pins are used for setting the device operation mode as listed in the table below.
<table>
<tr><th>MODE1</th><th>MODE0</th><th>Operation Mode</th></tr>
<tr><td>0</td><td>0</td><td>16-bit bus mode (68 series mode)</td></tr>
<tr><td>0</td><td>1</td><td>16-bit bus mode (80 series mode)</td></tr>
<tr><td>1</td><td>0</td><td>Reserved</td></tr>
<tr><td>1</td><td>1</td><td>PCI bus interface mode</td></tr>
</table></td></tr>
<tr><td>–INT</td><td>O/OD</td><td>Interrupt output pin. Either totem pole or open-drain output buffer can be selected. This pin has an internal weak pull-up resistor.</td></tr>
<tr><td>PO1, PO0</td><td>O</td><td>General purpose output ports that can control the external active SCSI bus terminator etc. Initial signal level on each pin is “L”. Thoese pins are available only for PCI bus interface mode.</td></tr>
<tr><td>PI1, PI0</td><td>IU</td><td>General purpose input ports. Available only for PCI bus interface mode.</td></tr>
<tr><td>V_{DD}</td><td>—</td><td>Power supply pin</td></tr>
<tr><td>V_{SS}</td><td>—</td><td>Ground pin</td></tr>
<tr><td>N.C.</td><td>—</td><td>No connection and unused pins. These pins exist on the only PCI bus mode. These are internally pulled-up, and do not connect to the pins.</td></tr>
</table>

I : Input pin
O : Output pin
I/O : Input and output pin
OD : Open-drain output pin
IU : Input pin with pull-up resistor

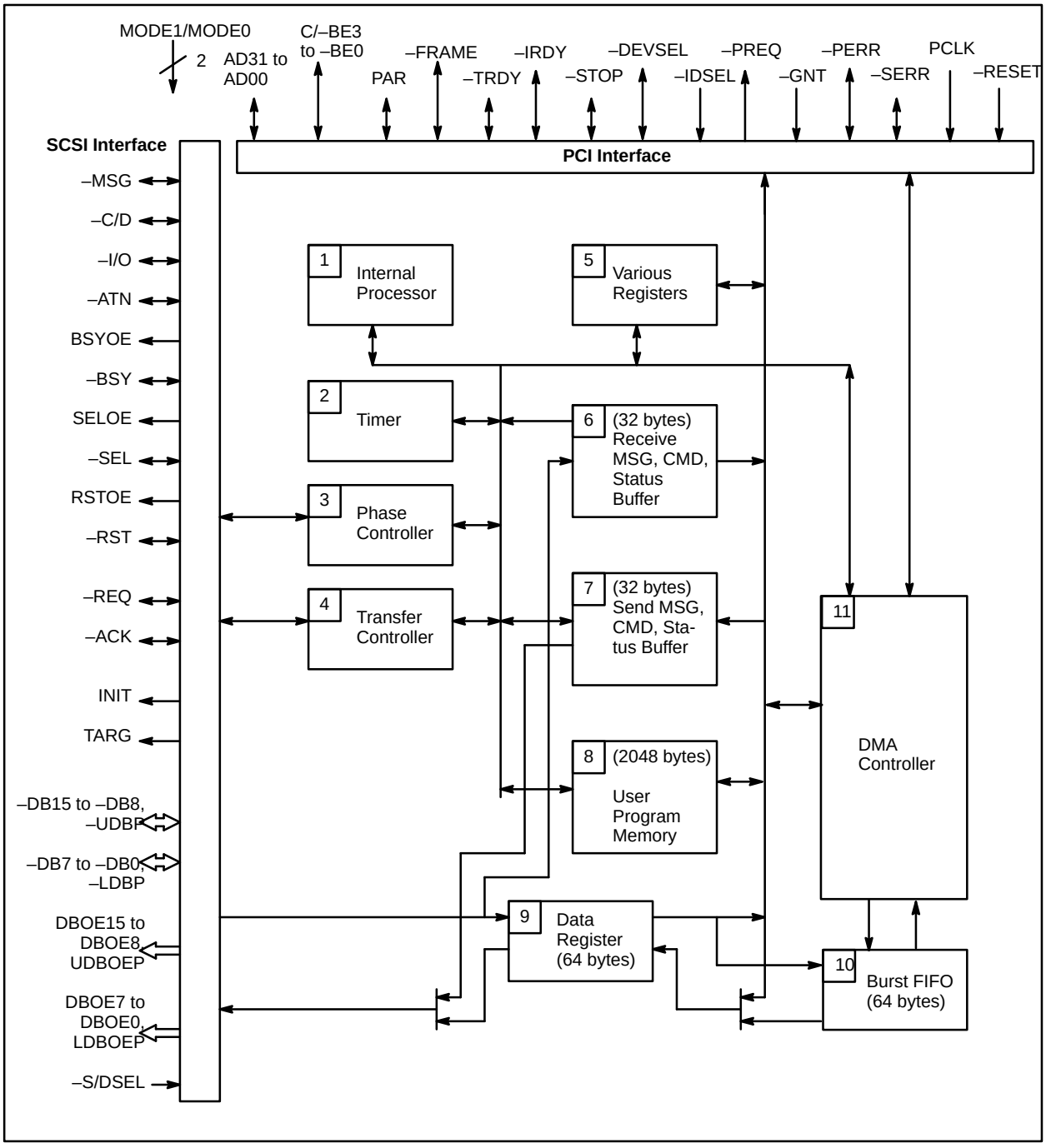
Note : A short bar “–” added to the pin name in the above table means the active-low signal pin.

3. FUNCTIONS

3.1. BLOCK DIAGRAM – 16-Bit Bus Mode



3.2. BLOCK DIAGRAM – PCI Bus Interface Mode



Note : A short bar "–" added to the pin name in the above diagram indicates the active-low signal pin.

3.3. BLOCK FUNCTIONS

1. Internal Processor

This processor provides the sequence control between each phase.

2. Timer

This timer manages the timing parameter, specified by SCSI standard:

- REQ/ACK assertion time for data at asynchronous transfer
- Selection/reselection retry time
- Selection/reselection timeout time
- REQ/ACK timeout time during transfer

Asynchronous transfer (target)	Time required for initiator to assert ACK signal after asserting REQ signal
Asynchronous transfer (initiator)	Time required for target to negate REQ signal after asserting ACK signal
Synchronous transfer (target only)	Time required for target to receive ACK signal for setting offset value to 0 from initiator after sending REQ signal

3. Phase Controller

Controls the arbitration, selection/reselection, data-in/out, command, status, and message-in/out phases executed on the SCSI bus.

4. Transfer Controller

This controller controls the information (data, command, status, message) transfer phases executed on the SCSI bus.

There are two types of transfer for executing the information transfer phases.

- Asynchronous transfer: Control by interlocking REQ and ACK signals
- Synchronous transfer: Control with maximum of 32-byte offset value in data-in/out phase

Depending on the data migration, there are the following two modes.

- Program transfer: Performed via MPU interface using data registers
- DMA transfer: Performed via DMA interface using DREQ and $\overline{\text{DACK}}$ pins

In synchronous transfer mode, the tansfer parameters can be saved for each ID and are automatically set when the data phase is started. The transfer byte count is determined by block length times the number of blocks.

- Transfer mode
- Minimum cycle period of REQ or ACK signal sent from SPC in synchronous transfer
- Maximum value between REQ and ACK signals in synchronous transfer

5. Control Registers

- Command register
  Specifies each command with an 8-bit code.
  When using the user program, specify “1” at the Bit 7. The lower 7 bits (Bit6 to Bit0) are invalid .
- Nexus status register
  Indicates the chip’s operating condition, the nexused partner’s ID, and data register status.
- SCSI control signal status register
  Indicates the status of SCSI control signals.
- Interrupt status register
  Indicates the interrupt status with an 8-bit code.
- Command step register
  Indicates the execution status of each command with an 8-bit step code.
  Error causes can be analyzed by referencing the interrupt status register and this register.
- Group 6/7 command length setting register
  Sets the group 6/7 command length not defined in the SCSI standard.
  Setting this register determines the group 6/7 command length.

6. Receive MSG, CMD, Status Buffer (Receive MCS Buffer)

This is a 32-byte receive-only information buffer that holds the information for the message, command, and status received from the SCSI bus.

7. Send MSG, CMD, Status Buffer (Send MCS Buffer)

This is a 32-byte send-only information buffer that holds the information for the message, command, and status sent on the SCSI bus.

8. User Program Memory

This is a 2048-byte program memory that stores programmable commands. It can be configured as a 1024-byte x 2 banks or 2048-byte x 1 bank.

9. Data Register

This is a 64-byte FIFO data register that holds data in the data phase executed on the SCSI bus.

10. Burst FIFO

64-byte FIFO type data buffer to perform burst transfer during the PCI bus interface mode. The MB86005 features a 128-byte FIFO with: 64 byte Data Register and 64 byte Burst FIFO in the PCI bus interface mode.

11. DMA Controller

This is a 32-bit DMA Controller that features scatter gather function. This DMAC is a bus master during the PCI bus interface mode.

3.4. INTERNAL REGISTERS

The MB86605 internal registers are composed of: BASIC Control Register, MCS Buffer, SCAM Register, User Program Memory, and Initial Setting Register.

In the 16-Bit bus mode, those registers are accessed by window switching. In the PCI bus interface mode, all the registers except SCAM Register can be directly accessed (the window is released). Access to the SCAM register is provided via the window.

In the 16-bit bus mode, the address of registers that have 2bytes or more length (data block register) depends on the MPU type. In the 68 series mode (Mode 0), the register address is Big Endian orientation and in the 80 series mode (Mode 1), it is Little Endian orientation. In the PCI bus interface mode, the register address is Little Endian orientation.

When enabling the access of the memory space in the PCI bus interface mode, the PCI bus can directly access the user program memory. In addition, the 32-byte MCS buffer registers can be directly accessed without switching the bank.

3.4.1 Internal Registers at the 16-Bit Bus Mode

(1) Register Access

Mode 0 : 68-series mode

–CS0	–UDS	–LDS	D15 to D8	D7 to D0
H	—	—	HI-Z	HI-Z
L	L	L	Even-address register	Odd-address register
L	L	H	Even-address register	HI-Z
L	H	L	HI-Z	Odd-address register
L	H	H	HI-Z	HI-Z

Note: Connect the A0 pin to the High level, or leave it open.

Mode 1 : 80-series mode

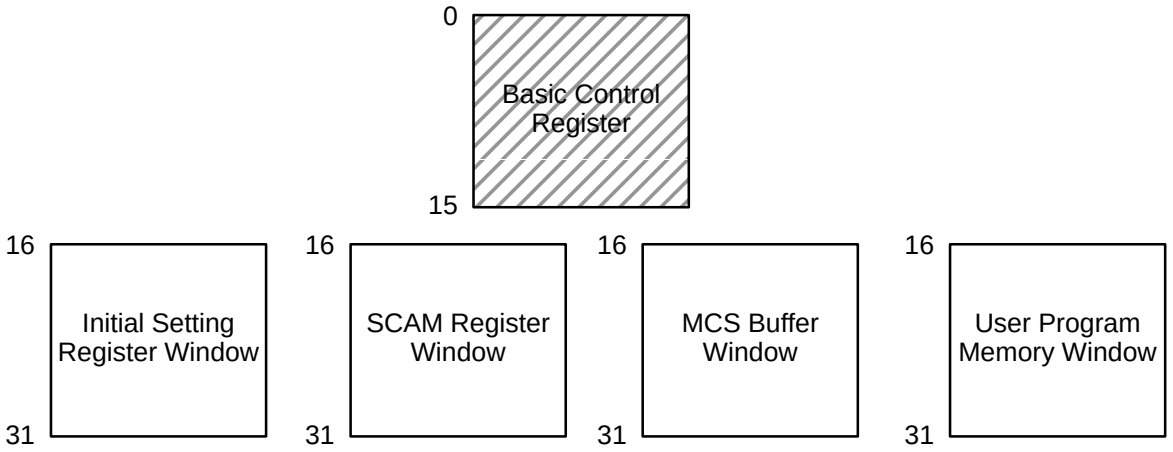
–CS0	–BHE	A0	D15 to D8	D7 to D0
H	—	—	HI-Z	HI-Z
L	L	L	Odd-address register	Even-address register
L	L	H	Odd-address register	HI-Z
L	H	L	HI-Z	Even-address register
L	H	H	HI-Z	HI-Z

(2) Register Allocation :

The following table is a register map in the 16-bit bus mode.

ADR	Register Window 00	Register Window 01	Register Window 10	Register Window 11
00h 0Fh	BASIC Control Register Window			
10h 1Fh	MCS Buffer Window	SCAM Register Window	User Program Memory Window	Initial Setting Register Window

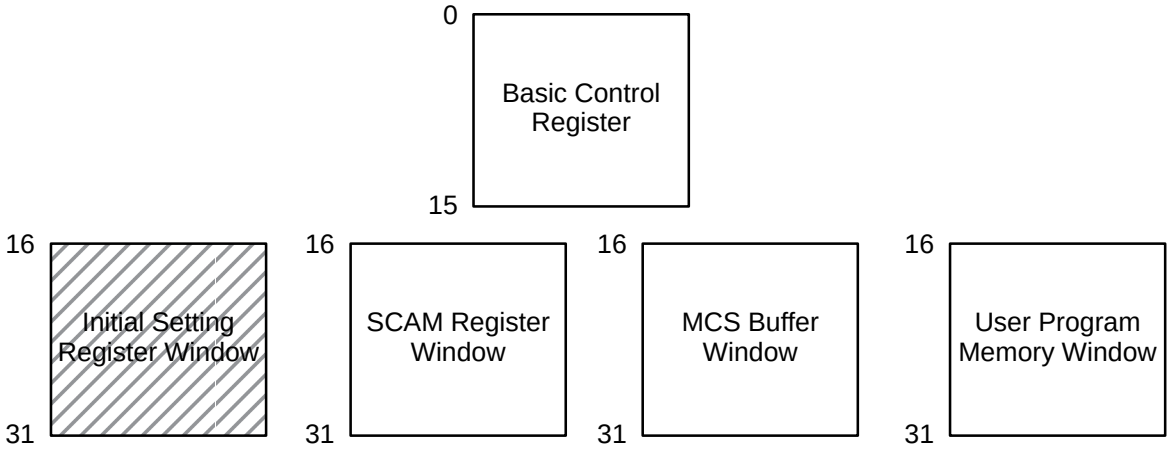
(3) BASIC Control Register



No		Address					When Write	When Read
MT	IN	A4	A3	A2	A1	A0		
00		0	0	0	0	0	SCSI output data register	SCSI input data register
01		0	0	0	0	1		
02		0	0	0	1	0	Direct control register	SPC status register
03		0	0	0	1	1	I/O control register	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	07	0	0	1	1	0	Data block register (MSB)	←
07	06	0	0	1	1	1	Data block register (LSB)	←
08	0A	0	1	0	0	0	Data byte register (MSB)	←
09		0	1	0	0	1	Data byte register (Mid)	←
0A	08	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

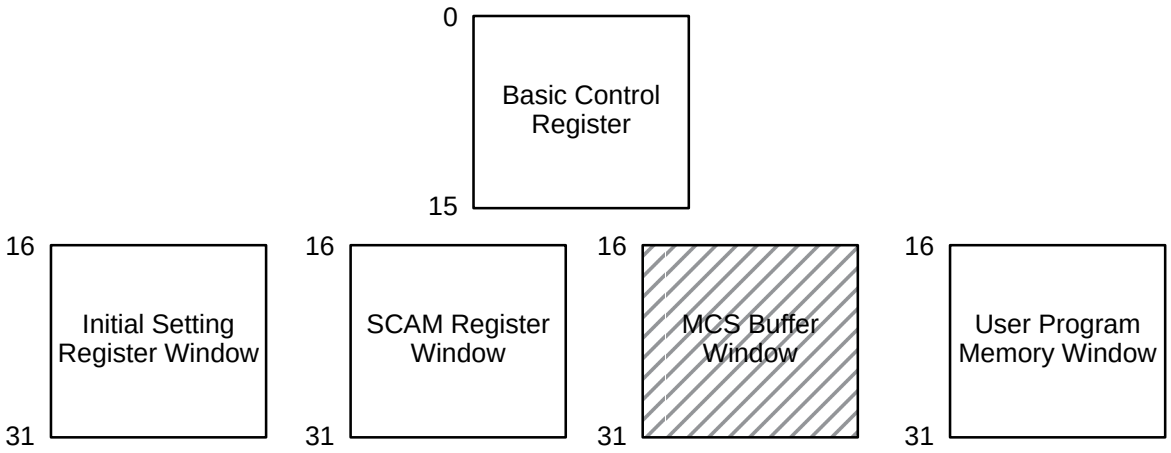
Note : MT : Address for the 68 series Interface Mode.
IN : Address for the 80 series Interface Mode.

(4) Initial Setting Register Window



No	Address					When Write	When 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 register	←
19	1	1	0	0	1	Interrupt enable setting register	←
1A	1	1	0	1	0	Group 6/7 command length setting register	←
1B	1	1	0	1	1	(reserved)	Responding own ID indication register
1C	1	1	1	0	0	Responding own ID setting register (MSB)	←
1D	1	1	1	0	1	Responding own ID setting register (LSB)	←
1E	1	1	1	1	0	Automatic operation mode setting register	←
1F	1	1	1	1	1	SPC timeout setting register	Revision indication register

(5) MCS Buffer Window



No	Address					When Write	When 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 is a 32-byte buffer to which the outgoing (send) messages, commands, and status to be sent are written.

Write the messages, commands, and status from address 0.

The SPC will send data from address 0.

When the command required to access the Send MCS buffer is issued, access to the Send MCS buffer is prohibited until the command is terminated.

– Receive MCS Buffer (Read)

This is a 32-byte buffer from which received messages, commands, and status are read.

The SPC will store the received from address 0. Read address 0 to receive MCS information.

Receive MCS Buffer access should be performed under the following conditions:

1. When the device is an Initiator :
Should be done after a completion of data receive command for Receive MCS buffer or after a completion of the automatic Reselection Response Operation until issuing “RESET ACK” command.

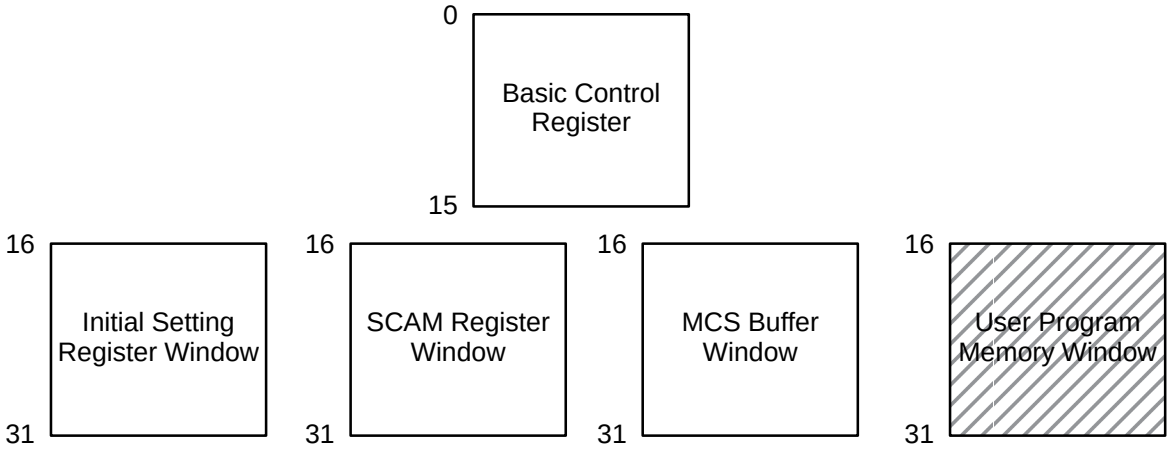
1. When the device is a Target :
Should be done after a completion of data receive command for Receive MCS buffer or after a completion of the automatic Selection Response Operation until a change of Phase.

When the command required to access the Receive MCS buffer is issued, access to the Receive MCS buffer is prohibited until the command is terminated.

When the message, command, and status are received, always read the Receive MCS buffer. The number of read bytes is given in the Modified Byte Register.

The next message, command, and status cannot be received unless all the data is read.

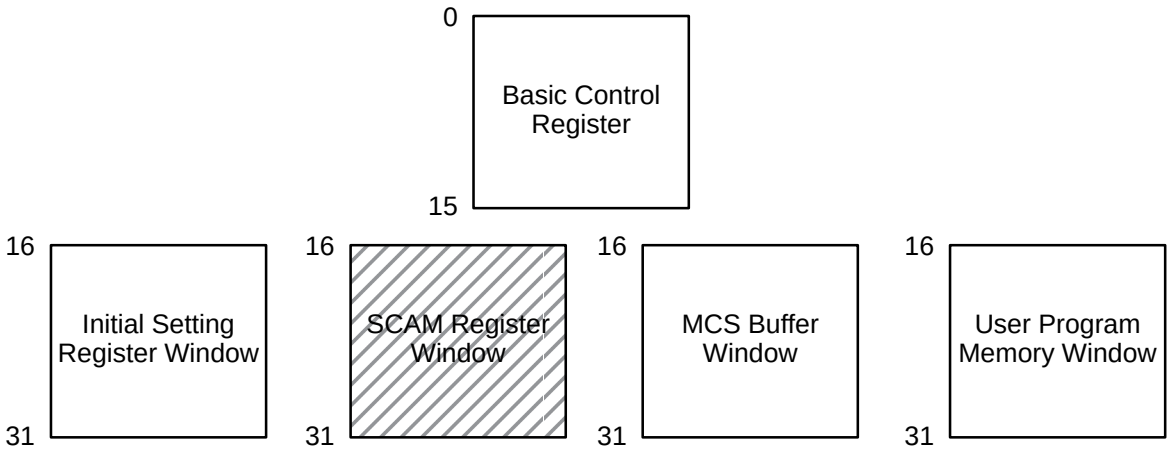
(6) User Program Memory Window



No		Address					When Write	When Read
MT	IN	A4	A3	A2	A1	A0		
10		1	0	0	0	0	User program memory control register	←
11		1	0	0	0	1	(reserved)	(reserved)
12	13	1	0	0	1	0/1	Transfer start address (MSB)	Current access pointer (MSB)
13	12	1	0	0	1	1/0	Transfer start address (LSB)	Current access pointer (LSB)
14		1	0	1	0	0	User program memory data port	←
15		1	0	1	0	1		
16		1	0	1	1	0	(reserved)	(reserved)
17		1	0	1	1	1	(reserved)	(reserved)
18	19	1	1	0	0	0/1	Program operation start address (MSB)	(reserved)
19	18	1	1	0	0	1/0	Program operation start address (LSB)	(reserved)
1A	1B	1	1	0	1	0/1	SEL/RESEL auto response start address (MSB)	(reserved)
1B	1A	1	1	0	1	1/0	SEL/RESEL auto response start address (LSB)	(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)

Note : MT : Address for the 68 series Interface Mode.
IN : Address for the 80 series Interface Mode.

(7) SCAM Register Window



No	Address					When Write	When 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)

3.4.2 Internal Registers at the PCI Bus Interface Mode

(1) Configuration Register

31 16	15 0	ADR
Device ID (=2001h)	Vendor ID (=10CFh)	00h
Status	Command	04h
Class Code (=010000h)	Revision ID	08h
——— Header Type	Latency Timer ———	0Ch
Base I/O Address		10h
Base Memory Address		14h
———		18h
———		1Ch
———		20h
———		24h
———		28h
———		2Ch
———		30h
———		34h
———		38h
Max_Lat Min_Gnt	Interrupt Pin Interrupt Line	3Ch

(2) Operation Registers (BASIC Control/Initial Setting/MCS Buffer/User Program Registers)

31 16	15 0			ADR
———	———	Data Register		00h
Data Transfer Address Register				04h
———	———	———	———	08h
———	———	———	———	0Ch
I/O Control / Nexus Status	Direct Control Reg / SPC Status	PCI Status Reg / PCI INT Enable Reg	PCI Modify Byte Reg / PCI Control Reg	10h
Data Block Register		Command Reg / Command Step Reg.	SEL/RESEL ID Reg. / Int. Status Reg.	14h
Diag / SCSI Control	Data Byte Register			18h
Modify Byte Register	Transfer Offset Reg.	Transfer Period Reg.	Transfer Mode Reg.	1Ch
SEL/RESEL Mode Reg.	Response Mode Reg.	Own ID Setting Reg.	Clock Conversion Reg.	20h
Async. Setup Reg.	REQ/ACK Timeout	SEL/RESEL Timeout	SEL/RESEL Retry	24h
SEL/RESELed ID	Group 6/7 Command	Interrupt Enable	Parity Error	28h
SPC Timeout / Rev.	Auto Mode Setting	Response Own ID		2Ch
User Program Data Transfer Address		———	User Program Memory Control	30h
User Program Memory Access Register				34h
SEL/RESEL Program Start Address		User Program Operation Start Address		38h
———	———	———	———	3Ch
MCS Buffer 03	MCS Buffer 02	MCS Buffer 01	MCS Buffer 00	40h
MCS Buffer 07	MCS Buffer 06	MCS Buffer 05	MCS Buffer 04	44h
MCS Buffer 0B	MCS Buffer 0A	MCS Buffer 09	MCS Buffer 08	48h
MCS Buffer 0F	MCS Buffer 0E	MCS Buffer 0D	MCS Buffer 0C	4Ch
MCS Buffer 13	MCS Buffer 12	MCS Buffer 11	MCS Buffer 10	50h
MCS Buffer 17	MCS Buffer 16	MCS Buffer 15	MCS Buffer 14	54h
MCS Buffer 1B	MCS Buffer 1A	MCS Buffer 19	MCS Buffer 18	58h
MCS Buffer 1F	MCS Buffer 1E	MCS Buffer 1D	MCS Buffer 1C	5Ch

3.4.3 List of Internal Registers

(1) BASIC Control Register

Bank	Address*		Register Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT / In	PCI										
XX	00h	00h	SCSI Data Register	R / W	SCSI First Data Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
XX	01h	01h			SCSI Second Data Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	02h	(Reserved)	–	X	X	X	X	X	X	X	X
–	–	03h	(Reserved)	–	X	X	X	X	X	X	X	X
–	–	04h	DMA Transfer Address Register	R / W	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
–	–	05h			DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8
–	–	06h			DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16
–	–	07h			DA31	DA30	DA29	DA28	DA27	DA26	DA25	DA24
–	–	08h	(Reserved)	–	X	X	X	X	X	X	X	X
–	–	09h	(Reserved)	–	X	X	X	X	X	X	X	X
–	–	0Ah	(Reserved)	–	X	X	X	X	X	X	X	X
–	–	0Bh	(Reserved)	–	X	X	X	X	X	X	X	X
–	–	0Ch	(Reserved)	–	X	X	X	X	X	X	X	X
–	–	0Dh	(Reserved)	–	X	X	X	X	X	X	X	X
–	–	0Eh	(Reserved)	–	X	X	X	X	X	X	X	X
–	–	0Fh	(Reserved)	–	X	X	X	X	X	X	X	X
–	–	10h	PCI Modified Byte Register	R	PCI Modified Byte Register PMB7	PMB6	PMB5	PMB4	PMB3	PMB2	PMB1	PMB0
			PCI Control Register	W	–	–	–	–	Burst Length BL1	BL0	Read multi	Read line
–	–	11h	PCI Status Register	R	detect PT err	Sig. Sys err	RX Mas. Abrt.	RX TRG. Abrt.	Sig TRG. Abrt.	Data Prty err	PI1	PI0
			PCI Interrupt Enable Register	W	DTPE Int. err	SSE INT Enbl	RMA Int. Enbl	RTA Int. Enbl	STA Int Enbl	Dape. int err	PO1	PO0
XX	02h	12h	SPC Status Register	R	INT	SPC Busy	Data tran. Rdy	SPC TMout	Ig. wide resi.	data tran. .req.	DREG full	DREG empty
			Direct Control Register	W	ATN CNTL	–	–	TMout clear	–	–	–	–

(1) BASIC Control Register (Continued)

Bank	Address*			Register Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI										
XX	03h		13h	Nexus Status Register	R	Init Mode	Targ. Mode	No padding	DMA misalin	Last Nexus ID LN3	LN2	LN1	LN0
				I/O Control Register	W	–	–	–	Manual Mis enbl	Auto Mis enbl	INT CNTL	SCSI DR CNTL DC1	DC0
XX	04h		14h	Interrupt Status Register	R	Interrupt Status IC7	IC6	IC5	IC4	IC3	IC2	IC1	IC0
				SEL/RESEL ID Register	W	–	–	–	–	SEL/RESEL ID SI3	SI2	SI1	SI0
XX	05h		15h	Command Step Register	R	Command #/Auto Sel Seq Code CS7	CS6	CS5	CS4	Command Step CS3	CS2	CS1	CS0
				Command Register	W	User pro. op.	Command Code CM6	CM5	CM4	CM3	CM2	CM1	CM0
XX	06h	07h	17h	Data Block Register	R / W	BL15	BL14	BL13	BL12	BL11	BL10	BL9	BL8
XX	07h	06h	16h			BL7	BL6	BL5	BL4	BL3	BL2	BL1	BL0
XX	08h	0Ah	1Ah	Data Byte/ MC Byte Register	R / W	BY23	BY22	BY21	BY20	BY19	BY18	BY17	BY16
XX	09h	09h	19h			BY15	BY14	BY13	BY12	BY11	BY10	BY9	BY8
XX	0Ah	08h	18h			BY7	BY6	BY5	BY4	BY3	BY2	BY1	BY0
XX	0Bh		1Bh	SCSI Control Signal Status Register	R	REQ	ACK	ATN	SEL	BSY	MSG	C/D	I/O
XX				Diagnostic Control Register	W	Diag REQ	Diag ACK	Diag ATN	'0'	Diag BSY	Diag MSG	Diag C/D	Diag I/O
XX	0Ch		1Ch	Transfer Mode Register	R/W	Sync trans.	Wide trans.	'0'	'0'	'0'	'0'	'0'	'0'
XX	0Dh		1Dh	Transfer Period Register	R/W	'0'	'0'	'0'	Transfer Period TP4	TP3	TP2	TP1	TP0
XX	0Eh		1Eh	Transfer Offset Register	R/W	'0'	'0'	'0'	Transfer Offset TO4	TO3	TO2	TO1	TO0
XX	0Fh		1Fh	Modified Byte Register	R	'0'	Modified Byte Register MB6	MB5	MB4	MB3	MB2	MB1	MB0
				Window Address Register	W	Window Select WS1	WS0	'0'	'0'	'0'	'0'	'0'	MCS window sel.

Note

MT : For 68 series MPU
In : For 80 series MPU

(2) Initial Setting Register

Bank	Address *			Register Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI										
11	10h		20h	Clock Conversion Setting Register	R/W	Clock Conversion							
						CC7	CC6	CC5	CC4	CC3	CC2	CC1	CC0
11	11h		21h	Own ID Setting Register	R/W	‘0’	‘0’	‘0’	‘0’	Own ID			
										OI3	OI2	OI1	OI0
11	12h		22h	Response Operation Mode Set Reg.	R/W	RESEL enbl	SEL enbl	Auto RX enbl	Parity Mode	‘0’	‘0’	Single init.	No arbt.
11	13h		23h	SEL/RESEL Operation Mode Set Reg.	R/W	SEL/ RESEL OP	Auto RESEL	Auto SEL	Targ. phase	‘0’	‘0’	‘0’	‘0’
11	14h		24h	SEL/RESEL Retry Setting Register	R/W	SEL/RESEL Retry				Retry Start Time			
						SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
11	15h		25h	SEL/RESEL Time out Setting Register	R/W	SEL/RESEL Timeout							
						ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0
11	16h		26h	REQ/ACK Timeout Setting Register	R/W	REQ/ACK Timeout							
						RT7	RT6	RT5	RT4	RT3	RT2	RT1	RT0
11	17h		27h	Async. Setup Time Setting Register	R/W	‘0’	‘0’	‘0’	‘0’	Async Setup Time			
										AT3	AT2	AT1	AT0
11	18h		28h	Parity Err Detection Setting Register	R/W	SCSI P–chk	SCSI P–gen	Sys. odd/even	MPU P–chk	MPU P–gen	‘0’	DMA P–chk	DMA P–gen
11	19h		29h	Interrupt Enable Setting Register	R/W	All int. enbl	–	Auto S/R enble	SEL/ RESEL enble	Report enbl	Phase err enbl	Trans. err enbl	SCSI res enbl
11	1Ah		2Ah	Group 6/7 Command Setting Register	R/W	Group 7 Command Length				Group 6 Command Length			
						GL7	GL6	GL5	GL4	GL3	GL2	GL1	GL0
11	1Bh		2Bh	Response Own ID Indication Register	R	‘0’	‘0’	‘0’	‘0’	Response Own ID			
										RO3	RO2	RO1	RO0
11	1Ch		2Ch	Response Own ID Setting Register	R/W	Response ID							
						ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
11	1Dh		2Dh		R/W	Response ID							
						ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
11	1Eh		2Eh	Auto Operation Mode Set Register	R/W	Com. ignore	Int. Mode	CR1	CR0	REQ ast mask	C. comp. mask	L2 init mask	Auto ACK res
11	1Fh		2Fh	Revision Indication Register	R	Revision Code							
						RV7	RV6	RV5	RV4	RV3	RV2	RV1	RV0
11				SPC Timeout Setting Register	W	SPC Timeout Timer							
						TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0

Note

MT : For 68 series MPU
In : For 80 series MPU

(3) MCS Buffer Register

MCS Window Sel.	Bank	Address *			Register Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		MT	In	PCI										
0	00	10h		40h	Receive MCS Buffer 00h	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
					Send MCS Buffer 00h	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	00	11h		41h	Receive MCS Buffer 01h	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
					Send MCS Buffer 01h	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	00	12h		42h	Receive MCS Buffer 02h	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
					Send MCS Buffer 02h	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	00	13h		43h	Receive MCS Buffer 03h	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
					Send MCS Buffer 03h	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	00	14h		44h	Receive MCS Buffer 04h	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
					Send MCS Buffer 04h	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	00	15h		45h	Receive MCS Buffer 05h	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
					Send MCS Buffer 05h	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	00	16h		46h	Receive MCS Buffer 06h	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
					Send MCS Buffer 06h	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	00	17h		47h	Receive MCS Buffer 07h	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
					Send MCS Buffer 07h	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	00	18h		48h	Receive MCS Buffer 08h	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
					Send MCS Buffer 08h	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	00	19h		49h	Receive MCS Buffer 09h	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
					Send MCS Buffer 09h	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	00	1Ah		4Ah	Receive MCS Buffer 0Ah	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
					Send MCS Buffer 0Ah	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	00	1Bh		4Bh	Receive MCS Buffer 0Bh	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
					Send MCS Buffer 0Bh	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	00	1Ch		4Ch	Receive MCS Buffer 0Ch	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
					Send MCS Buffer 0Ch	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	00	1Dh		4Dh	Receive MCS Buffer 0Dh	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
					Send MCS Buffer 0Dh	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	00	1Eh		4Eh	Receive MCS Buffer 0Eh	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
					Send MCS Buffer 0Eh	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	00	1Fh		4Fh	Receive MCS Buffer 0Fh	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
					Send MCS Buffer 0Fh	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

* : MT : For 68 series MPU
In : For 80 series MPU

(3) MCS Buffer Register **(Continued)**

MCS Window Sel.	Bank	Address *		Register Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		MT / In	PCI										
1	00	10h	50h	Receive MCS Buffer 10h	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Send MCS Buffer 10h	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	00	11h	51h	Receive MCS Buffer 11h	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Send MCS Buffer 11h	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	00	12h	52h	Receive MCS Buffer 12h	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Send MCS Buffer 12h	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	00	13h	53h	Receive MCS Buffer 13h	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Send MCS Buffer 13h	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	00	14h	54h	Receive MCS Buffer 14h	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Send MCS Buffer 14h	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	00	15h	55h	Receive MCS Buffer 15h	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Send MCS Buffer 15h	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	00	16h	56h	Receive MCS Buffer 16h	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Send MCS Buffer 16h	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	00	17h	57h	Receive MCS Buffer 17h	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Send MCS Buffer 17h	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	00	18h	58h	Receive MCS Buffer 18h	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Send MCS Buffer 18h	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	00	19h	59h	Receive MCS Buffer 19h	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Send MCS Buffer 19h	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	00	1Ah	5Ah	Receive MCS Buffer 1Ah	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Send MCS Buffer 1Ah	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	00	1Bh	5Bh	Receive MCS Buffer 1Bh	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Send MCS Buffer 1Bh	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	00	1Ch	5Ch	Receive MCS Buffer 1Ch	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Send MCS Buffer 1Ch	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	00	1Dh	5Dh	Receive MCS Buffer 1Dh	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Send MCS Buffer 1Dh	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	00	1Eh	5Eh	Receive MCS Buffer 1Eh	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Send MCS Buffer 1Eh	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	00	1Fh	5Fh	Receive MCS Buffer 1Fh	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Send MCS Buffer 1Fh	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

* : MT : For 68 series MPU
In : For 80 series MPU

(4) User Program Memory Register

Bank	Address *			Register Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI										
10	10h		30h	User Program Memory Control Register	R/W	'0'	'0'	'0'	'0'	'0'	'0'	Bank Mode	Bank Sel.
10	11h		31h	(Reserved)	–	X	X	X	X	X	X	X	X
10	12h	13h	33h	Current Access Pointer	R	'0'	'0'	'0'	'0'	'0'	CA10	CA9	CA8
				Transfer Start Address	W						TS10	TS9	TS8
10	13h	12h	32h	Current Access Pointer	R	CA7	CA6	CA5	CA4	CA3	CA2	CA1	'0'
				Transfer Start Address	W	TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0
10	14h		34h	User Program Memory Data Port	R/W	User Program First Data							
						Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
10	15h		35h			User Program Second Data							
						Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–		36h			User Program Third Data							
						Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–		37h			User Program Fourth Data							
						Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
10	18h	19h	39h	User Program Operation Start Address Register	R/W	'0'	'0'	'0'	'0'	'0'	UA10	UA9	UA 8
10	19h	18h	38h			UA7	UA6	UA5	UA4	UA3	UA2	UA1	UA0
10	1Ah	1Bh	3Bh	SEL/RESEL Auto Response Start Address Register	R/W	'0'	'0'	'0'	'0'	'0'	SA10	SA9	SA8
10	1Bh	1Ah	3Ah			SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0
10	1Ch		3Ch	(Reserved)	–	X	X	X	X	X	X	X	X
10	1Dh		3Dh	(Reserved)	–	X	X	X	X	X	X	X	X
10	1Eh		3Eh	(Reserved)	–	X	X	X	X	X	X	X	X
10	1Fh		3Fh	(Reserved)	–	X	X	X	X	X	X	X	X

* : MT : For 68 series MPU
In : For 80 series MPU

(5) SCAM Register

Bank	Address *			Register Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI										
01	10h		20h	SCAM Data Bus Register	R / W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
01	11h		21h	SCAM Control Register	R / W	'0'	'0'	'0'	SEL	BSY	MSG	C/D	I/O
01	12h		22h	(Reserved)	–	X	X	X	X	X	X	X	X
01	13h		23h	(Reserved)	–	X	X	X	X	X	X	X	X
01	14h		24h	(Reserved)	–	X	X	X	X	X	X	X	X
01	15h		25h	(Reserved)	–	X	X	X	X	X	X	X	X
01	16h		26h	(Reserved)	–	X	X	X	X	X	X	X	X
01	17h		27h	(Reserved)	–	X	X	X	X	X	X	X	X
01	18h		28h	(Reserved)	–	X	X	X	X	X	X	X	X
01	19h		29h	(Reserved)	–	X	X	X	X	X	X	X	X
01	1Ah		2Ah	(Reserved)	–	X	X	X	X	X	X	X	X
01	1Bh		2Bh	(Reserved)	–	X	X	X	X	X	X	X	X
01	1Ch		2Ch	(Reserved)	–	X	X	X	X	X	X	X	X
01	1Dh		2Dh	(Reserved)	–	X	X	X	X	X	X	X	X
01	1Eh		2Eh	(Reserved)	–	X	X	X	X	X	X	X	X
01	1Fh		2Fh	(Reserved)	–	X	X	X	X	X	X	X	X

* : MT : For 68 series MPU
In : For 80 series MPU

3.5. PCI Bus Interface Function

MB86605 has a PCI Bus Interface conforming to the PCI LOCAL BUS SPECIFICATION Revision 2.0 enabling the device to be directly connected to the PCI bus. In addition, the MB86605 may be used in conjunction with traditional MPU and DMA Interfaces.

(1) How to Set the PCI Bus Interface Mode

The operation mode is set via pins MODE 1 and MODE 0. By setting "1" to both MODE 1 and MODE 0 pins, the interface is set to PCI Bus Mode.

(2) PCI Bus Commands

The MB86605 supports the PCI bus commands as listed in the table below.

Command Code	Command Type	Master Operation	Target Operation
0000	Special INT ACK	X	X
0001	Special cycle	X	X
0010	I/O read	X	O *1
0011	I/O write	X	O *1
0100	–	–	–
0101	–	–	–
0110	Memory read	O	O *1
0111	Memory write	O	O *1
1000	–	–	–
1001	–	–	–
1010	Config. read	X	O
1011	Config. write	X	O
1100	Memory read multi	O *2	O *3
1101	Dual ADR cycle	X	X
1110	Memory read line	O *2	O *3
1111	Memory write & Invalidate	X	O *4

O : Supported
X : Not supported

*1 : Register/memory access area is specified by the bits 1and 0 of the PCI configuration register.
*2 : Can be issued instead of the Memory read command. (It can be selected by the PCI read command control register.)
*3 : Recognizes it as the Memory read command.
*4 : Recognizes it as the Memory write command.

3.5.1 PCI Configuration Registers

(1) Vendor ID (Address 00h and 01h : Read only)

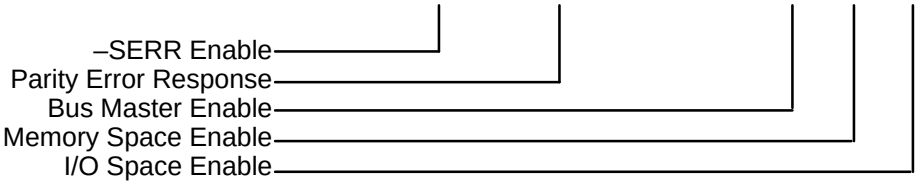
10CFh are read out from these addresses.

(2) Device ID (Address 02h and 03h : Read only)

2001h are read out from these addresses.

(3) Command Register (Address 04h and 05h : R/W)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0		0		0	0	0			
Write	X	X	X	X	X	X	X		X		X	X	X			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



The following table lists the each bit function.

Bit #	Bit Name	R/W	Value	Function
15–9	Unused	Read	0	Read data is always 0.
8	–SERR Enable	Read/ Write	0	Does not output –SERR signal.
			1	Outputs the –SERR signal. The –SERR signal is asserted only when the address parity error is detected during the target operation.
7	Unused	Read	0	Read data is always 0.
6	Parity Error Response	Read/ Write	0	Does not detect the parity error.
			1	Detects the address and data parity error.
5,4,3	Unused	Read	0	Read data is always 0.
2	Bus Master Enable	Read/ Write	0	Inhibits the PCI Bus master operation.
			1	Makes the PCI Bus master operation enable.

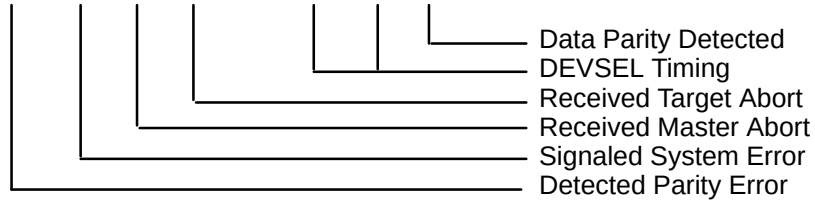
Continued from the previous page.

<table>
<tr><th>Bit #</th><th>Bit Name</th><th>R/W</th><th>Function</th></tr>
<tr><td>1

0</td><td>Memory and I/O Space Enable</td><td>Read/ Write</td><td>
<table>
<tr><th>Bit 1</th><th>Bit 0</th><th>User program memory access</th><th>Register access</th></tr>
<tr><td>0</td><td>0</td><td>Disable</td><td>Disable</td></tr>
<tr><td>0</td><td>1</td><td>Disable</td><td>Enable (I/O command)</td></tr>
<tr><td>1</td><td>0</td><td>Enable</td><td>Enable (Memory command)</td></tr>
<tr><td>1</td><td>1</td><td>Enable</td><td>Enable (I/O command)</td></tr>
</table>
</td></tr>
</table>

(4) Status Register (Address 06h and 07h : R/W)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read					0	0	1		0	0	0	0	0	0	0	0
Write	1R	1R	1R	1R	X	X	X	1R	X	X	X	X	X	X	X	X
Initial Value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0



The following table lists the each bit function.

Bit #	Bit Name	R/W	Value	Function
15	Detected Parity Error	Read	0	No Data Parity Error.
			1	Detects the data parity error. This bit is set when the data parity error is detected regardless of the value for the bit6 of the Control Register (Parity Error Response).
		Write	0	No operation. (Does not influence.)
			1	Clears this bit.
14	Signaled System Error	Read	0	No Address Parity Error.
			1	Detects the address parity error and asserted –SERR.
		Write	0	No operation. (Does not influence.)
			1	Clears this bit.
13	Received Master Abort	Read	0	No master abortion.
			1	Pauses transfer by the master–abort
		Write	0	No operation. (Does not influence.)
			1	Clears this bit.

Continued from the previous page.

Bit #	Bit Name	R/W	Value	Function
12	Received Target Abort	Read	0	No Target abortion.
			1	Pauses transfer by the target-abort
		Write	0	No operation. (Does not influence.)
			1	Clears this bit.
11	Unused	Read	0	0 is always read out.
10, 9	DEVSEL Timing	Read	01	01h is always read out (for medium).
8	Data Parity Detected	Read	0	No Data Parity Error.
			1	Detects the data parity error. However, this bit is set in case that the –PERR is asserted when the bit6 of the control register =1 (Parity Error Response Bit) during the master operation. or the master detects that the –PERR is asserted by a Target.
		Write	0	No operation. (Does not influence.)
			1	Clears this bit.
7–0	Unused	Read	0	0 is always read out.

(5) Revision ID Register (Address : 08h ; Read only)

The same value as the Revision register of the Initial Setting Registers is read out.

(6) Class Code (Address : 09h,0Ah, and 0Bh : Read only)

Class code = 010000h.

(7) Cache Line Size (Address : 0Ch)

This is not supported by MB86605. Initial value is 00h, and can not be written to this register.

(8) Latency Timer (Address : 0Dh, R/W)

The initial value is 00h. During the master operation, the counter is decremented by a rising edge of the PCLK after the assertion of the –FRAME signal. The SPC aborts the bus after the Timeout (with the master-abort). The upper 5 bits can be written, but the lower 3 bits are set to 0 and cannot be written. This register does not function if 00 is written. (A value must be set to this register or the PCI specification will be violated.)

(9) Header Type (Address : 0Eh, Read only)

Header type = 00h.

(10) BIST (Built-in Self Test) (Address : 0Fh, R/W)

This is not supported by MB86605. Initial value is 00h, and can not be written.

(11) Base Address (Address 10h to 27h : R/W)

– Base I/O Address : this is a register to set the base address of the MB86605 internal register. The address of the internal registers is allocated from this base address.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Write																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	*	*	*	*	*	*	*	*	0	0	0	0	0	0	0	1
Write									X	X	X	X	X	X	X	X
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

1) FFFFFF01h is read out after writing FFFFFFFFh (all “1”) to this register. (Maximum I/O space is 256-byte.)

2) The written value can be read out after writing the I/O address.

3) The I/O address can be set in 256-byte boundary.

4) The actual internal registers are 96 bytes and they are allocated in the addresses 00h to 5Fh. Addresses 60h to FFh are unused.

– Base Memory Address : If a direct memory access is performed to the user program memory, set the base address of the user program memory to be allocated. Also, the user program memory can perform the I/O access through the user program memory data port in addition to the memory access.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Write																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	*	*	*	*	0	0	0	0	0	0	0	0	0	0	0	0
Write					X	X	X	X	X	X	X	X	X	X	X	X
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

1) FFFFF000h is read out after writing FFFFFFFFh (all “1”) to this register. (A maximum of memory space is 4096-byte.)

2) The written value can be read out after writing the memory address.

3) The memory address can be set in 4K-byte boundary.

4) The actual user program memory is 2Kbytes and it is allocated in the addresses 800h to FFFh. The addresses 000h to 7FFh are unused. (When the internal registers are assigned in the memory space, the addresses 000h to 05Fh are allocated for the internal registers.)

– The addresses 18h to 27h are unused and they can not be modified. The read out value is all 0.

(12) Expansion ROM Base Address (address 30h to 33h)

They are unused spaces and can not be written. The read out value is all 0.

(13) Interrupt Line (Address 3Ch, R/W)

This is an 8-bit register to set the interrupt line to be used. (It can be read and written.)

(14) Interrupt Pin (Address 3Dh, Read only)

The read value is 01h.

(15) Min_Gnt (Address 3Eh, Read only)

The read value is 00h.

(16) Max_Lat (Address 3Fh, Read only)

The read value is 00h.

3.6. FUNCTIONAL DESCRIPTIONS OF INTERNAL REGISTERS

3.6.1. BASIC Control Register

The BASIC Control Registers are initialized by the external reset or the issue of a software reset command. Except the Bits 2, 1, and 0 of I/O control register and Window address register that can be initialized by only the external reset. (The register keeps the previous values during Software reset, and rewrite is possible during the software reset. But, the Command Register accepts only the software reset command during the software reset.)

(1) SCSI Data Register :

Bank	Address*			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
XX	00h		00h	R / W	SCSI First Data							
					Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
XX	01h		01h		SCSI Second Data							
					Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

– This register is used to input and output the data on the SCSI bus at the program transfer mode in the data phase.

– The written data are output on the SCSI bus in order of the register address.

– The data input from the SCSI bus can be read from this register in order of the address.

– The initial value for this register is undefined.

– If the data transfer is performed via this register, it is first required that the Data Trans REQ bit (Bit 2) of SPC Status Register be polled. Also, in order to maintain the data transfer rate of the system side, the data transfer should be performed via the DMA transfer (via the bus master operation in case of the PCI bus interface mode).

(2) DMA Transfer Address Register : (Valid only for the PCI bus interface mode.)

This register is used exclusively for the DMA data transfer specifically when issuing the commands during PCI bus interface mode. The register setting is not necessary when DMA transfer is performed in the user program operation. The DMA transfer addresses should be held in the table in the user program memory.

Bank	Address*			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
–	–		04h	R / W	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
–	–		05h		DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8
–	–		06h		DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16
–	–		07h		DA31	DA30	DA29	DA28	DA27	DA26	DA25	DA24

– This register is used to set the destination address for the data to be transferred in the bus master operation of the PCI bus interface mode.

– The address can be set in byte unit. In the first transfer, the SPC performs the data transfer for the long word misallignment. In the later transfers (2nd transfer, 3rd transfer...), data is transfered in long word units.

– If this register is read during the data transfer, the address that will be accessed next can be read.

– The initial value is undefined.

(3) PCI Modified Byte Register (R)/PCI Control Register(W) : (Valid only for PCI bus interface mode)

Bank	Address*			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
–	–		10h	R	PCI Modified Byte Register							
					PMB7	PMB6	PMB5	PMB4	PMB3	PMB2	PMB1	PMB0
				W					Burst Length		Read multi	Read line
					‘0’	‘0’	‘0’	‘0’	BL1	BL0		
Initial Value					‘0’	‘0’	‘0’	‘0’	‘0’	‘0’	‘0’	‘0’

– In the bus master operation of the PCI bus interface mode this register indicates the data byte number transferred to the system side. (Initial value=00h.)

– When the data transfer instruction is executed, the lower 8-bits of data in the Data Byte Register is copied to this register and the data transfer is performed. The SPC counts down the number every byte in this register.

– The MB86605 incorporates a 64-byte burst transfer buffer (FIFO) in the PCI bus interface block. When the data transfer command is completed holding the data in the buffer (abnormal termination occurs), the number of byte data remained in the buffer can be calculated with the values for this register, Modified Byte Register, and Data Byte Register. As for the calculation of the number of data bytes remaining, see Exit Processing section in this manual.

– The following is descriptions of each bit.

Bit	Bit Name	R/W	Value	Function
7 to 0	PMB7 to PMB0	Read	–	PCI Modified Byte Counter.
7 to 4	Unused	Write	0	Always write ‘0’ for the write operation.
3 2	Burst Length 1 Burst Length 2	Write	–	These bits are used to set the burst length of the data transfer in the master operation as listed in the table below.

BL1	BL0	Burst Length
0	0	1 long word
0	1	4 long words
1	0	8 long words
1	1	16 long words

<table>
<tr><th>Bit</th><th>Bit Name</th><th>R/W</th><th>Value</th><th>Function</th></tr>
<tr><td>1
0</td><td>Read Multi
Read Line</td><td>Write</td><td>–</td><td>These bits are used to select the commnad to perform the memory read operation in the master operation. The comand is selected as listed in the table below.
<table>
<tr><th>Read Multi</th><th>Read Line</th><th>Command</th></tr>
<tr><td>0</td><td>0</td><td>Memory read command</td></tr>
<tr><td>0</td><td>1</td><td>Memory read line command</td></tr>
<tr><td>1</td><td>0</td><td>Memory read multiple command</td></tr>
<tr><td>1</td><td>1</td><td>(reserved)</td></tr>
</table></td></tr>
</table>

(4) PCI Status Register(R)/PCI Interrupt Enable Register(W) : (Valid for only PCI bus interface mode)

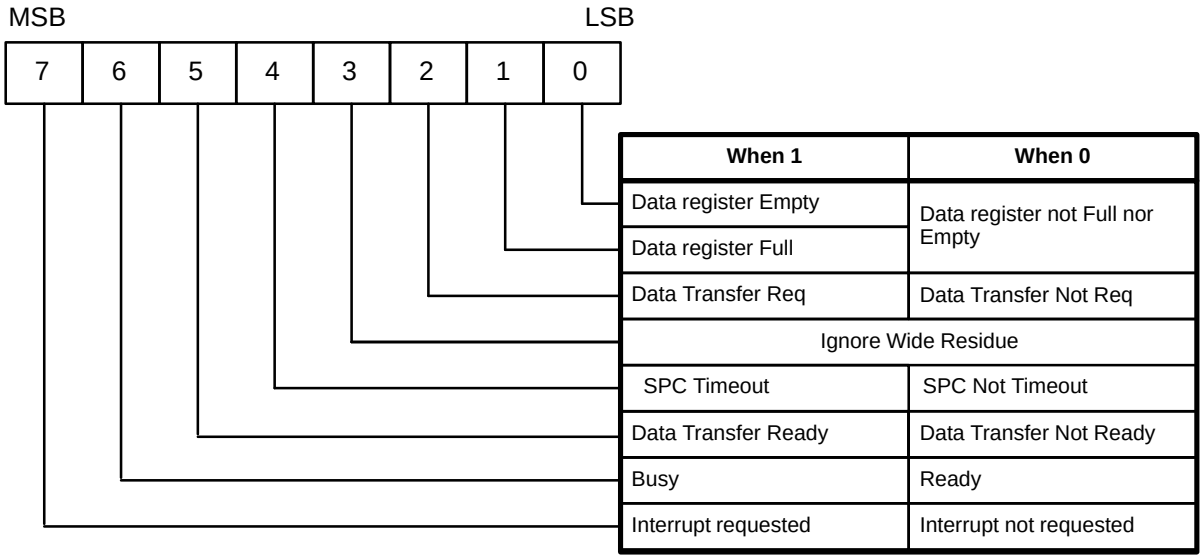
Bank	Address*			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
–	–		11h	R	Detect PT Err	SiG Sys Err	RX Mas Abrt	RX TRG Abrt	Sig TRG Abrt	Data PT Err	PI1	PI0
				W	DTPE Int enbl	SSE INT enbl	RMA Int enble	RTA int enbl	STA int enbl	DAPE Int err	PO1	PO0
Initial Value					‘0’	‘0’	‘0’	‘0’	‘0’	‘0’	‘0’	‘0’

Bit	Bit Name	R/W	Value	Function
7	Detect PT Err	Read	–	The value for the bit 15 of the Status register in the Configuration registers (Detected Parity Error) can be read from this bit.
	DTPE Int Enbl	Write	0	A Detect PT Err interrupt is disabled to generate.
			1	An interrupt pin –IRQ is asserted by setting the Detect PT Err bit.
6	Sig Sys Err	Read	–	The value for the bit 14 of the Status register in the Configuration registers (Signaled System Error) can be read from this bit.
	SSE Int Enbl	Write	0	A Sig Sys Err interrupt is disabled to generate.
			1	An interrupt pin –IRQ is asserted by setting the Sig Sys Err bit.
5	RX Mas Abrt	Read	–	The value for the bit 13 of the Status register in the Configuration registers (Received Master Abort) can be read from this bit.
	RMA Int Enbl	Write	0	An RX Mas Abrt interrupt is disabled to generate.
			1	An interrupt pin –IRQ is asserted by setting the RX Mas Abrt bit.
4	RX Targ Abrt	Read	–	The value for the bit 12 of the Status register in the Configuration registers (Received Target Abort) can be read from this bit.
	RTA Int Enbl	Write	0	An RX Targ Abrt interrupt is disabled to generate.
			1	An interrupt pin –IRQ is asserted by setting the RX Targ Abrt bit.

Bit	Bit Name	R/W	Value	Function
3	Sig Targ Abrt	Read	–	The value for the bit 11 of the Status register in the Configuration registers (Signaled Target Abort) can be read from this bit.
	STA Int Enbl	Write	0	A Sig Targ Abrt interrupt is disabled to generate.
			1	An interrupt pin –IRQ is asserted by setting the Sig Targ Abrt bit.
2	Data Prty Err	Read	–	The value for the bit 8 of the Status register in the Configuration registers (Data Parity Detected) can be read from this bit.
	DAPE Int Err	Write	0	A Data Prty Err interrupt is disabled to generate.
			1	An interrupt pin –IRQ is asserted by setting the Data Prty Err bit.
1 0	PI1 PI0	Read	–	The values for external PI1 and PI0 pins can be read.
1 0	PO1 PO0	Write	–	The values set to those bits are output from the external PO1 and PO0 pins.

(5) SPC Status Register(R)/Direct Control Register(W)

Bank	Address*			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
XX	02h		12h	R	INT	SPC Busy	Data tran Rdy	SPC TMout	Ig. wide resi.	Data tran. .req.	DREG full	DREG empty
				W	ATN CNTL	'0'	'0'	TMout clear	'0'	'0'	'0'	'0'
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'1'



When 1	When 0
Data register Empty	Data register not Full nor Empty
Data register Full	
Data Transfer Req	Data Transfer Not Req
Ignore Wide Residue	
SPC Timeout	SPC Not Timeout
Data Transfer Ready	Data Transfer Not Ready
Busy	Ready
Interrupt requested	Interrupt not requested

– SPC Status Register (read) :

Bit 7: Interrupt requested

Bit 7 indicates 1 when an interrupt request is made from the SPC to the host MPU (when interrupt code is held in the Interrupt Status Register).

Although an interrupt request is interlocked with the INT signal (interrupt request signal) to the host MPU, this bit indicates 1 when an interrupt request is made irrespective of whether an interrupt is enabled or disabled (Enable/Disable: interrupt enable register).
This bit never becomes '1' while the bit 6 (SPC busy) = '1'. After the completion of the operation and the bit 6='0', '1' is set to this bit (only when an interrupt occurs).

Bit 6: Busy

Bit 6 indicates that the SPC is busy.

This bit indicates 1 when a command is received or the operation is performed automatically (in automatic selection/reselection response mode).

This bit indicates 0 when the operation is terminated normally or abnormally (when an error occurs).

If a command is issued when this bit is 1, any command other than the **SOFTWARE RESET** command will be ignored, and the COMMAND REJECTED interrupt is reported.

Bit 5: Data Transfer Ready

Bit 5 indicates that data transfer is ready (in progress).

This bit indicates 1 when the command related to transfer is received to complete internal setup.

Program transfer must be performed after checking this bit.

Bit 4: SPC Timeout

Bit 4 indicates the SPC timeout. When the SPC operates for longer than the time specified in the SPC Timeout Setting register (1Fh) of the Initial Setting Register, '1' is set to this bit. When the SPC completes the operation within the time as specified in the Timeout setting register, this bit holds '0'.

Bit 3: Ignore Wide Residue

In the data phase, bit 3 indicates whether or not the final byte data to be transferred is valid in wide data transfer mode. This means, when the transferred byte is an odd number, '1' is set to this bit after the completion of data transfer. Even if two or more commands are issued to transfer data in the data phase, the numbers of bytes transferred are summed up and the result is reported by this bit.

Note that the value which is read during the data transfer is invalid since this bit changes during the transfer.

Bit 2: Data Transfer Req

Bit 2 requests transfer in the data phase.

In the SCSI-INPUT mode, this bit indicates 1 when two bytes or more can be read from the data register (when there is data of two bytes or more in the data register) or when the last byte is in the data register.

In the SCSI-OUTPUT mode, this bit indicates 1 when data of two bytes or more can be written to the data register (when there is data of 62 bytes or less in the data register) or when the last byte can be written to the data register.
In DMA transfer mode, this bit indicates 1 while the DREQ signal is asserting.

Program transfer should be performed after referencing the status of this bit.

Bit 1: Data register Full

Bit 1 indicates 1 when the data register is Full.

Bit 0: Data register Empty

Bit 0 indicates 1 when the data register is Empty.

– Direct Control Register (write) :

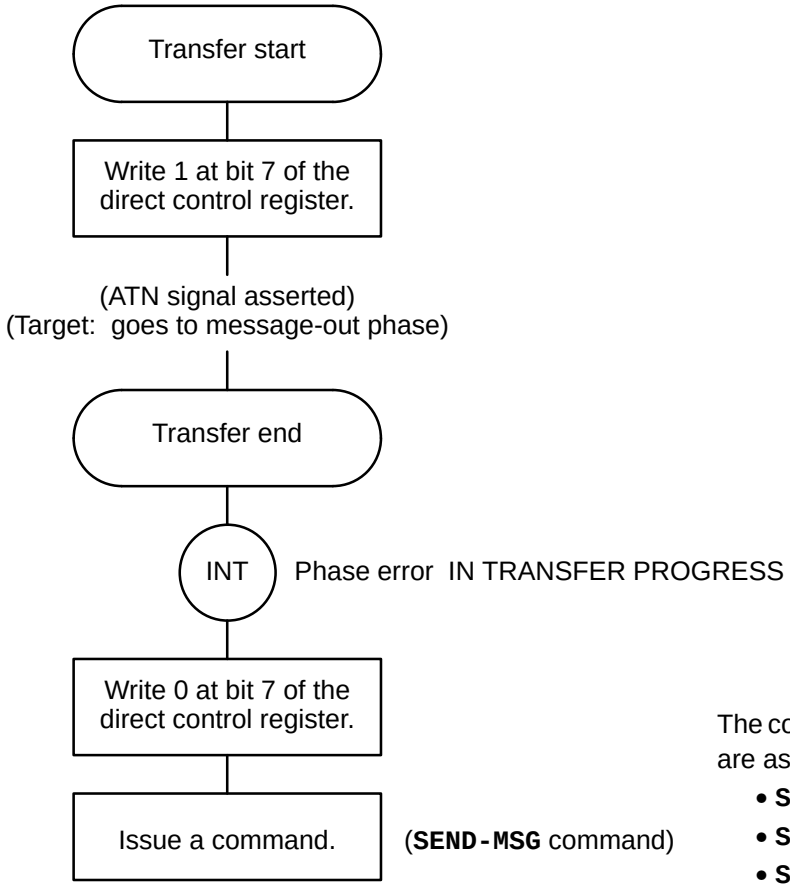
Bit 7: ATN signal control

Bit 7 directly controls assertion of the ATN signal.

When the SPC is operating as the initiator, writing 1 at bit 7 enables assertion of the ATN signal.
This bit is valid for stopping the data phase halfway (generation of attention condition).

If the ATN signal is asserted by this bit, always write 0 to reset this bit before issuing the next command.

If the ATN signal is asserted by this bit, it will not be negated even by writing 0. To negate the ATN signal, issue the RESET ATN command.



The commands related to SEND-MSG are as follows:

- **Send 1-MSG**
- **Send 1-MSG with ATN**
- **Send N-Byte-MSG**

Bit 4: SPC Timeout Clear

If '1' is written at this location while SPC Busy bit (Bit 6) = '1', the SPC Timeout flag (Bit 4) of SPC Status Register can be cleared.

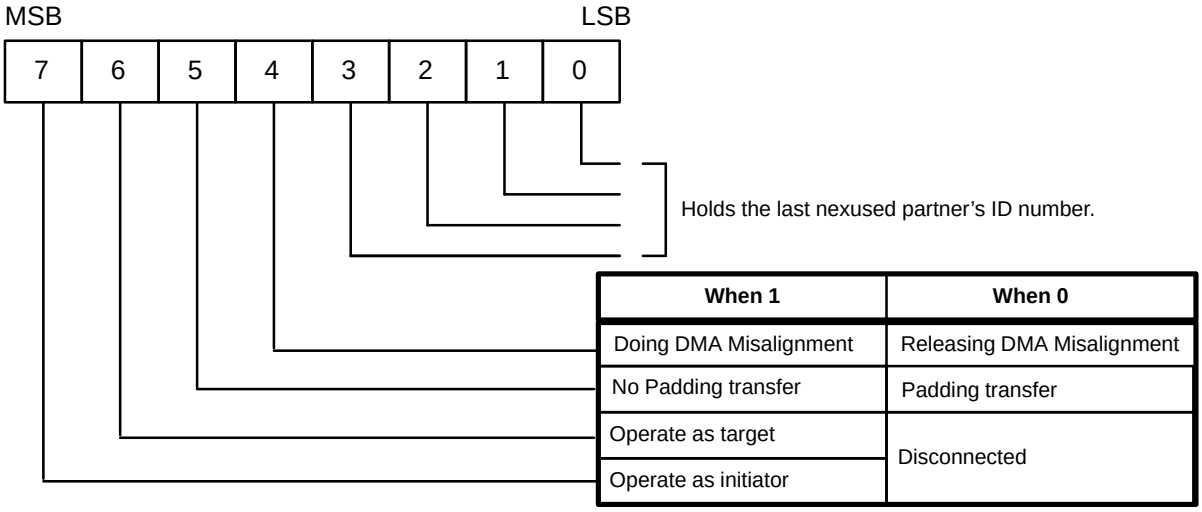
Bits 6,5,3,2,1,0 : Unused

Unused bits. Write '0' at these bits when writing at this Register.

(6) Nexus Status Register(R) / I/O Control Register(W)

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
XX	03h		13h	R	INIT MODE	TARG MODE	No Padding	DMA Misalin	Last Nexus ID LN3	LN2	LN1	LN0
				W	–	–	–	Manual Mis enbl	Auto Mis enbl	INT CNTL	SCSI DR CNTL DC1	DC0
Initial Value					‘0’	‘0’	‘0’	‘0’	‘0’	‘0’	‘0’	‘0’

– Nexus Status Register (read) :



When 1	When 0
Doing DMA Misalignment	Releasing DMA Misalignment
No Padding transfer	Padding transfer
Operate as target	Disconnected
Operate as initiator	

Bit 7: Operation as initiator

Bit 7 indicates 1 when the SPC starts operation as the initiator.

- When SPC reselected by target (when SEL and I/O signals true and own ID bit is true)
- When SPC, as initiator, acquires right to use the bus to send ID bit on SCSI bus in selection phase after asserting SEL signal or simultaneously asserts ATN signal

This bit indicates ‘0’ when the SPC is disconnected by the target to clear the I-T nexus.

Bit 6: Operation as target

Bit 6 indicates ‘1’ when the SPC starts operation as the target.

- When SPC selected by initiator (when SEL signal true, I/O signal false, and own ID bit true)
- When SPC, as target, acquires right to use the bus to send ID bit on SCSI bus in reselection phase after asserting SEL signal and simultaneously asserts I/O signal

This bit indicates 0 when the SPC is disconnected by the initiator to clear the I-T nexus.

Bit 5: No Padding (READ):

This bit is intended as a confirmation mechanism for the Initiator to determine if padding bytes were actually sent or received when a data transfer with padding command is issued (e.g., SEND-DATA from MPU with Padding). Under certain circumstances the target might change phases after receiving the specified number of real data bytes, but before the padded bytes are sent or received. Bit 5 of the Nexus Status Register is used by the Initiator to determine if the padded data was actually sent or received.

Bit 5 indicates “1” when SPC did not send/receive padding bytes and “0” when the SPC has sent/received padding-data after transferring the specified number of real or non-padded data.

The value of bit 5 will not change until the next nexus is established. In the event a transfer command “without padding” is issued, the value for this bit does not change, and it still holds the previous data. This bit can not be referenced via the user program.

Bits 4: DMA Misalignment

This bit indicates the status of the current DMA misalignment. For details, see section 5.9. DMA MIS-ALIGNMENT. The MB86605 is capable of setting either a manual or auto DMA misalignment mode by the bits 4 and 3 of I/O Control register.

Bits 3 to 0: Last nexused partner’s ID

Bits 3 to 0 hold the nexused partner’s ID number.

They are set:

- When SPC operating as initiator and receives BSY signal from corresponding target in selection phase
- When SPC operating as target and receives BSY signal from corresponding initiator in reselection phase
- When SPC is operating as initiator in response to reselection request from target and asserts BSY signal
- When SPC operating as target in response to selection request from initiator and asserts BSY signal

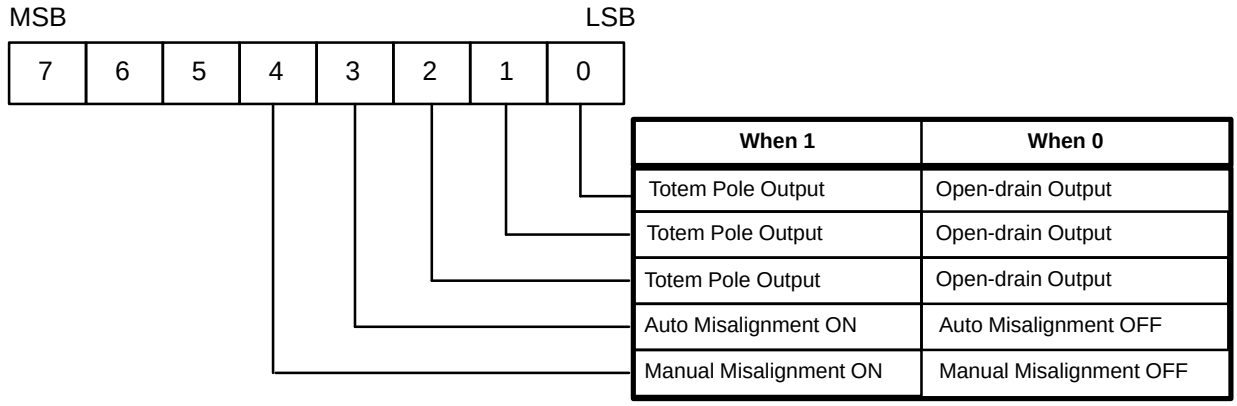
Even if the SPC is disconnected, these bits hold data and update it when the new nexus is established.

If the SPC is operating as the target in the automatic selection response mode*, it will keep operating automatically until it goes into the command phase. At this time, the host MPU can determine the initiator’s ID by referencing this register.

Similarly, if the SPC is operating as the initiator in the automatic reselection response mode*, it will keep operating automatically until it goes into the message-in phase. At this time, the host MPU can determine the target ID by referencing this register.

*See section 3.6.3. (4) SEL/RESEL Mode Setting Register.

– I/O Control Register (write) :



When 1	When 0
Totem Pole Output	Open-drain Output
Totem Pole Output	Open-drain Output
Totem Pole Output	Open-drain Output
Auto Misalignment ON	Auto Misalignment OFF
Manual Misalignment ON	Manual Misalignment OFF

Bits 7, 6, 5 : Unused

Unused bits. Write '0' at these bits when writing at this register.

Bit 4 :Manual DMA Transfer Misalignment

When '1' is set to this bit, the connection between the external DMA data bus pins and Data Register is swapped in LSB and MSB.

Note: Bit 4 has the same function with the bit 7 of SEL/RESEL ID Register in the MB86603 (DMA Transfer Misalignment bit).

Bit 3 :Auto DMA Transfer Misalignment

When '1' is set to the Bit 3, an auto misalignment is performed. This automatically aligns the data for each transfer instruction in word boundary when two or more data transfer commands are issued in the same data phase. When the data transfer is performed by the bus master operation in the PCI Bus interface mode, '1' must be set to this bit.

Note : When '1' is set to Bit 3 (Auto DMA Transfer Misalignment), The bit 4 setting (Manual DMA Transfer misalignment) is ignored.

Bit 2 : Interrupt Signal Output

This bit is used to select the type of output from either totem pole or open-drain output. When this bit=0, INT signal output is an open-drain. Since the signal line is internally pulled up with a 50kΩ resistor, it outputs "H" level at the inactive state. When this bit=1, INT signal output is a totem-pole.
The INT ouput pin (–INT) is an active-low output regardless of this bit setting.

Bits 1, 0 : SCSI Driver Output Control

Bit 1 is used to set the output type of on-chip single-ended driver for SCSI control signals –MSG, –C/D, –I/O, and –ATN. Bit 0 is used to set the output type of driver for –DB15 to –DB0, –UDBP, and –LDBP.

When these bits=0, the corresponding SCSI output pins are open-drain. When these bits=1, they are tote-pole output providing active-negation.

(7) Interrupt Status Register(R)/ SEL/RESEL ID Register(W)

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
XX	04h		14h	R	Interrupt Status							
					IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
				W	'0'	'0'	'0'	'0'	SEL/RESEL ID			
									SI3	SI2	SI1	SI0
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

– Interrupt Status Register (Read) :

MSB LSB

IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0

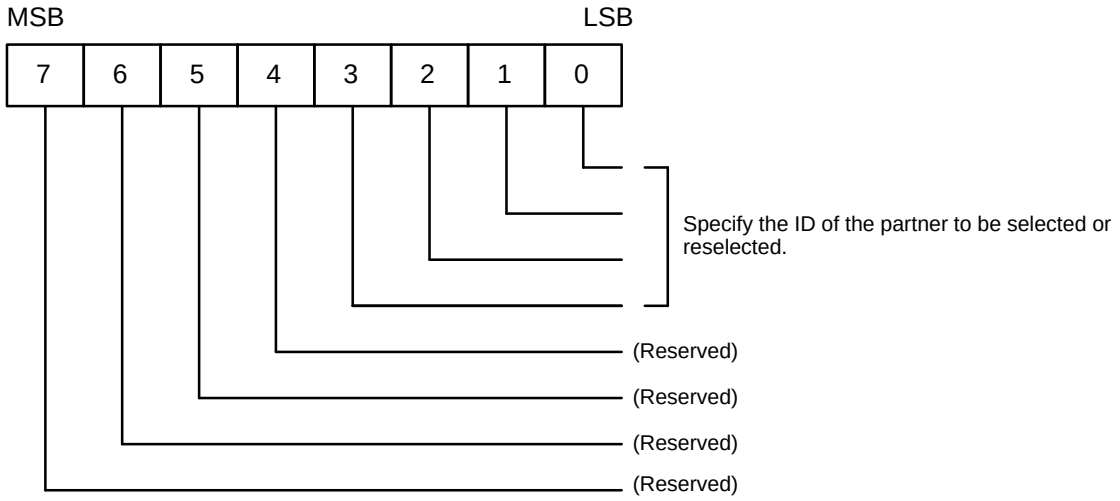
This register indicates the interrupt source in 8-bit code.

This register is 8-byte FIFO and holds up to eight pending interrupt codes until the command is complete. Read this register when bit 7 of SPC Status Register (Addr 02h) = 1. That bit indicates 0 when MPU has read out all the interrupt codes held by the SPC. This register may hold two or more interrupt codes (up to 8 codes), and it indicates the next interrupt code by reading the current code by host MPU.

The contents of interrupt status register show the interrupt code occurred at the step indicated by the Command Step Register (addr 05h). Therefore, read both this register and command step register at the same time. For the byte access, read the command step register after reading this register. Otherwise, the next interrupt status will not be indicated.

Note : when 1 is set to the bit 7 of Auto Operation Mode Register (addr 1Ch) , commands issued will be ignored if all the interrupt codes are not read out from this register. Also, do not access this register after the command is issued until SPC BUSY bit (Bit 6) of SPC STatus Register (addr 02h) =0. Read out the interrupt code for the command which has been executed before issuing the new command.

–SEL/RESEL ID Register (Write) :



Note: Write 0 at reserved bits.

Bits 3 to 0: SEL/RESEL-ID setting

This register has two functions as follows;

1) Initial setup for the transfer mode and parameter

The MB86605 incorporates a transfer parameter memory that stores transfer parameters for every partner device (up to 15 devices). By storing the valid transfer mode and parameter at which first nexus is established, the SPC can automatically set each transfer mode and parameter from the ID of partner device at the second and later nexuses.

To set the transfer mode and transfer parameters (period and offset value), specify the corresponding partner's ID for the SEL/RESEL-ID register and then set the following registers.

- Transfer mode register (address 12h)
- Transfer period register (address 13h)
- Transfer offset register (address 14h)

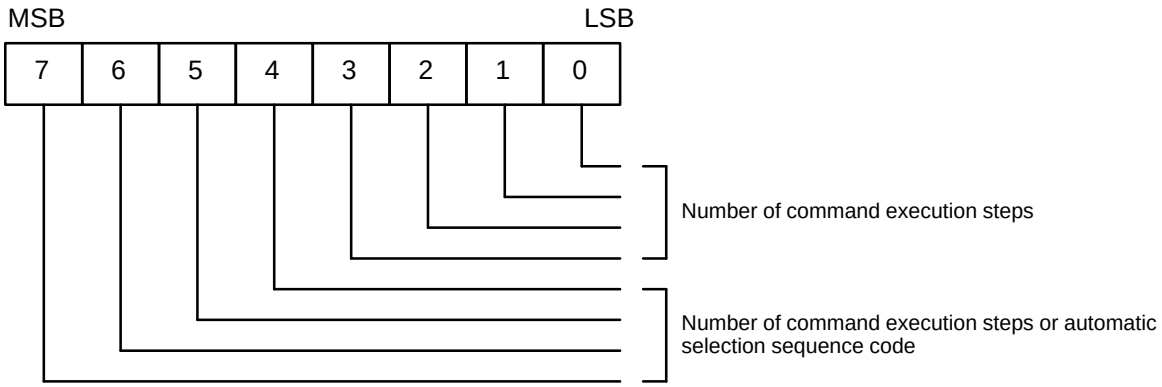
To target the system (single initiator) in which the selection phase is executed by sending the target ID bit only, set the transfer parameters with the initiator's ID taken as 0.

2) Set for the bus device ID of partner device to select/reselect.

(8) Command Step Register(R) / Command Register(W)

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
XX	05h		15h	R	Command #/Auto Sel Seq Code				Command Step			
					CS7	CS6	CS5	CS4	CS3	CS2	CS1	CS0
				W	User pro operate	Command Code						
						CM6	CM5	CM4	CM3	CM2	CM1	CM0
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

– Command Step Register (Read) :



This register indicates the numbers of command execution steps and program steps in the user program. This is an 8-byte FIFO register that holds the command steps corresponding to the interrupt status register.

This register and the Interrupt Status Register must be read out at the same time except the case of the byte access. For byte access, read out this register following the read out of the interrupt status register.

Bits 7 to 4: Number of command execution steps in user program or sequence code in automatic selection response mode

(1) When SPC is operating under user program

This register serves as the ring counter indicating the number of command execution steps. This counter increments each time a command execution is started.

Only Discrete Commands and not Special Commands are counted. For each command, see CHAPTER 4.

(2) When SPC operating in automatic selection response mode (target only)

These bits indicate the response sequence type.

Bits 3 to 0: Number of command execution steps

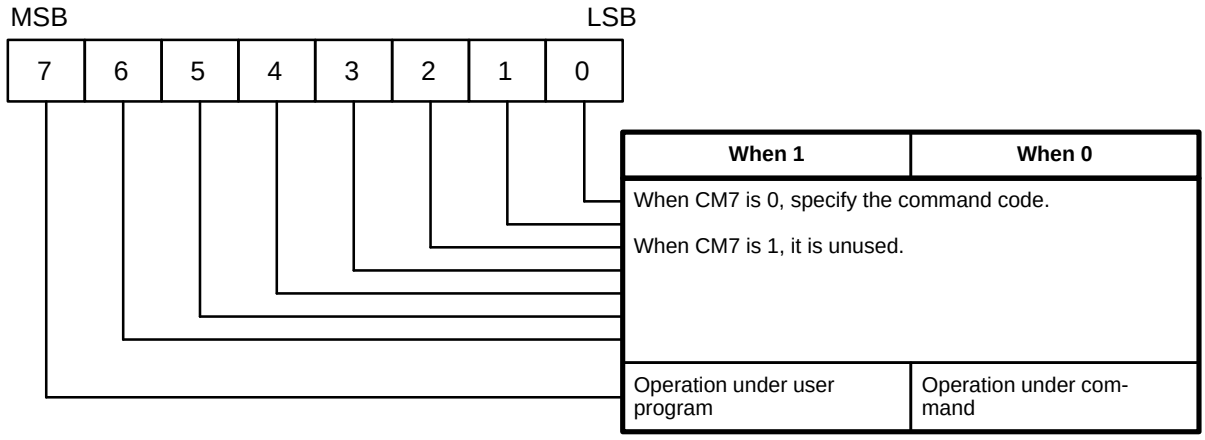
Bits 3 to 0 indicate the number of command execution steps.

The step code is specified for each command. For the command step, see CHAPTER 4.

This also applies to the cases where the automatic selection/reselection response mode is set. If the automatic receive mode is set, information is received; this does not affect these bits.

(Example)
If the “Initial Phase Error and MSG Received” interrupt occurs, the execution steps at which a phase error occurs are indicated.

– Command Register (write) :



Bit 7: Operation under user program

Bit 7 specifies whether to operate the SPC under the user program or under the command issued from the MPU. The functions of bits 6 to 0 vary with the setting of this bit.

When 0 Write the command code at bits 6 to 0.

For the command code, see CHAPTER 4. (It is unnecessary to care for the status of bit 7 since it is fixed with ‘0’ for all commands which can be executed by writing to this register.)

When 1: The bits 6 to 0 have no meaning since the user program operation is selected.

The MB86605 can automatically execute the commands in the user program memory in sequential. When ‘1’ is written at this bit after writing the start address for the command execution at the addresses 18h and 19h in the user program memory window, the SPC fetches the command from the address written and executes it.

(9) Data Block Register(R/W)

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
XX	06h	07h	17h	R	BL15	BL14	BL13	BL12	BL11	BL10	BL9	BL8
XX	07h	06h	16h	W	BL7	BL6	BL5	BL4	BL3	BL2	BL1	BL0
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

Specify the number of blocks to be transferred in the data phase. Write at this register when SPC Busy=0 (bit 6 of the SPC status register = '0'.)

For fixed length — Specify the number of blocks to be transferred.
For variable length — 0001H must be specified.

The read value of this register becomes valid at the report of termination (including normal/abnormal termination).

At normal termination, the values of both the data block register and byte register go to 00.

At abnormal termination, the numbers of blocks and bytes unsuccessfully transferred are reported.

(Example)

Set value: Block is 000AH and byte length is 000100H.

Normal: Block is 0000H and byte length is 000000H.

Abnormal 1: Block is 0004H and byte length is 0000ACH.
Data unsuccessfully transferred is 4 blocks and ACH bytes.
Data successfully transferred is 5 blocks and 54H bytes.

Abnormal 2: Block is 0004H and byte length is 000000H.
Data unsuccessfully transferred is 4 blocks.
Data successfully transferred is 6 blocks.

Abnormal 3: Block is 0005H and byte length is 000100H.
Data unsuccessfully transferred is 6 blocks (5 blocks and 100H bytes).
Data successfully transferred is 4 blocks.

Note: Register value when read
When data transfer is performed by the user program in the PCI bus interface mode, specify the variable length (0001H).

(10) Data Byte / MC Byte Register (R/W)

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
XX	08h	0Ah	1Ah	R / W	BY23	BY22	BY21	BY20	BY19	BY18	BY17	BY16
XX	09h	09h	19h		BY15	BY14	BY13	BY12	BY11	BY10	BY9	BY8
XX	0Ah	08h	18h		BY7	BY6	BY5	BY4	BY3	BY2	BY1	BY0
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

– Data byte register (BY23 to BY0) :

Specify the number of bytes transferred in the data phase. Write at this register when SPC Busy=0 (bit 6 of the SPC status register = 0.)

For fixed length — Specify the length of one block. Only even number (word boundary) can be specified for the block length. If odd number is specified, the data are corrupt at the block boundary.

As an additional note; If it is necessary to perform the block transfer of odd length, issue the commands to transfer the variable length data (odd number data) as the number of block to be transferred or set the total transfer counts as the variable length data transfer. because the MB86605 can be issued two or more transfer commands in one data phase.

For variable length — Specify the number of bytes transferred.

– MC byte register (BY7 to BY0) :

Specify the number of bytes transferred in the message and command phases.

Specify the number of bytes transferred for this register only when issuing any command requiring the setting of the number of bytes transferred.

The read value of this register becomes valid at the report of termination (including normal/abnormal termination).

At normal termination, the values of both the data block register and byte register go to 00.

At abnormal termination, the numbers of blocks and bytes unsuccessfully transferred are reported.

(Example)

Set value: Block is 000AH and byte length is 000100H.

Normal: Block is 0000H and byte length is 000000H.

Abnormal 1: Block is 0004H and byte length is 0000ACH.
Data unsuccessfully transferred is 4 blocks and ACH bytes.
Data successfully transferred is 5 blocks and 54H bytes.

Abnormal 2: Block is 0004H and byte length is 000000H.
Data unsuccessfully transferred is 4 blocks.

Data successfully transferred is 6 blocks.

Abnormal 3: Block is 0005H and byte length is 000100H.
Data unsuccessfully transferred is 6 blocks (5 blocks and 100H bytes).
Data successfully transferred is 4 blocks.

Note: Register value when read
The read values of the data byte and MC byte registers become valid at the report of termination (normal/abnormal termination). The initial value is indicated before the termination report.

When using as the MC byte register, the initial value is always indicated.

(11) SCSI Control Signal Status Register (R) / Diagnostic Control Register(W)

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
XX	0Bh		1Bh	R	REQ	ACK	ATN	SEL	BSY	MSG	C/D	I/O
				W	Diag REQ	Diag ACK	Diag ATN	'0'	Diag BSY	Diag MSG	Diag C/D	Diag I/O
Initial Value					'X'	'X'	'X'	'X'	'X'	'X'	'X'	'X'

– SCSI Control Signal Status Register (read) :

This register indicates the status of the SCSI control signals. The value of this register is valid even in the diagnostic mode.

– Diagnostic Control Register (write) :

This register is used to emulate the SCSI control signals in the diagnostic mode. By writing '1' at the bit where each SCSI control signal is assigned, the signal is asserted.

The value of this register is valid when issuing the **INIT DIAG START** or **TARG DIAG START** command and remains valid until the **DIAG END** command is issued.

Note: Write '0' at bit 4 when writing at this register.

(12) Transfer Mode Register

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
XX	0Ch		1Ch	R/W	Sync Trans	Wide Trans	‘0’	‘0’	‘0’	‘0’	‘0’	‘0’
Initial Value					‘0’	‘0’	‘0’	‘0’	‘0’	‘0’	‘0’	‘0’

This register can store transfer mode parameters for every connected-device. Set this register after specifying the target bus device IDs at the SEL/RESEL ID Register (addr 04h). Also, once this register is set, it is automatically set when SEL/RESEL ID Register setting is done. However, a system reset or software reset initializes this register.

Bit 7: Transfer mode

Specify the data phase transfer mode of the bus device ID specified by the SEL/RESEL-ID register (address 04h).

When 1: Transfer data in synchronous mode.

When 0: Transfer data in asynchronous mode.

Bit 6: Transfer width setting

Specify the transfer width of bus device ID specified by the SEL/RESEL-ID register.

When 1: Transfers 16-bit wide SCSI data in the data phase.

When 0: Transfers 8-bit SCSI data in the data phase.

Bits 5 to 0 : Unused

Write ‘0’ at those bits when writing at this register.

(13) Transfer Period Register(R/W)

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
XX	0Dh		1Dh	R/W	‘0’	‘0’	‘0’	Transfer Period				
								TP4	TP3	TP2	TP1	TP0
Initial Value					‘0’	‘0’	‘0’	‘0’	‘0’	‘0’	‘0’	‘0’

The Transfer Period Register is used to set the synchronous transfer period for the bus device ID specified in the SEL/RESEL ID Register. This register can store transfer period for every connected-device. Set this register after the specifying the target bus device IDs at the SEL/RESEL ID Register (addr 04h). Also, once this register is set, this register is automatically set when SEL/RESEL ID Register is sete. A system reset or software reset initializes this register.

Bits 7 to 5 : Unused

Write ‘0’ at those bits when writing at this register.

Bits 4 to 0 : Transfer Period

Specify the synchronous transfer period of the bus device ID specified by the SEL/RESEL-ID register (address 04h). The relationship between the input frequency and maximum synchronous transfer rate is represented by the following expression.

(Maximum synchronous transfer rate) x (Transfer period) = Input frequency (Example: 10 Mbytes/s x 2 = 20 MHz)

If the input frequency is high, do not specify a baud rate greater than the actual value.
(Example) 40 MHz — 4 or greater, 30 MHz — 3 or greater, 20 MHz — 2 or greater

(Unit: Tclf)

TP4	TP3	TP2	TP1	TP0	TRANS PERIOD	ASSERT PERIOD	NEGATE PERIOD	TP4	TP3	TP2	TP1	TP0	TRANS PERIOD	ASSERT PERIOD	NEGATE PERIOD
0	0	0	0	1	Prohibited	Prohibited	Prohibited	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

(14) Transfer Offset Register(R/W)

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
XX	0Eh		1Eh	R/W	'0'	'0'	'0'	Transfer Offset				
								TO4	TO3	TO2	TO1	TO0
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

The Transfer Offset Register is used to set the synchronous transfer maximum offset value for the bus device ID specified in the SEL/RESEL ID Register. This register can store transfer offset for every connected-device. Set this register after specifying the target bus device IDs at the SEL/RESEL ID Register (addr 04h). Also, once this register is set, this register is automatically set when SEL/RESEL ID Register setting is done. A system reset or software reset initializes this register.

Bits 7 to 5 : Unused

Write '0' at those bits when writing at this register.

Bits 4 to 0 : Transfer Offset

Specify the maximum synchronous transfer offset value of the bus device ID specified by the SEL/RESEL-ID register.

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

(15) Modified Byte Register(R) / Window Access Register(W)

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
XX	0Fh		1Fh	R	'0'	Modified Byte Register						
						MB6	MB5	MB4	MB3	MB2	MB1	MB0
				W	Window Select		'0'	'0'	'0'	'0'	'0'	MCS Window sel
					WS1	WS0						
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

– Modified Byte Register (read) :

This register serves as the 7-bit down counter indicating the number of bytes transferred by the SPC. The value indicated by this register varies with the phase as follows:

(1) Message, command, and status phases

The value indicates the number of bytes (valid bits of RECEIVE-MCS buffer) received by the SPC, or the number of bytes (sent from SEND-MCS buffer) sent by the SPC.

In the automatic selection response mode or automatic target receive mode, the different phases may be received continuously. At this time, this register indicates the total number of bytes received in each phase.

(2) Data phase

The value indicates the number of bytes transferred between the SPC and MPU or external memory.

The seven lower bits of the data byte register are set and are decremented at every access.

If data transfer is terminated with data remaining in the data register (abnormal termination), the remaining number of data bytes can be calculated from the values of the data byte register and this register. For details of this calculation, see section 5.7.5.

If data transfer is terminated with data remaining in the data register, the remaining data will be invalid. Therefore, when continuing data transfer, calculate the remaining number of bytes of the data register and return the data pointer by that value.

– Window Address Register (write) :

This register is used to specify the register window (10h to 1Fh) in the 16-bit bus interface mode and to switch the address 20h to 2Fh to the SCAM register window in the PCI bus interface mode.

There are four types of window: initial setting register window, MCS buffer window, SCAM register window, and user program memory window.

Bits 7 and 6: Window selection

Bits 7 and 6 select one window from four windows at addresses 10h to 1Fh.

WS7	WS6	Window Selected
0	0	MCS buffer window *
0	1	SCAM Register Window
1	0	User program memory window *
1	1	Initial setting register window

* : Window selection for MCS buffer and User Program Memory is prohibited in the PCI bus interface mode.

Bits 5 to 1 : Unused

Write '0' at these bits when writing at this register.

Bit 0 : MCS Buffer Window Switch

Bit 0 switches the first half/second half of the MCS buffer. Specify the upper most bit of the MCS buffer address. The first half/second half of MCS buffer can be switched in 16 bytes. *Do not write '1' at this bit in the PCI bus interface mode.*

3.6.2. INITIAL SETTING REGISTER

In the MB86605, does not provide a setup command (SET UP), so the registers allocated in the initial setting window can be written at any time. However, in order to make the written values valid, the following conditions must be satisfied.

1) For Clock Conversion Register : The value for this register is valid only after the Software Reset command. The value written during the execution of the Software Reset becomes valid after releasing the software reset. Do not rewrite the register except during the execution of the Software Reset.

2) For other registers : The values for those registers are valid after the Software Reset and the end of the SPC operation. The value written before issuing the Software Reset command or during the execution of the Software Reset becomes valid after releasing the software reset. The value written while the SPC is working (during SPC Busy = '1') becomes valid after the completion of the SPC operation (after SPC Busy ='0'). (The value written while SPC Busy ='0' becomes valid immediately.)

The initial setting register is initialized by the external reset, and the initial values are all '00h'. Even if the software reset command is issued, the register holds the previous value. (The rewrite can be done during the execution of the software reset.)

(1) Clock Conversion Setting Register(R/W)

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
11	10h		20h	R/W	Clock Conversion							
					CC7	CC6	CC5	CC4	CC3	CC2	CC1	CC0
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

Set the values listed in table below according to the input frequency.

CC7	CC6	CC5	CC4	CC3	CC2	CC1	CC0	Input Frequency	Internal Operating Freq.
0	0	0	0	1	0	1	1	20.00 [MHz]	20.00 [MHz]
0	0	0	1	0	0	0	0	20.00 [MHz]	10.00 [MHZ]
0	0	1	1	0	0	1	0	30.00 [MHz]	15.00 [MHz]
0	0	1	1	1	0	0	0	30.00 [MHz]	10.00 [MHz]
0	1	0	1	0	0	1	1	40.00 [MHz]	20.00 [MHz]
0	1	0	1	1	0	0	1	40.00 [MHz]	13.33 [MHz]

* Do not write at this register except during the software reset.

(2) Own ID Setting Register (R/W)

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
11	11h		21h	R/W	'0'	'0'	'0'	'0'	Own ID OI3	OI2	OI1	OI0
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

Set the bus device own ID number that outputs when the arbitration and selection/reselection request are output.

Bits 7 to 4 are unused. Write '0' at these bits when writing at this register.

(3) Response Operation Mode Setting Register (R/W)

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
11	12h		22h	R/W	RESEL enbl	SEL enbl	Auto RX enbl	Parity Mode	'0'	'0'	Single Init	No Arbit
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

Bit 7: Reselection Response Enable

Bit 7 specifies whether or not the SPC responds as the initiator to a reselection request from the target.

When 1: The SPC responds to a reselection request from the target and establishes the I-T nexus.

The operation after the I-T nexus is established depends on the SEL/RESEL Operation Mode Setting Register.

When 0: The SPC does not respond to a reselection request from the target (does not assert BSY signal in reselection phase).

Bit 6: Selection Response Enable

Bit 6 specifies whether or not the SPC responds as the target to a selection request from the initiator.

When 1: The SPC responds to a selection request from initiator and establishes the I-T nexus.

The operation after the I-T nexus is established depends on the SEL/RESEL Operation Mode Setting Register.

When 0: The SPC does not respond to a selection request from the initiator (does not assert BSY signal in selection phase).

Bit 5: Automatic receive mode

Bit 5 specifies whether or not to receive a message in response to the attention condition generated by the initiator when the SPC is operating as the target, or to receive the information of the required error phase when the SPC is executing the command as the initiator with a phase-transition series.

The automatic receive mode that can be specified by this bit is as follows:

(1) Operation as initiator

a. Phase error occurs and target requires message-in phase.

When 1: Recognize a phase error and receive one message.

Negate the ATN signal if asserted, and then receive one message.

Report the Initial Phase Error and MSG Received interrupt with the last ACK signal asserted.

When 0: Recognize a phase error, negate the ATN signal if asserted, and then report the Initial Phase Error interrupt.

b. Phase error occurs and target requires status phase.

When 1: Recognize a phase error and receive one status.

Negate the ATN signal if asserted, and then receive one status.

Report the Initial Phase Error and Status Received interrupt with the last ACK signal asserted.

When 0: Recognize a phase error, negate the ATN signal if asserted, and then report the Initial Phase Error interrupt.

Note: The automatic receive mode is not activated for a phase error during transfer (all bytes not transferred). For a phase error during transfer, the Phase Error in Transfer Progress interrupt is reported.

(2) Operation as target

a. A command that will be terminated in phases other than the message-out phase is executed and the ATN signal is asserted when the last ACK signal is asserted in the last phase.

When 1: Detect the attention condition generated by the initiator, switch to the message-out phase, and then receive one message.

Report the Command Complete (ATN Condition Detected) and MSG Received interrupt.

When 0: Detect the attention condition generated by the initiator and report the Command Complete (ATN Condition Detected) interrupt.

Note: A command that will be terminated in the message-out phase is executed and the Command Complete (ATN Condition Detected) interrupt occurs irrespective of the setting of this bit when the attention condition is continuously detected.

b. The SPC is executing the command as the target and switches to the next phase or detects the attention condition generated by the initiator at the transfer block boundary.

When 1: Detect the attention condition generated by the initiator, switch to the message-out phase, and then receive one message.

During execution of the data phase, stop data transfer at the transfer block boundary, switch to the message-out phase, and then receive one message.

Report the Command Stop (ATN Condition Detected) and MSG Received interrupt.

When 0: Detect the attention condition generated by the initiator and report the Command Stop (ATN Condition Detected) interrupt.

During execution of the data phase, stop data transfer at the transfer block boundary and report the Command Stop (ATN Condition Detected) interrupt.

Notes:

1. When a parity error occurs on the system side of the target and data transfer is stopped between blocks (stop in blocks) during execution of the data-in phase, and if the attention condition is generated by the initiator, a DMA Parity Error or MPU Parity Error will be reported without reaction to the attention condition.
2. With the SPC operating as the target, if it redetects the attention condition after detecting the attention condition, changing to the message-out phase, and then receiving one message, it will report the following interrupt.

   Interrupt code = 61H
   Sequence step = XXH (value when 1 added to number of steps in phase before changing to message-out phase)

(Example)
When the **SEND DATA from MPU** command is executed and the attention condition is detected at the end of transfer, the SPC switches to the message-out phase, receives one message, and then redetects the attention condition. In this case, the SPC will report the following two interrupts.

	Interrupt Code	Sequence Step
1	62H	02H
2	61H	03H

Saving information automatically received

(1) When operating as Initiator :

Saves from address 0 in the Receive MCS Buffer.

(2) When operating as Target:

Saves from address 0 in the Receive MCS Buffer when the executed command is on other than command phase. When the command is on command phase, saves from the continuous address of CDB received in the Receive MCS Buffer.

Automatic Receive in the user program mode

The SPC does not receive information automatically regardless of this bit setting.

Bit 4: Data phase parity error

Bit 4 specifies the operation when the SPC is operating as the target and detects a parity error in the data phase.

When 1: Stop transfer and report the interrupt when a parity error is detected in the data phase (stop in bytes).

For details, see section 5.6. Operation when Error Detected

When 0: Stop transfer and report the interrupt when transfer of the block where a parity error is detected in the data phase is terminated (stop in blocks).

Bit 1: Single initiator option

Set Bit 1 to 1 when the SPC is included in a system that will operate on a single initiator option in the SCSI-1 mode.

When 1: When SPC selected as target

Respond even if only one ID bit (target ID) is output on the data bus in the selection phase.

When 0: When SPC selected as target

Do not respond if only one ID bit (target ID) is output on the data bus in the selection phase.

This bit is used to specify the target operation. Therefore, the own ID bit and target ID bit (two bits) are output on the data bus in the selection phase irrespective of the setting of this bit.

Bit 0: Arbitration

Bit 0 specifies whether or not the SPC executes the arbitration phase.

Set Bit 0 to 1 if the SPC is included in a system that will operate on a signal initiator option in the SCSI-1 mode and does not execute the arbitration phase.

When 1: Do not execute arbitration phase.

When 0: Execute arbitration phase.

(4) SEL/RESEL Mode Setting Register(R/W)

This register sets the automatic response mode when the SPC is selected or reselected.

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
11	13h		23h	R/W	SEL/ RESEL op	Auto Sel	Auto Resel	Targ. Phase	'0'	'0'	'0'	'0'
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

Parameter	When 1	When 0
Target's phase after the selection	Only Message phase.	Only Message or Command phase.
Auto Selection response	Enable	Disable
Auto Reselection response	Enable	Disable
Sel/Resel Response Mode	User Program	Auto Response

Bit 7: SEL/RESEL Response Operation Mode

Bit 7 specifies whether or not the response operation activated at the Selection/Reselection is performed by the user program or Auto Response Mode preset in the SPC.

When 1 : The SPC starts the response operation from the auto response start address specified in the addresses 1Ah and 1Bh in the user program window when selected/reselected.

When 0 : The SPC starts the operation according to the contents of bits 6 to 4 of this register.

Bit 6: Automatic reselection response (Auto Reselection Response)

Setting Bit 6 changes the operation of the initiator when reselected.

When 1: In response to a reselection request, execute the automatic reselection response until a message is received (Figure 3.1).

When 0: In response to a reselection request, generate the RESELECTED interrupt.

Bit 5: Automatic selection response* (Auto Selection Response)

Setting Bit 5 changes the operation of the target when selected.

When 1: In response to a selection request, execute the automatic selection response until a message or command is received (see Figure 3.2).

When 0: In response to a selection request, generate the SELECTED or SELECTED WITH ATN interrupt.

* Use bits 7 and 6 of the response mode setting register to specify whether or not the SPC responds (establishes I-T nexus) to a selection or reselection request.

Bit 4: Phase after selection

Bit 4 specifies the automatic response mode of the target and also defines the phase after selection as the message phase (SCSI-2 mode) or as the message or command phase.

When 1: Defined as message-out phase. Report the AUTO Mode Phase Error if the ATN signal is not asserted by the initiator in the selection phase.

When 0: Defined as message-out or command phase. Switch to either phase depending on the state of the ATN signal in the selection phase.

Auto Reselection Response by Initiator

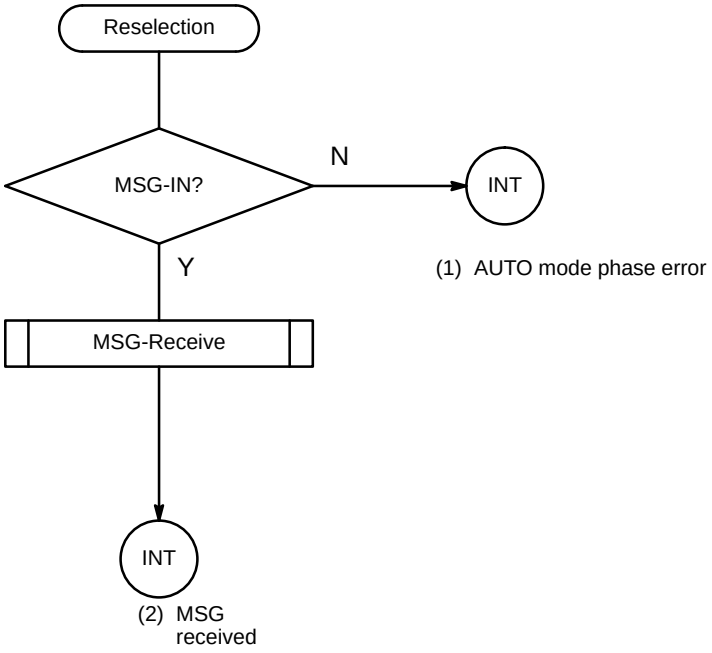


Fig. 3.1 Auto Reselection by Initiator

• Auto Sequence (INITIATOR)

Flow Chart	Step	Description
Reselection	0	Detect reselection. Establish nexus.
MSG-IN	1	Wait for MSG-IN phase. MSG-IN phase executing
END	2	Receiving MSG completed

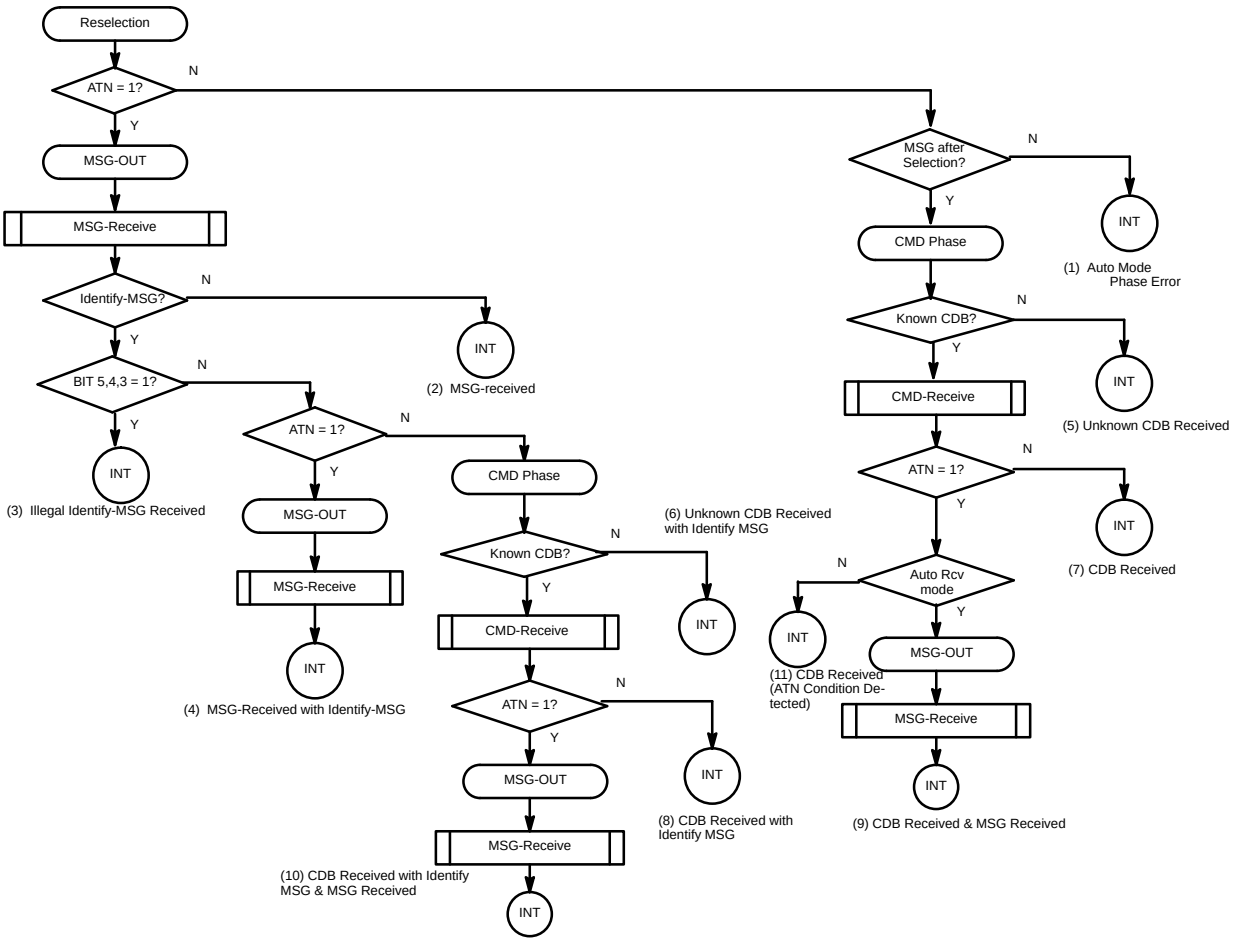
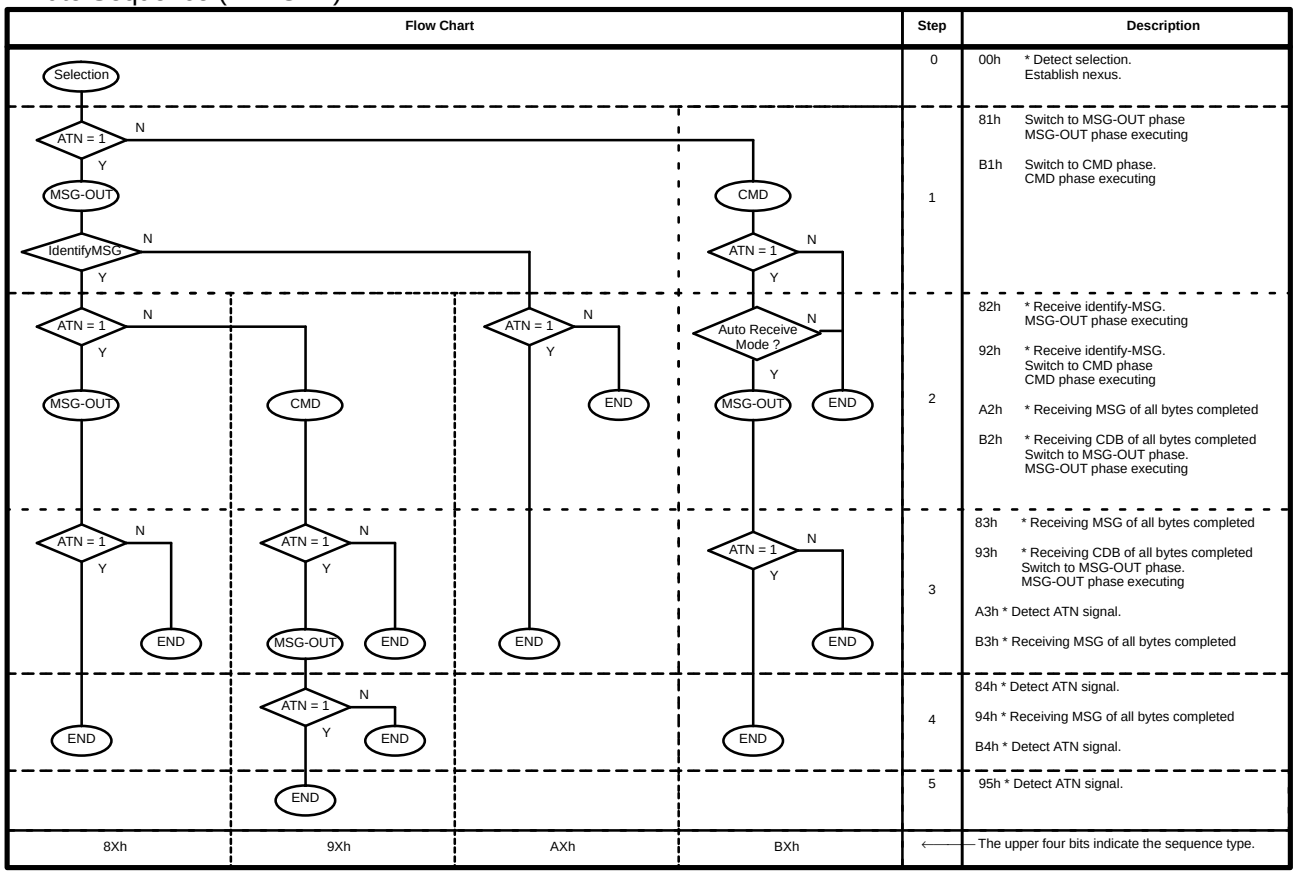


Figure 3.2 Flow Chart of Auto Selection Responseby Target

(9) (I) CMD Received and MSG Received
(10) (II) CMD Received with Identify-MSG and MSG Reseived

• Auto Sequence (TARGET)



Note: If bits 5, 4, and 3 of the received identify MSG are 1, the Illegal Identify MSG interrupt will occur.

At this time, the command step codes are as follows:

When initiator continuously asserts ATN signal — 82h

When initiator does not continuously assert ATN signal — 92h

(5) SEL/RESEL Retry Setting Register(R/W)

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
11	14h		24h	R/W	SEL/RESEL Retry			Retry Start Time				
					SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

Bits 7 to 5: Specifying number of retry times for selection/reselection

Even if the selection/reselection timeout occurs during execution of the command including selection/reselection, the number of retries specified by these bits is performed.

Specify the number of retry times according to the following table.

SR7	SR6	SR5	Number of times	SR7	SR6	SR5	Number of times
1	1	1	Unlimited	0	1	1	8
1	1	0	255	0	1	0	2
1	0	1	32	0	0	1	1
1	0	0	16	0	0	0	0

Bits 4 to 0: Specifying retry start time

If the selection/reselection timeout occurs at the first selection/reselection attempt, a retry is started after the time specified by these bits from the point that a bus free condition is established.

When 00 are specified, no time is set and a retry is started as soon as the bus free state is established.

$t_{CLF} \times C_{NV} \times [(SR4 \times 2^{19}) + (SR3 \times 2^{17}) + (SR2 \times 2^{15}) + (SR1 \times 2^{13}) + (SR0 \times 2^{11})]$

t_{CLF}: Input frequency, 1 cycle
C_{NV}: (Numerical value represented by bits 4 and 3 of clock conversion register)
SR4 to SR0: Set values of bits 4 to 0

(Example)
When input frequency 30 MHz, internal operating clock frequency 10MHz, t_{CLF} = 33.33 ns and C_{NV} = 3 and so ;

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

Consequently, the time range can be set by this register as follows:

(1) Upper limit (SR 4 to SR0 = 1) — 69 ms

(2) Lower limit (SR4 to SR1 = 0, SR0 = 1) — 204 μs

(6) SEL/RESEL Timeout Setting Register (R/W)

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
11	15h		25h	R/W	SEL/RESEL Timeout							
					ST7	ST6	ST5	ST4	ST3	ST2	St1	ST0
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

Set the selection/reselection timeout time at this register. The timeout time can be represented by the following expression (the timeout is detected even if this register = 00h).

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

t_{CLF}: Input frequency, 1 cycle

C_{NV}: (Numerical value represented by bits 4 and 3 of clock conversion register)

ST7 to ST0: Set values of bits 7 to 0

(Example)

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

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

Consequently, the time range can be set by this register as follows:

(1) Upper limit (ST7 to ST0 = 1) — 4.4 sec

(2) Lower limit (ST7 to ST1 = 0, ST0 = 1) — 204 µs

When '00h' is written at this register, the timeout is as follows:

t_{CLF} x C_{NV} X (FFFFFFh) x 8

(Example)

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

33.33 x 3 x FFFFFFh (= 16777215) x 8 = approx 13.4 sec.

(7) REQ/ACK Timeout Setting Register(R/W)

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
11	16h		26h	R/W	REQ/ACK Timeout							
					RT7	RT6	RT5	RT4	RT3	RT2	RT1	RT0
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

Set the REQ/ACK signal timeout time.

Asynchronous transfer (target)	Time required for initiator to assert ACK signal after asserting REQ signal
Asynchronous transfer (initiator)	Time required for target to negate REQ signal after asserting ACK signal
Synchronous transfer (target only)	Time required for target to receive signal for setting offset value to 0 from initiator after sending REQ signal

The timeout time can be represented by the following expression (the timeout is not detected for 00h).

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

t_{CLF}: Input frequency, 1 cycle

C_{NV}: (Numerical value represented by bits 4 and 3 of clock conversion register)

RT7 to RT0: Set values of bits 7 to 0

(Example)

When input frequency 30 MHz, internal operating clock frequency 10MHz (clock conversion =38h), t_{CLF} = 33.33 ns and C_{NV} = 3

$$33.33 \times 3 \times (RT7 \times 2^{28}) + (RT6 \times 2^{26}) + (RT5 \times 2^{24}) + (RT4 \times 2^{22}) + (RT3 \times 2^{20}) + (RT2 \times 10^{18}) + (RT1 \times 10^{16}) + (RT0 \times 2^{14})]$$

Consequently, the time range can be set by this register as follows:

(1) Upper limit (RT7 to RT0 = 1) — 35.8 sec

(2) Lower limit (RT7 to RT1 = 0, RT0 = 1) — 1.6 ms

(8) Asynchronous Setup Time Setting Register(R/W)

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
11	17h		27h	R/W	‘0’	‘0’	‘0’	‘0’	Async Setup Time AT3	AT2	AT1	AT0
Initial Value					‘0’	‘0’	‘0’	‘0’	‘0’	‘0’	‘0’	‘0’

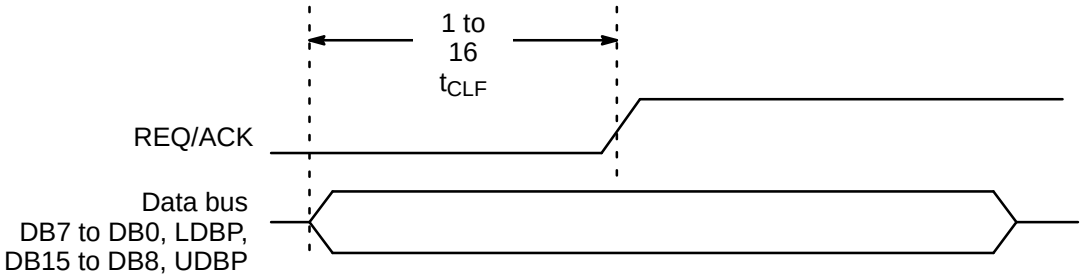
This register sets the timing for asserting the REQ/ACK signal for data in asynchronous transfer mode. (The range of setup time is from 1 to 16tclf.) Bit 7 to Bit 4 are unused. Write ‘0’ at those bits.

(1) Target

Time required to assert REQ signal after setting data to data bus

(2) Initiator

Time required to assert ACK signal after setting data to data bus



AT3	AT2	AT1	AT0	SET UP TIME
0	0	0	1	1 tclf
0	0	1	0	2 tclf
0	0	1	1	3 tclf
~	~	~	~	~
1	1	1	1	15 tclf
0	0	0	0	16 tclf

(9) Parity Error Detection Setting Register(R/W)

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
11	18h		28h	R/W	SCSI Pt check	SCSI Pt gen.	Sys odd/even	MPU Pt check	MPU Pt gen.	'0'	DMA Pt check	DMA Pt gen.
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

Bit	Item	When 1	When 0
Bit 0	DMA Parity Generate	Yes	No
Bit 1	DMA Parity Check	Yes	No
Bit 2	(Reserved)	–	–
Bit 3	MPU Parity Generate	Yes	No
Bit 4	MPU Parity Check	Yes	No
Bit 5	PARITY Type Of System Side	Odd	Even
Bit 6	SCSI Parity Generate	Yes	No
Bit 7	SCSI Parity Check	Yes	No

This register is used to set the parity type. If the system (both MPU and DMA buses) does not support parity, '11X01001' should be set to this register. Also, if the PCI bus interface mode is used, '11X01001' should be set since the parity is processed by the PCI bus interface circuit.

Note: Enabling/disabling parity generation is valid only in the data phase.

Bit 7: SCSI Parity Check

When 1: (1) Operation as initiator

Check the parity of the input data in the message-in, status and data-in phases, and of the output data in the data-out phase.

(2) Operation as target

Check the parity of the input data in the message-out, command and data-out phases, and of the output data in the data-in phase.

When 0: Do not check the parity.

Bit 6: SCSI Parity Generate

When 1: (1) Operation as initiator

Generate the parity for the input data in the data-in phase, and for the output data in the data-out phase.

(2) Operation as target

Generate the parity for input data in the data-out phase, and for the output data in the data-in phase.

When 0: Do not generate the parity.

Bit 5: Parity type of system side

Bit 5 specifies the MPU/DMA parity type for a system including the SPC.

When 1: Odd parity

When 0: Even parity

Bit 4: MPU parity check

When 1: (1) Register write

Check the parity of data when writing to the SPC registers. Parity is not checked when data is read from the registers.

(2) Data register read/write

Check the parity of data when writing and reading to and from the data register during execution of the data phase in the program transfer mode.

When 0: Do not check the parity.

Bit 3: MPU parity generate

When 1: Generate the parity for data when writing to and reading from the data register during execution of the data phase in the program transfer mode.

When 0: Do not generate the parity.

Note: When data is written to the register, parity is not generated irrespective of the setting of this bit. When data is read from the register, parity is always generated. An MPU parity check should not be performed for a system without a parity bit on the MPU data.

Bit 1: DMA parity check

When 1: Check the parity at data input and output from and to the DMA data bus during execution of the data phase in the DMA transfer mode.

When 0: Do not check the parity.

Bit 0: DMA parity generate

When 1: Generate the parity on data input and output from and to the DMA data bus during the execution of the data phase in DMA transfer mode.

When 0: Do not generate the parity.

Notes on Parity :

1. When accessing the external bus, parity check/generate is not done regardless of this register setting.
2. When SCSI data bus parity check is done in the system that does not have parity bits on the MPU and DMA data buses, set the MPU and DMA parity generate bits to 1. Also, in this case, pull up the UDP, LDP, UDMDP, and LDMDP pins with approx. 10kΩ resistor.

(10) Interrupt Enable Setting Register(R/W)

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
11	19h		29h	R/W	All Int enbl	–	Auto S/R enbl	Sel/resel enbl	Report enbl	Phase err enbl	Trans err enbl	SCSI Res enbl
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

Bit	Item	When 1	When 0
Bit 0	SCSI Reset Condition	Enable	Disable
Bit 1	Transfer Error	Enable	Disable
Bit 2	Phase Error	Enable	Disable
Bit 3	Report	Enable	Disable
Bit 4	Sel/Reselection	Enable	Disable
Bit 5	Auto Select/Reselect	Enable	Disable
Bit 6	(Reserved)		
Bit 7	All Interrupt Sources	Enable	Disable

Bit 7: All Interrupt Sources

When 1: Depends on setting of bits 5 to 0 and asserts the –INT signal

When 0: All interrupt sources disabled irrespective of bits 5 to 0 and prohibits the –INT signal assertion.

The upper three bit values of the interrupt codes at the interrupt status register (address 04) correspond to the bit values of this register.

Bit #	Interrupt Code			Interrupt Code Type
	IS7	IS6	IS5	
0	0	0	0	SCSI reset
1	0	0	1	Interrupt related to transfer
2	0	1	0	Interrupt related to phase transition
3	0	1	1	Interrupt related to report
4	1	0	0	Interrupt related to selection/reselection
5	1	0	1	Interrupt related to automatic selection/reselection

For description on each interrupt code, see INTERRUPT CODE section. Also, the interrupt is always reported in the interrupt status register (addr 04h) regardless of this register setting.

(11) Group 6/7 Command Length Setting Register(R/W)

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
11	1Ah		2Ah	R/W	Group 7 Command Length				Group 6 Command Length			
					GL7	GL6	GL5	GL4	GL3	GL2	GL1	GL0
Initial Value					‘0’	‘0’	‘0’	‘0’	‘0’	‘0’	‘0’	‘0’

This register specifies the Group 6 and 7 command length.

Setting each value at this register enables the SPC to determine the Group 6 and 7 command length. Set the values listed in table below.

If each value is not set at this register, the COMMAND INVALID interrupt will be reported when the SPC receives the first bytes of Groups 6 and 7.

GL7	GL6	GL5	GL4	Group 7 Command Length
0	0	0	0	Not specified
0	0	0	1	2 bytes
0	0	1	0	4 bytes
0	0	1	1	6 bytes
~		~	~	
~		~	~	
1	1	1	0	28 bytes
1	1	1	1	30 bytes

GL3	GL2	GL1	GL0	Group 6 Command Length
0	0	0	0	Not specified
0	0	0	1	2 bytes
0	0	1	0	4 bytes
0	0	1	1	6 bytes
~		~	~	
~		~	~	
1	1	1	0	28 bytes
1	1	1	1	30 bytes

(12) Response Own ID Display Register (R/W)

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
11	1Bh		2Bh	R/W	'0'	'0'	'0'	'0'	Response Own ID			
									RO3	RO2	RO1	RO0
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

This register indicates the own ID in hexadecimal which responds to the selection/reselection request from a partner device. By reading out this register, you can recognize the ID which responds to the selection/reselection request among IDs specified in the Response Own ID Setting Register.

(13) Response Own ID Setting Register (MSB)

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
11	1Ch		2Ch	R/W	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

(14) Response Own ID Setting Register (LSB)

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
11	1Dh		2Dh	R/W	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

These registers are used to set the own ID to respond to the selection/reselection request from a partner device. The SPC responds for the ID where '1' is set to the corresponding bit. When all bit values are 0, the SPC does not respond to the selection/reselection.

(15) Auto Operation Mode Setting Register

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
11	1Eh		2Eh	R/W	Com. Ignore	Int. Mode	Command Reject Mode CR1	Command Reject Mode CR2	REQ Assert Mask	Com Complete Mask	L2 Int. Mask	Auto ACK Res
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

Bit 7 : Command Receive Invalid/Valid

When 1 : The SPC ignores the command which is received while the interrupt source is held in the Interrupt Status Register. This case, 'Command Reject' interrupt is not reported.

When 0 : The SPC executes the command received.

Bit 6 : INT signal Hold/Not Hold

When 1 : The SPC holds the –INT signal asserted while the Bit 7 of SPC Status Register (INT flag) is '1'. Even if the SPC busy = '1' (the SPC is in operation by the command or auto response operation), the –INT signal is asserted.

When 0 : The SPC negates the –INT signal when the first interrupt status is read out. The –INT signal is negated even if the SPC busy = '1' (the SPC is in operation by the command or auto response operation).

Bit 5 : Command Reject Report / Not Report #1

When 1 : The SPC does not report the 'Command Reject' interrupt for the command received during the SPC Busy condition.

When 0 : The SPC reports the 'Command Reject' interrupt for the command received during the SPC Busy condition. The SPC reports the 'Command Reject' interrupt following a result of a command currently being executed or of the automatic process.

Bit 4 : Command Reject Report / Not Report #2

In case that the SPC responds to the selection/reselection while the SPC is on standby or in execution after receiving the command to select/reselect, this bit functions as follows:

When 1 : The SPC reports only the interrupt related to the Reselected/Selected and does not report the 'Command Reject' interrupt.

When 0 : The SPC reports the interrupt related to the Reselected/Selected following the 'Command Reject' interrupt.

Bit 3 : REQ Asserted Report / Not Report

When 1 : The SPC does not report the 'REQ Asserted' interrupt to inform the REQ signal is asserted.

When 0 : The SPC reports the 'REQ Asserted' interrupt to inform the REQ signal is asserted.

However, the 'REQ Asserted' is reported in the Data phase regardless of this bit status.

Bit 2 : Command Complete Report / Not Report

When 1 : The SPC does not report the 'Command Complete' interrupt for the Set/Reset commands.

When 0 : The SPC reports the Command Complete interrupt for the Set/Reset commands.

The applicable Set/Reset commands are as follows:
– Initiator commands : SET ATN, RESET ATN, SET ACK, RESET ACK
– Target commands : SET REQ, RESET REQ
– Common commands : SET RST, RESET RST

Bit 1: Level-2 Interrupt Report/Not Report

This bit specifies whether or not the SPC reports the level-2 interrupts that occur in the user program operation.

When 1 : Does not report. (Does not hold the interrupt in the Interrupt Status Register.) The result of the command execution is held in the accumulator and another register in the SPC. The value which is held in them is as follows:
1) If Command Complete is reported, 00h is held.
2) If Initial Phase Error or ATN Condition Detected is reported, 01h is held.
The result of this operation can be recognized by the conditional branch instruction that follows.

When 0 : Reports the interrupt code and it's sequence step. In this case, it is necessary to read out and execute the interrupt code and sequence step by the MOVE instruction for every execution of a discrete command.

Bit 0 : Auto ACK Reset / Manual ACK Reset

This bit specifies how to negate the last ACK signal for the transfer command during the Initiator's operation.

When 1 : The SPC negates the last ACK signal during the execution of the transfer command. (This is the auto ACK Reset mode.)

When 0 : The SPC does not negate the last ACK during the execution of the transfer command but, negates by the RESET ACK command. (This is the manual ACK Reset mode.)

1) Operations/Commands that follow the auto ACK Reset mode regardless of this bit setting:
– Send 1–MSG with ATN (19h)
–Send/Receive DATA from MPU/DMA / to MPU/DMA with Padding (14h to 17h).

2) Operations/Commands that follow the manual ACK Reset mode 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 the SPC is in auto reselection response mode

3) Operations/Commands that follow this bit setting:
– All the transfer commands except the commands listed in the above 1) and 2)
– SET ATN (0Ah)
– When ‘Initial Phase Error & Status Received’ interrupt is generated

The value for command step at the auto ACK Reset mode is “08h + value for command step reported at the manual ACK Reset mode”. However, for the commands/operations listed below, the command step at the manual ACK Reset mode is reported.
– SEND/RECEIVE DATA from MPU/DMA / to MPU/DMA with PADDING (14h,15h,16h,17h)
– RECEIVE N-MSG (07h)
–RECEIVE MSG (1Ah)
– RESET ATN (0Bh)
– When ‘Initial Phase Error & MSG Received’ interrupt is generated
– When ‘Initial Phase Error & Status Received’ interrupt is generated
– When the SPC is in auto reselection response mode.

(16) Revision Display Register / SPC Timeout Setting Register

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
XX	1Fh		2Fh	R	Revision Code Display							
					RV7	RV6	RV5	RV4	RV3	RV2	RV1	RV0
				W	SPC Timeout Timer							
					TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0

– Revision Display Register (read) :

This register shows the device revision code of the MB86605.

– SPC Timeout Setting Register (write) :

This register is used to set the SPC timeout time. If the time after receiving command or processing the auto response until the INT flag is set exceds the time specified in this register, the SPC sets the Bit 4 of the SPC Status Register (Timeout flag) and asserts the external –INT signal. The timeout time is shown in the following expression: (Note that the timeout is not detected if '00h' is set to this register.)

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

t_{CLF} : Input clock frequency (1 cycle)
C_{NV} : (Numerical value represented by bits 4 and 3 of clock conversion register)
TO7 to TO0: Set values of bits 7 to 0

Example) : When input frequency 30MHz, internal operating clock frequency 10MHz, (CLK CNV=38h), t_{CLF} = 33.33ns, and C_{NV} = 3;

$33.33 \times 3 \times [(TO7 \times 2^{30}) + (TO6 \times 2^{28}) + (TO5 \times 2^{26}) + (TO4 \times 2^{24}) + (TO3 \times 2^{22}) + (TO2 \times 2^{20}) + (TO1 \times 2^{18}) + (TO0 \times 2^{16})]$.

Consequently, the time range can be set by this register as follows:

(1) Upper limit (TO7 to TO0 = 1) — 143.1 sec

(2) Lower limit (TO7 to TO1 = 0,TO0 = 1) — 6.5 msec

3.6.3. USER PROGRAM MEMORY REGISTER

The user program memory register is initialized by the external reset or Software Reset command.

(1) User Program Memory Control Register

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
10	10h		30h	R	'0'	'0'	'0'	'0'	'0'	'0'	Bank Mode	Bank Select
				W	'0'	'0'	'0'	'0'	'0'	'0'		
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

This register is used to set the user program memory.

Bits 7 to 2 : Unused bits
Write '0' at those bits. Do not write '1'.

Bit 1 : Bank Mode

When 1 : The construction of memory bank is 1KB x 2 banks.

When 0 : The construction of memory bank is 2KB x 1 bank.

Bit 0 : Bank Select

When 1 : Executes the user program in the bank 1. The bank 0 can be accessed via MPU bus.

When 0 : Executes the user program in the bank 0. The bank 1 can be accessed via MPU bus.

Note :

1) Do not access while the SPC is in operation (SPC Busy = '1').
2) Bit 0 is valid only when Bit 1 =1.

(2) Current Address Pointer / Transfer Start Address Register

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
10	12h	13h	33h	R	'0'	'0'	'0'	'0'	'0'	CA10	CA9	CA8
				W	'0'	'0'	'0'	'0'	'0'	TS10	TS9	TS8
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'
10	13h	12h	32h	R	CA7	CA6	CA5	CA4	CA3	CA2	CA1	'0'
				W	TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

– Current Address Pointer (read) :

This register indicates the address which can be accessed from the user program memory port. In the 1KB x 2 banks mode, the upper case bit (CA10) is always '0' regardless of the bank mode.

– Transfer Start Address Register (write) :

This register is used to set the start address where user program memory is sequentially accessed from the user program memory port. In the 1KB x 2 banks mode, always set the upper case bit (TS10) to '0' regardless of the bank mode.

(3) User Program Memory Data Port

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
10	14h		34h	R / W	User Program First Data							
					Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
10	15h		35h		User Program Second Data							
					Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–		36h		User Program Third Data							
					Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–		37h		User Program Fourth Data							
					Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

This register serves the data I/O port used to read/write for the user program memory. Since the sequential access is possible from the address specified in the Transfer Start Address Register, user program data can be transferred from the system side by the burst transfer instructions (I/O strings instructions).

In the 1KB x 2 banks mode, another bank can be accessed through this data port even if the user program is executing (SPC Busy = '1').

The initial value for this data port is undefined.

(4) User Program Operation Start Address Register

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
10	18h	19h	39h	R / W	'0'	'0'	'0'	'0'	'0'	UA10	UA9	UA8
10	19h	18h	38h		UA7	UA6	UA5	UA4	UA3	UA2	UA1	UA0
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

This register is used to set the start address of the user program to operate. In the 1KB x 2 banks mode, always set '0' to the upper case bit (UA10) regardless of the bank mode.

(5) SEL/RESEL Auto Response Start Address Register

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
10	1Ah	1Bh	3Bh	R / W	'0'	'0'	'0'	'0'	'0'	SA10	SA9	SA8
10	1Bh	1Ah	3Ah		SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

This register is used to set the start address of the program that performs the auto response operation for the Selection/Reselection by the user program. In the 1KB x 2 banks mode, set '0' to the upper case bit (SA10) regardless of the bank mode.

3.6.4. 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	PCI									
01	10h		20h	R / W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Initial Value					'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 reversed 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	PCI									
01	11h		21h	R / W	'0'	'0'	'0'	SEL	BSY	MSG	C/D	I/O
Initial Value					'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 resersed 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.

4. COMMAND

4.1. COMMANDS

The SPC commands can be specified for the command register and user program memory.

The commands can be classified into three types according to the setting.

(1) Sequential commands: Continuous sequence (including phase transition) and can be specified only by the command register.
(2) Discrete commands: Discrete commands can be specified for both command register and user program memory. The commands do not have operands when specified for the command register, but have 1 byte operand for the user program memory according to the type of command.
(3) Common commands: Commands that perform the MB86605 common basic operation such as a software reset and diagnostic tests.
(4) Program commands: Can be used only when the device operates by the user program. If the commands are specified for the user program memory, they have 1 byte operand according to the type of commands.

Setting / Command type	Setting to Command Register	Setting to User Program Memory
Sequential commands	○	×
Discrete and Common commands	○	○
Program commands	×	○

(○ : Possible × : Impossible)

4.2. DATA TRANSFER INSTRUCTIONS IN USER PROGRAM OPERATION

Data transfer instructions have 1 byte operand if user program operates. So, the number of byte to be transferred or the user program memory address in which the number of byte and the destination address to transfer are stored should be specified to the operand according to the operation modes. If the bank of destination is different from that of the program, add the command to specify the bank prior to the transfer instruction.

Address	16-Bit Bus Mode (68 series)	16-Bit Bus Mode (80 series)	PCI Bus I/F Mode
N	00h	# of Transfer data (LSB)	# of Transfer data (LSB)
N+1	# of Transfer data (MSB)	# of Transfer data (Mid)	# of Transfer data (Mid)
N+2	# of Transfer data (Mid)	# of Transfer data (MSB)	# of Transfer data (MSB)
N+3	# of Transfer data (LSB)	00h	00h
N+4			Destination Addr (LSB)
N+5			Destination Addr (Mid, L)
N+6			Destination Addr (Mid, H)
N+7			Destination Addr (MSB)

Note on Data Transfer in User Program Operation :

When executing the data transfer instructions in the user program operation, please set up the Data Block Register (addr : 06h & 07h in Basic Control Register Window) before starting the user program operation. Set up value of the data block register is effective for all the transfer instructions during user program operation.

4.3. INITIATOR COMMANDS

	No	Command Code									Operand (When User Program executed)	Command Name
Sequential	1	00H	0	0	0	0	0	0	0	0	—	SELECT and CMD
	2	01H	0	0	0	0	0	0	0	1	—	SELECT and 1-MSG and CMD
	3	02H	0	0	0	0	0	0	1	0	—	SELECT and N-Byte-MSG and CMD
	4	03H	0	0	0	0	0	0	1	1	—	SELECT and 1-MSG
	5	04H	0	0	0	0	0	1	0	0	—	SELECT and N-Byte-MSG
	6	05H	0	0	0	0	0	1	0	1	—	SEND N-Byte-MSG
	7	06H	0	0	0	0	0	1	1	0	—	SEND N-Byte-CMD
	8	07H	0	0	0	0	0	1	1	1	—	RECEIVE N-Byte-MSG
Discrete	9	08H	0	0	0	0	1	0	0	0	—	SELECT
	10	09H	0	0	0	0	1	0	0	1	—	SELECT with ATN
	11	0AH	0	0	0	0	1	0	1	0	—	SET ATN
	12	0BH	0	0	0	0	1	0	1	1	—	RESET ATN
	13	0CH	0	0	0	0	1	1	0	0	—	SET ACK
	14	0DH	0	0	0	0	1	1	0	1	—	RESET ACK
	15	0EH	0	0	0	0	1	1	1	0	Address for the number of transfer	SEND Chain DATA from MPU
	16	0FH	0	0	0	0	1	1	1	1	Address for the number/addr of trs	SEND Chain DATA from DMA
	17	10H	0	0	0	1	0	0	0	0	Address for the number of transfer	SEND DATA from MPU
	18	11H	0	0	0	1	0	0	0	1	Address for the number/addr of trs	SEND DATA from DMA
	19	12H	0	0	0	1	0	0	1	0	Address for the number of transfer	RECEIVE DATA to MPU
	20	13H	0	0	0	1	0	0	1	1	Address for the number/addr of trs	RECEIVE DATA to DMA
	21	14H	0	0	0	1	0	1	0	0	Address for the number of transfer	SEND DATA from MPU (Padding)
	22	15H	0	0	0	1	0	1	0	1	Address for the number/addr of trs	SEND DATA from DMA (Padding)
	23	16H	0	0	0	1	0	1	1	0	Address for the number of transfer	RECEIVE DATA to MPU (Padding)
	24	17H	0	0	0	1	0	1	1	1	Address for the number/addr of trs	RECEIVE DATA to DMA (Padding)
	25	18H	0	0	0	1	1	0	0	0	Address of message to be sent	SEND 1-MSG
	26	19H	0	0	0	1	1	0	0	1	Address of message to be sent	SEND 1-MSG with ATN
	27	1AH	0	0	0	1	1	0	1	0	Address where message written	RECEIVE MSG
	28	1BH	0	0	0	1	1	0	1	1	Address of command to be issued	SEND CMD
	29	1CH	0	0	0	1	1	1	0	0	Address where status written	RECEIVE STATUS
	30	1DH	0	0	0	1	1	1	0	1	_	(Reserved)

4.3.1. INITIATOR SEQUENTIAL COMMANDS

No	Command name	Function	Setup
1	SELECT and CMD	(1) Select. (2) Send CDB*.	• Write target ID to SEL/RESEL ID register. • Write CDB to Send MCS buffer.
2	SELECT and 1-MSG and CMD	(1) Select. (2) Send one message. (3) Send CDB*.	• Write target ID to SEL/RESEL ID register. • Write message and CDB in sequence to Send MCS buffer.
3	SELECT and N-Byte-MSG and CMD	(1) Select. (2) Send N-byte message. (3) Send CDB*.	• Write target ID to SEL/RESEL ID register. • Write message and CDB in sequence to Send MCS buffer. • Write number of bytes for message to MC byte register.
4	SELECT and 1-MSG	(1) Select. (2) Send one message*.	• Write target ID to SEL/RESEL ID register. • Write message to Send MCS buffer.
5	SELECT and N-Byte-MSG	(1) Select. (2) Send N-byte message*. (Used to send two or more messages)	• Write target ID to SEL/RESEL ID register. • Write message to Send MCS buffer. • Write number of bytes for message to MC byte register.
6	SEND and N-Byte-MSG	(1) Send N-byte message*. (Used to send two or more messages)	• Write message to Send MCS buffer. • Write number of bytes for message to MC byte register.
7	SEND and N-Byte-CMD	(1) Send N-byte CDB*. (Used to send undefined CDB)	• Write CDB to Send MCS buffer. • Write number of bytes for CDB to MC byte register.
8	RECEIVE N-Byte-MSG	(1) Receive N-byte message*. (Used to receive message of 33 bytes or more)	• Write number of bytes for message to MC byte register.

CDB: Command Descriptor Block

* The transfer termination is reported with the last ACK signal asserted.
The ACK signal must be negated by the RESET ACK command before issuing the next command. (In case that the Auto ACK Reset flag =‘0’.)

4.3.2. INITIATOR DISCRETE COMMANDS

No	Command Name	Function	Setup
9	SELECTION	Select.	Write ID to SEL/RESEL ID register.
10	SELECT with ATN	Select and assert ATN signal.	Write ID to SEL/RESEL ID register.
11	SET ATN	Assert ATN signal.	None
12	RESET ATN	Negate ATN signal.	None
13	SET ACK	Assert ACK signal.	None
14	RESET ACK	Negate ACK signal.	None
15	SEND Chain DATA from MPU	Send data in program transfer mode*1 *3.	Write number of blocks/bytes for data to be sent to data block register/data byte register.
16	SEND Chain DATA from DMA	Send data in DMA transfer mode*1 *3.	Write number of blocks/bytes for data to be sent to data block register/data byte register.
17	SEND DATA from MPU	Send data in program transfer mode*1.	Write number of blocks/bytes for data to be sent to data block register/data byte register.
18	SEND DATA from DMA	Send data in DMA transfer mode*1.	Write number of blocks/bytes for data to be sent to data block register/data byte register.
19	RECEIVE DATA to MPU	Receive data in program transfer mode*1.	Write number of blocks/bytes for data to be received to data block register/data byte register.
20	SEND DATA to DMA	Receive data in DMA transfer mode*1.	Write number of blocks/bytes for data to be received to data block register/data byte register.
21	SEND DATA from MPU (PADDING)	Send data in program transfer mode as specified by counter and perform padding transfer until target changes phase*2.	Write number of blocks/bytes for data to be sent to data block register/data byte register.
22	SEND DATA from DMA (PADDING)	Send data in DMA transfer mode as specified by counter and perform padding transfer until target changes phase*2.	Write number of blocks/bytes for data to be sent to data block register/data byte register.
23	RECEIVE DATA to MPU (PADDING)	Receive data in program transfer mode as specified by counter and receive no data until target changes phase*2.	Write number of blocks/bytes for data to be received to data block register/data byte register.
24	RECEIVE DATA to DMA (PADDING)	Receive data in DMA transfer mode as specified by counter and receive no data until target changes phase*2.	Write number of blocks/bytes for data to be received to data block register/data byte register.
25	SEND 1-MSG	Send one message*1.	Write message to Send MCS buffer.
26	SEND 1-MSG with ATN	Keep asserting ATN signal and send one message.	Write message to Send MCS buffer.
27	RECEIVE MSG	Receive one message*1.	None
28	SEND CMD	Send one CDB*1.	Write CDB to Send MCS buffer.
29	RECEIVE STATUS	Receive one status*1.	None

CDB: Command Descriptor Block

*1 The transfer termination is reported with the last ACK signal asserted. The ACK signal must be negated by the RESET ACK command before issuing the next command. (In case that the auto ACK Reset flag = '0'.)

*2 Padding transfer

Transfer of empty data is called padding transfer.

When output --- Send data with all bits set to 0.

When input --- Do not receive data.

- Padding transfer start and end

  When the number of bytes transfered equals the number of data bytes specified by the data block and data byte registers, the SPC initiates the padding transfer process. Note that when the SPC performs synchronous transfer in the SCSI Input mode an Empty flag may not be set in the data register since padded data corresponding to number of REQ signals are latched. (The padded bytes have been received after the completion of the true data transfer.) If the initial values of the data block and data byte registers are 0, the padding transfer is performed from the first byte. In this case, the data block, data byte, and modified-byte registers hold the values before the start of padding transfer.

  The padding transfer is held until the target changes the phase.

*3 When issuing two or more data transfer commands (N commands) in one data phase, use the Chain Command for the 1 to N–1 commands and the normal command for the last Nth command. (The MB86605 support the Chain command for SEND only, and it is not necessary to use it for Receive.)

The Chain Command does not transfer the last 1 byte when the transferred byte is odd number, and the last data will be transferred by the next transfer command as the first data. If the transferred byte data is always even number, it is not necessary to use the Chain Command.

Please note that the Chain Command can be used for only the variable-length data transfer.

4.4. TARGET COMMANDS

	No	Command Code									Operand (When User Program executed)	Command Name
Sequential	1	20H	0	0	1	0	0	0	0	0	—	RESELECT and 1-MSG
	2	21H	0	0	1	0	0	0	0	1	—	RESELECT and N-Byte-MSG
	3	22H	0	0	1	0	0	0	1	0	— —	RESE- LECT and 1-MSG and TER- MINATE
	4	23H	0	0	1	0	0	0	1	1	— —	RESELECT and 1-MSG and LINK TERMINATE
	5	24H	0	0	1	0	0	1	0	0	—	TERMINATE
	6	25H	0	0	1	0	0	1	0	1	—	LINK TERMINATE
	7	26H	0	0	1	0	0	1	1	0	—	DISCONNECT SEQUENCE
	8	27H	0	0	1	0	0	1	1	1	—	SEND N-Byte-MSG
	9	28H	0	0	1	0	1	0	0	0	—	RECEIVE N-Byte-CMD
	10	29H	0	0	1	0	1	0	0	1	—	RECEIVE N-Byte-MSG
	11	2AH	0	0	1	0	1	0	1	0	—	RESELECT and N-Byte-MSG and TERMINATE
	12	2BH	0	0	1	0	1	0	1	1	—	RESELECT and N-Byte-MSG and LINK TERMINATE
	13	2CH	0	0	1	0	1	1	0	0	—	DISCONNECT SEQUENCE 2
	14	2DH	0	0	1	0	1	1	0	1	—	(Reserved)
	15	2EH	0	0	1	0	1	1	1	0	—	(Reserved)
	16	2FH	0	0	1	0	1	1	1	1	—	(Reserved)
Discrete	17	30H	0	0	1	1	0	0	0	0	—	RESELECT
	18	31H	0	0	1	1	0	0	0	1	—	SET REQ
	19	32H	0	0	1	1	0	0	1	0	—	RESET REQ
	20	33H	0	0	1	1	0	0	1	1	—	DISCONNECT
	21	34H	0	0	1	1	0	1	0	0	Address for number of transfer	SEND DATA from MPU
	22	35H	0	0	1	1	0	1	0	1	Addr for number/addr of transfer	SEND DATA from DMA
	23	36H	0	0	1	1	0	1	1	0	Address for number of transfer	RECEIVE DATA to MPU
	24	37H	0	0	1	1	0	1	1	1	Addr for number/addr of transfer	RECEIVE DATA to DMA
	25	38H	0	0	1	1	1	0	0	0	Address of message to be sent	SEND 1-MSG
	26	39H	0	0	1	1	1	0	0	1	Address where message written	RECEIVE MSG
	27	3AH	0	0	1	1	1	0	1	0	Address of status to be sent	SEND STATUS
	28	3BH	0	0	1	1	1	0	1	1	Address where CDB written	RECEIVE CMD
	29	3CH	0	0	1	1	1	1	0	0	—	(Reserved)
	30	3DH	0	0	1	1	1	1	0	1	—	(Reserved)
	31	3EH	0	0	1	1	1	1	1	0	Address for number of transfer	SEND Chain DATA from MPU
	32	3FH	0	0	1	1	1	1	1	1	Addr for number/addr of transfer	SEND Chain DATA from DMA

CDB: Command Descriptor Block

4.4.1. TARGET SEQUENTIAL COMMANDS

No	Command Name	Function	Setup
1	RESELECT and 1-MSG	(1) Reselect. (2) Switch message-in phase. (3) Send one message.	• Write initiator ID to SEL/RESEL ID register. • Write message to Send MCS buffer.
2	RESELECT and N-Byte-MSG	(1) Reselect. (2) Switch to message-in phase. (3) Send N-byte message.	• Write initiator ID to SEL/RESEL ID register. • Write message to Send MCS buffer. • Write number of bytes for message to MC byte register.
3	RESELECT and 1-MSG and TERMINATE	(1) Reselect. (2) Switch to message-in phase. (3) Send one message. (4) Execute TERMINATE sequence.	• Write initiator ID to SEL/RESEL ID register. • Write message, status (1 byte), and message (1 byte) in sequence to Send MCS buffer.
4	RESELECT and 1-MSG and LINK TERMINATE	(1) Reselect. (2) Switch to message-in phase. (3) Send one message. (4) Execute LINK TERMINATE sequence.	• Write initiator ID to SEL/RESEL ID register. • Write message, status (1 byte), and message (1 byte) in sequence to Send MCS buffer.
5	TERMINATE	(1) Reselect. (2) Send one-byte status. (3) Switch to message-in phase. (4) Send one-byte message. (5) Disconnect.	• Write status (1 byte) and message (1 byte) in sequence to Send MCS buffer.
6	LINK TERMINATE	(1) Switch to status phase. (2) Send one-byte status. (3) Switch to message-in phase. (4) Send one-byte message.	• Write status (1 byte) and message (1 byte) in sequence to Send MCS buffer.
7	DISCONNECT SEQUENCE	(1) Switch to message-in phase. (2) Send two-byte message. (3) Disconnect.	• Write message (2 byte) to Send MCS buffer.
8	SEND N-Byte-MSG	(1) Switch to message-in phase. (2) Send N-byte message. (Used to send two or more messages)	• Write message to Send MCS buffer. • Write number of bytes for message to MC byte register.
9	RECEIVE N-Byte-MSG	(1) Switch to command phase. (2) Receive N-byte CDB. (Used to receive undefined CDB)	• Write number of bytes for CDB to MC byte register.
10	RECEIVE N-Byte-MSG	(1) Switch to message-out phase. (2) Receive N-byte message. (Used to receive message of 33 bytes or more)	• Write number of bytes for message to MC byte register.
11	RESELECT and N-Byte-MSG and TERMINATE	(1) Reselect. (2) Switch to message-in phase. (3) Send N-byte message. (4) Execute TERMINATE sequence.	• Write initiator ID to SEL/RESEL ID register. • Write message, status (1 byte), and message (1 byte) in sequence to Send MCS buffer. Write number of bytes for first message to MC byte register.
12	RESELECT and N-Byte-MSG and LINK TERMINATE	(1) Reselect. (2) Switch to message-in phase. (3) Send N-byte message. (4) Execute LINK TERMINATE sequence.	• Write initiator ID to SEL/RESEL ID register. • Write message, status (1 byte), and message (1 byte) in sequence to Send MCS buffer. Write number of bytes for first message to MC byte register.
13	DISCONNECT SEQUENCE 2	(1) Switch to message-in phase. (2) Send one-byte message. (3) Disconnect.	• Write message (1 byte) to Send MCS buffer.

CDB: Command Descriptor Block

4.4.2. TARGET DISCRETE COMMANDS

No	Command Name	Function	Setup
14	RESELECT	Reselect.	• Write initiator ID to 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	(1)Switch to data-in phase. (2) Send data in program transfer mode.	• Write number of blocks/bytes for data to be sent to data block register/data byte register.
19	SEND DATA from DMA	(1) Switch to data-in phase. (2) Send data in DMA transfer mode.	• Write number of blocks/bytes for data to be sent to data block register/data byte register.
20	RECEIVE DATA to MPU	(1) Switch to data-out phase. (2) Receive data in program transfer mode.	• Write number of blocks/bytes for data to be received to data block register/data byte register.
21	RECEIVE DATA to DMA	(1) Switch to data-out phase. (2) Receive data in DMA transfer mode.	• Write number of blocks/bytes for data to be received to data block register/data byte register.
22	SEND 1-MSG	(1) Switch to message-in phase. (2) Send one message.	• Write message to Send MCS buffer.
23	RECEIVE MSG	(1) Switch to message-out phase. (2) Receive one message.	• None
24	SEND STATUS	(1) Switch to status phase. (2) Send one-byte status.	• Write status (1 byte) to Send MCS buffer.
25	RECEIVE CMD	(1) Switch to command phase. (2) Receive CDB.	• None
26	SEND Chain DATA from MPU	(1)Switch to data-in phase. *1 (2) Send data in program transfer mode.	• Write number of blocks/bytes for data to be sent to data block register/data byte register.
27	SEND Chain DATA from DMA	(1) Switch to data-in phase. *1 (2) Send data in DMA transfer mode.	• Write number of blocks/bytes for data to be sent to data block register/data byte register.

CDB: Command Descriptor Block

*1 : When issuing two or more data transfer commands (N commands) in one data phase, use the Chain Command for the 1 to N–1 commands and the normal command for the last Nth command. (The MB86605 support the Chain command for SEND only, and it is not necessary to use it for Receive.)

The Chain Command does not transfer the last 1 byte when the transferred byte is odd number, and the last data byte will be transferred by the next transfer command as the first data byte. If the transferred byte data is always an even number, it is not necessary to use the Chain Command.

Please note that the Chain Command can be used for only the variable-length data transfer only.

4.5. COMMON COMMANDS

No	Command Code									Operand (When user program executed)	Command Name
1	40H	0	1	0	0	0	0	0	0	–	SOFTWARE RESET
2	41H	0	1	0	0	0	0	0	1	–	TRANSFER RESET
3	42H	0	1	0	0	0	0	1	0	–	SCSI RESET
4	43H	0	1	0	0	0	0	1	1	–	(Reserved)
5	44H	0	1	0	0	0	1	0	0	–	INIT DIAG START
6	45H	0	1	0	0	0	1	0	1	–	TARG DIAG START
7	46H	0	1	0	0	0	1	1	0	–	DIAG END
8	47H	0	1	0	0	0	1	1	1	–	COMMAND PAUSE
9	48H	0	1	0	0	1	0	0	0	–	SET RST
10	49H	0	1	0	0	1	0	0	1	–	RESET RST

4.5.1. COMMON COMMANDS

No	Command Code	Function	Setup
1	SOFTWARE RESET (Note 1)	(1) Reset internal circuit in SPC. (2) Set state of internal circuit in SPC from Reset to Ready.	None
2	TRANSFER RESET	Reset hardware for transfer of internal circuit in SPC.	None
3	SCSI RESET	Send RST signal which is SCSI bus signal.	None
4	INIT DIAG START (Note 2)	The SPC itself can perform self-diagnosis to simulate transfer as the initiator.	Write ‘07h’ to the Own ID Setting Register.
5	TARG DIAG START (Note 2)	The SPC itself can perform self-diagnosis to simulate transfer as the target.	Write ‘07h’ to the Own ID Setting Register.
6	DIAG END (Note 2)	Set mode from self-diagnostic to normal.	None
7	COMMAND PAUSE (Note 3)	(1) Forcibly terminate transfer during execu tion of data phase. (2) Stop selection/reselection.	None
8	SET RST	Assert the RST signal of SCSI bus signals and holds it.	None
9	RESET RST	Negate the RST signal asserted by the SET RST command.	None

Notes:

1. If the **SOFTWARE RESET** command is issued, the Command Complete interrupt will not occur and the SPC will enter the fully-stopped state.
   To clear the Software Reset state, reissue the **SOFTWARE RESET** command (the SPC enters the Ready

state). Also, more than $4t_{CLF}$ must be taken between sequential **SOFTWARE RESET** commands.

2. Self-diagnostic function

Issuing the **INIT DIAG START** or **TARG DIAG START** command causes the SPC to perform self-diagnosis.
Self-diagnosis is made for the following items and the results are reported.

- Arbitration sequence
- Selection/reselection sequence

Some commands allow the SPC to continuously simulate transfer as the initiator or target (for details, see 5.9).

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

3. **COMMAND PAUSE** command

(1) Data phase termination:

- SCSI-OUTPUT

  In the DMA transfer mode, output of the DREQ signal is stopped immediately. In the program transfer mode, 0 is indicated at the DATA REQ bit (bit 2 of the SPC status register).
  Transfer (data output) to the SCSI is continued until the data register becomes empty.

- SCSI-INPUT

  The new output of the REQ or ACK signal to the SCSI bus is stopped.
  Transfer on the system side (DMA or program transfer) is continued until the data register becomes empty.
  The target has completed REQ or ACK handshaking. However, the initiator may have received an unnecessary REQ signal.

(2) Selection/reselection can be stopped only before executing the selection/reselection phase. If the SPC has started selection/reselection, the **COMMAND PAUSE** command becomes invalid and selection/reselection is executed.

(3) If the **COMMAND PAUSE** command has been received and completed, the COMMAND PAUSE interrupt is reported.

4.6. PROGRAM COMMANDS

Command Code									Operand)	Command Name
5XH	0	1	0	1	Partner ID				—	SELECT ID (Initiator command)
6XH	0	1	1	0	Partner ID				—	SELECT ID with ATN (Initiator command)
7XH	0	1	1	1	Partner ID				—	RESELECT ID (Target command)
80H	1	0	0	0	0	0	0	0	—	NOP
90H	1	0	0	1	0	0	0	0	User Status Code	STOP
91H	1	0	0	1	0	0	0	1	User Status Code	STOP with INTERRUPT
AXH	1	0	1	0	Mode/Register set				Address/Immediate Data	AND
BXH	1	0	1	1	Mode/Register set				Address/Immediate Data	AND TEST
CXH	1	1	0	0	Mode/Register set				Address/Immediate Data	COMPARE
DXH	1	1	0	1	0	0	Condition		Jump Address	CONDITION JUMP
EXH	1	1	1	0	Mode/Register set				Address/Immediate Data	MOVE
FXH	1	1	1	1	0	Address segment			—	ADDRESS SEGMENT SET

4.7. PROGRAMMABLE COMMANDS

The programmable commands can be used in the only user program. The programmable commands are classified into discrete commands and special commands.

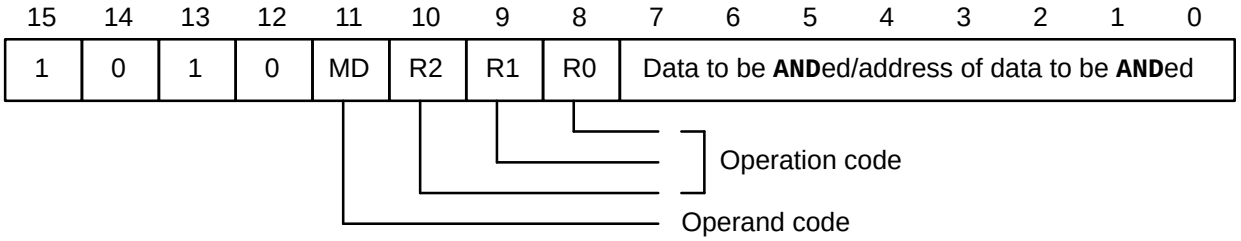
4.7.1. DISCRETE COMMANDS

No.	Command Name	Function	Setup
1	SELECT ID	Selects. (Initiator command)	None
2	SELECT ID with ATN	Selects and asserts ATN signal. (Initiator command)	None
3	RESELECT ID	Reselects. (Target command)	None

4.7.2. SPECIAL COMMANDS

Special commands are used for logical operations and conditional branch etc. in the user program.

(1) AND Command



Load the value specified by the operation code to the accumulator and carry out the logical AND between the values indicated by the accumulator and the operand. The results are stored in the accumulator.

Bit 11: Operand code

Bit 11 specifies function of operand

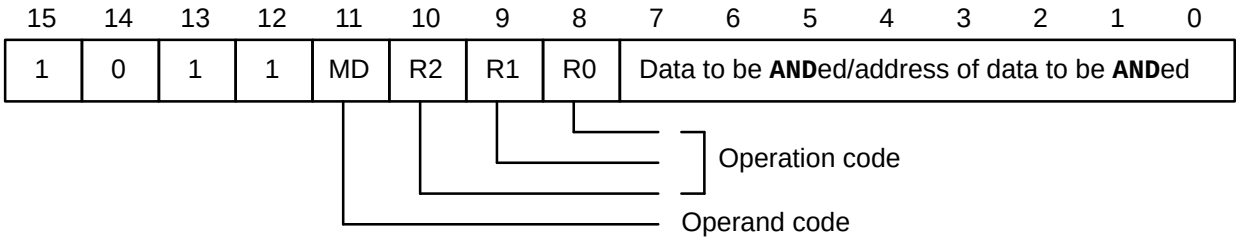
When 1: Specify address with data to be ANDed at operand.
When 0: Specify data to be ANDed at operand.

Bits 10 to 8: Operation code

Bits 10 to 8 specify the register for the AND operation as illustrated in the following table:

R2	R1	R0	AND
0	0	0	First byte received in message, command and status phases
0	0	1	Second byte received in message, command and status phases
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

(2) TEST AND Command



Load the value specified by the operation code to the accumulator and carry out the logical AND between the values indicated by the accumulator and operand. The results are stored in the SPC and can be used for the conditional branch command. The value in the accumulator is unchanged.

Bit 11: Operand code

Bit 11 specifies the function of the operand.

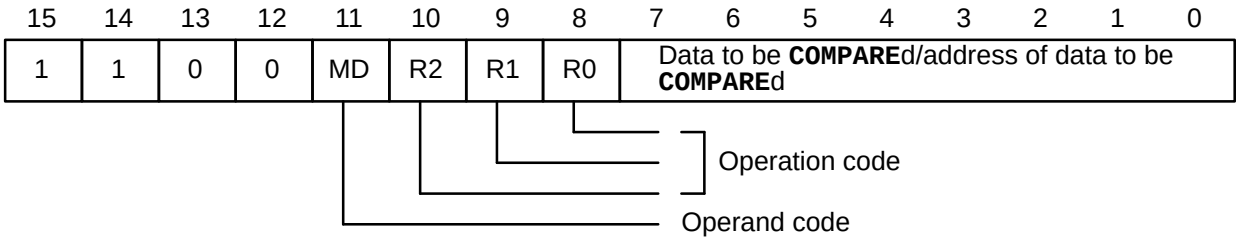
When 1: Specify address with data to be ANDed at operand.
When 0: Specify data to be ANDed at operand.

Bits 10 to 8: Operation code

Bits 10 to 8 specify the register for the AND operation and are listed in the table below.

R2	R1	R0	AND
0	0	0	First byte received in message, command and status phases
0	0	1	Second byte received in message, command and status phases
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

(3) COMPARE Command



Load the value specified by the operation code to the accumulator and carry out the comparison operation between the values indicated by the accumulator and operand. The results are stored in the SPC and can be used for the conditional branch command. The value in the accumulator is unchanged.

Bit 11: Operand code

Bit 11 specifies the function of the operand.

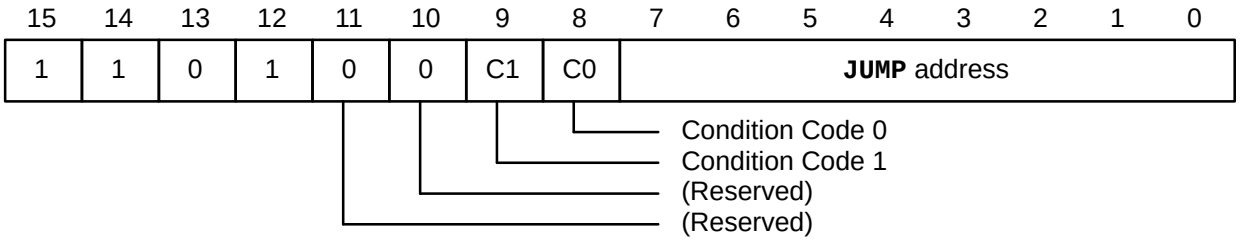
When 1: Specify address with data to be compared at operand.
When 0: Specify data to be compared at operand.

Bits 10 to 8: Operation code

Bits 10 to 8 specify the register for the comparison operation and are listed in the table below.

R2	R1	R0	Compare
0	0	0	First byte received in message, command and status phases
0	0	1	Second byte received in message, command and status phases
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) Conditional Branch Command



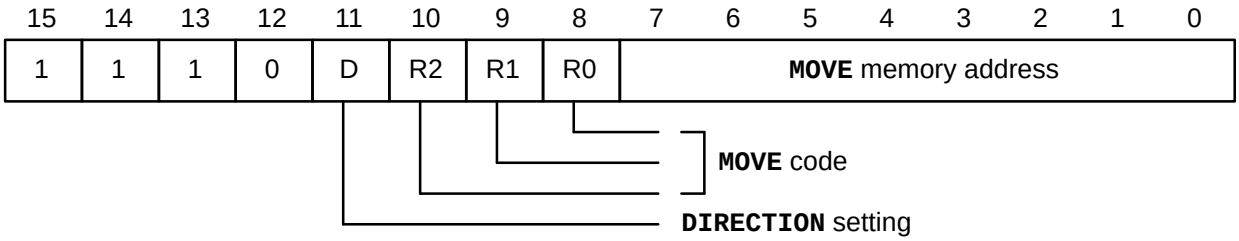
This command is used for conditional or unconditional branches, using the operation results obtained by executing the **COMPARE** and **TEST AND** commands. Even if the SPC is in Level-2 Interrupt Not Report Mode (Bit 1 of Auto Response Mode Setting Register = '1'), the result of the command execution can be found by this conditional branch since the result of command execution is held in the SPC.

Bits 9 and 8: Condition code

Bits 9 and 8 specify the branch condition.

R2	R1	Operation
0	1	When operation result ≠ 0, transfer execution control to address specified at operand; when operation result = 0, go to next command (conditional branch).
1	0	When operation result = 0, transfer execution control to address specified at operand; when operation result ≠ 0, go to next command (conditional branch).
0	0	Unconditionally transfer execution control to address specified at operand (unconditional branch).
1	1	Reserved

(5) MOVE Command



This command copies the value specified by the **MOVE** code to the memory address specified at the operand or the value of the memory address specified at the operand to the accumulator.

Bit 11: **DIRECTION** setting

Bit 11 specifies the **MOVE** direction.

When 1: Copy value specified by **MOVE** code to memory address specified at operand (register → memory).
When 0: Copy value of memory address specified at operand to accumulator (memory → accumulator).

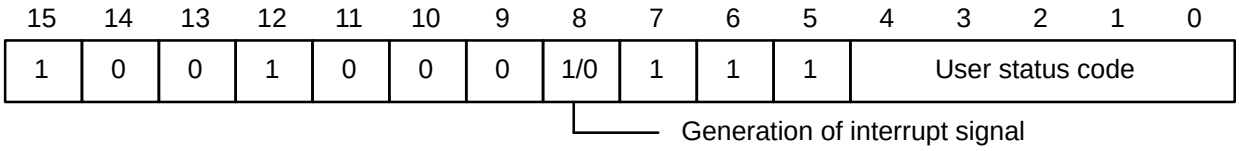
Bits 10 to 8: **MOVE** code

Bits 10 to 8 specify the register for **MOVE** and are listed in the table below.

R2	R1	R0	MOVE
0	0	0	First byte received in message, command and status phases
0	0	1	Second byte received in message, command and status phases
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

(6) STOP Command

This command causes the SPC to stop the user program operation.
When execution of this command is terminated, the SPC is brought from user program operation into normal operation (enters Ready state and waits to receive command from command register).



Bit 8: Generation of interrupt signal

Bit 8 specifies generation of an interrupt signal (INT signal) to the host MPU.

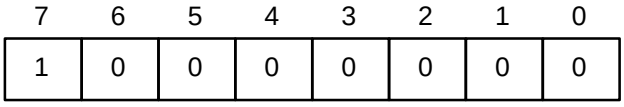
When 1: Generate interrupt signal (INT signal).
When 0: Do not generate interrupt signal (INT signal).

Bit 7 (interrupt requested) of the SPC status register indicates 1 irrespective of the setting of this bit when an interrupt request is made from the SPC to the host MPU.

Bits 4 to 0: User status code

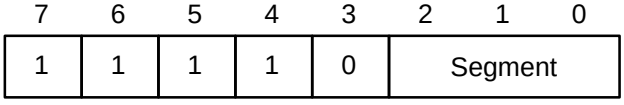
Bits 4 to 0 specify the user status code.
Bits 7 to 5 are preset to 1 and are invalid.
There are no interrupt status codes preset at the SPC which can set all upper three bits to 1. Therefore, if such interrupt status code is reported, the user program operation is terminated.

(7) NOP Command

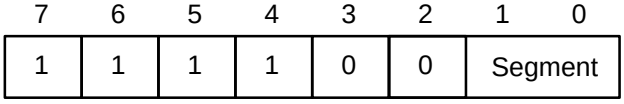


This **NO OPERATION** command causes the SPC to proceed to the next step without any execution.

(8) ADDRESS SEGMENT SET Command



: When User Program Memory is in 2KB x 1 bank mode.



: When User Program Memory is in 1KB x 2 banks mode.

Since Special Commands have 8-bit operand to indicate the user program address, they cannot address the entire memory space of 2K bytes. By combining with this command to specify the upper 3 bits of the operand address, the entire memory space can be addressed.

The upper 3 bits address specified by this command is valid until another value is specified. Also, the initial value for the upper address is the same as one specified in the user program operation start address register.

Example) :

Make an AND operation between a data in address 255h and a data in the accumulator by the AND command in address 0C0h, and jump to address 100h if the result of the AND operation is '00h'.

Address	Command Code	
0BFh	F2h	Address BANK SET TO 2XXh
0C0h	AFh	AND ACC, (55h)
0C1h	55h	
0C2h	F1h	ADDRESS BANK SET TO 1Xh
0C3h	D2h	JUMP TO 00h if 0
	00h	
100h	XXh	Command at jumped

Also, when transfer command is executed in the user program, the transfer byte number and the transfer address (only for PCI bus interface mode) are specified by indirect addressing. If the byte number and transfer address data are assigned in the different segment, specify the segment by the ADDRESS SEGMENT SET Command prior to the transfer command.

4.8. PRECAUTIONS FOR CREATING USER PROGRAM

When creating a user program, take care about the following points.

(1) Precautions for using **AND**, **TEST AND**, **COMPARE** or **MOVE** command

Both the interrupt status register and command step register specified by the **AND**, **TEST AND**, **COMPARE** or **MOVE** command (register → memory only) are FIFO registers and indicate the next interrupt status and command step if they hold more than one interrupt source.

When executing the **AND** command several times, the contents of the registers must be saved by the **MOVE** command (register → memory). In this case, the **AND** command can be executed more than once by using the **MOVE** command (memory → accumulator).

When executing the **TEST AND** or **COMPARE** command several times, specify the register as the accumulator from the second execution.

Execute the **TEST AND** or **COMPARE** command after loading the contents of the specified register to the accumulator. The contents of the accumulator are unchanged.

(2) Automatic mode in user program operation

The automatic receive mode is invalid.

The automatic selection/reselection response mode is valid and precedes the user program operation.

If the SPC is selected or reselected during execution of the discrete commands including SELECT/RESELECT in the user program, it stops the user program operation to go to the automatic selection/reselection response mode. At this point, the user program is ignored.

Even if the automatic selection/reselection mode is not set, the SPC, when selected or reselected, stops the user program operation and responds to a selection/reselection request.

(3) Interrupt in user program operation

If an interrupt source occurs due to an error in the user program operation, the SPC operates differently depending on the interrupt source level.

Level 1: SPC stops user program operation, generates interrupt to host MPU and enters Ready state

Level 2: SPC continues user program operation :

When Bit 1 of Auto Operation Mode Setting Register (Level 2 interrupt not report/report) = '0': Interrupt status and command step are held by the SPC and so, they can be referred by commands.

When Bit 1 of Auto Operation Mode Setting Register (Level 2 interrupt not report/report) = '1': Interrupt status and command step are not held by the SPC, but the result of the command execution is held as well as the result of the operation and so, the result can be found by the conditional branch command.

The interrupts corresponding to level 2 are as follows. Others correspond to level 1.

Interrupt Type	Operation Mode	Description of Interrupt
REPORT	INIT/TARG	Normal command termination
	TARG	ATN condition detected
Phase error	INIT	Initial phase error

5 OPERATION

5.1. SPC INITIALIZATION

The SPC is reset by the following methods.

- RESET pin input = Low (Hardware reset)
- Issuing **SOFTWARE RESET** command (Software reset)
- Issuing **TRANSFER RESET** command (Transfer reset)
- SCSI reset condition (Reset condition)

Hardware reset is always be made at power-on.

(1) Hardware reset

All control circuits are reset.

During hardware reset, writing to the internal registers is disabled.

The MB86605 enters the software reset state after released from the hardware reset.
At the power-on of the SPC, perform the hardware reset.

(2) Software reset

All control circuits are reset.

During software reset, writing commands to the BASIC Control Register are disabled.

Like the hardware reset, the values of the BASIC Control register and SCAM Register are initialized except the Bits 2,1, and 0 of the I/O Control Register and Window Address Register that hold the previous values. The values for the Initial Setting Register set before issuing the **SOFTWARE RESET** command are held, and they can be rewritten during software reset.

The values of the MCS buffer and user program memory are held without being initialized.

(3) Transfer reset

The data transfer circuits in the SPC are reset.

The connection state of the SCSI is unchanged. Execution of the selection/reselection phase is not affected.

With the SPC operating as the initiator, if the **TRANSFER RESET** command is issued concurrently with the start of target data transfer (input), only the first byte (only the first and second bytes in the 16-bit SCSI bus mode) can be received (applies only in asynchronous transfer mode). Internal Registers are not initialized by this reset.

(4) Reset condition

When the SPC is connected, all outputs of the SCSI pin are stopped. The values of the internal registers are not initialized.

5.2. SCSI DATA BUS PHASE

5.2.1. SELECTION FROM OTHER BUS DEVICES

When selected or reselected from another bus devices, the SPC automatically executes the response sequence providing the following response conditions are satisfied.

(1) Selection phase response

A. Bit 6 (selection response) of the response mode setting register is 1.

B. The ID of the SCSI data bus satisfies the following conditions.

- The bit set at the own ID setting register is true.
- Bit 1 (single initiator option) of the response mode setting register is 1 and the true bit is 1 bit.
- Bit 1 (single initiator option) of the response mode setting register is 0 and the true bit is 2 bits.
- The parity bit is legal (a parity check is always made irrespective of the setting of bit 7 of the parity error detection setting register).

(2) Reselection phase response

A. Bit 7 (reselection response) of the response mode setting register is 1.

B. The ID of the SCSI data bus satisfies the following conditions.

- The bit set at the own ID setting register is true.
- The true bit is 2 bits.
- The parity bit is legal (a parity check is always made irrespective of the setting of bit 7 of the parity error detection setting register).

5.2.2. OPERATION AFTER RESPONSE

There are the following three operations after selection or reselection response.

(1) The SELECTED, SELECTED with ATN, or RESELECTED interrupt is reported immediately after selection or reselection response.

(2) The automatic selection response mode or automatic reselection response mode preset at the SPC is executed.

Automatic selection response mode — SEL/RESEL mode setting register: bit 7 = 0, bit 5 = 1
Automatic reselection response mode — SEL/RESEL mode setting register: bit 7 = 0, bit 6 = 1

(3) The user program operation is performed.
1) Set the program start address to the SEL/RESEL Auto Response Start Address Register.
2) Write ‘1’ at the Bit 7 of SEL/RESEL mode setting register.

Note: For a program example, see User Program Operation section.

5.2.3. EXECUTION OF SELECTION/RESELECTION PHASE

The sequence from arbitration to selection/reselection is performed by issuing the command including selection/reselection. When the SPC loses with the arbitration, it will participate in the arbitration again at the detection of the next bus-free phase. This operation will be executed until the SPC wins the arbitration. However, if the SPC is selected by another device that won the arbitration, it stops the command execution and enters the response operation. Also, the SPC can execute the SCSI operation from the selection phase without the arbitration phase.

If the operation after response (Section 5.2.2) is specified as automatic selection/reselection mode or user program operation, the SPC may be already in operation when the command including selection/reselection is received.

To issue the command including selection/reselection, make sure that bit 6 (BUSY state) of the SPC status register is 0. In case that selection/reselection timeout is detected in the selection/reselection phase, the SPC performs the retry as set in the SEL/RESEL Retry Setting Register (addr : 14h). The retry is performed from the arbitration phase.

5.2.4. EXECUTION OF MESSAGE, COMMAND, AND STATUS PHASES

The MCS buffer should be used to execute the message, command, and status phases.

(1) Sending

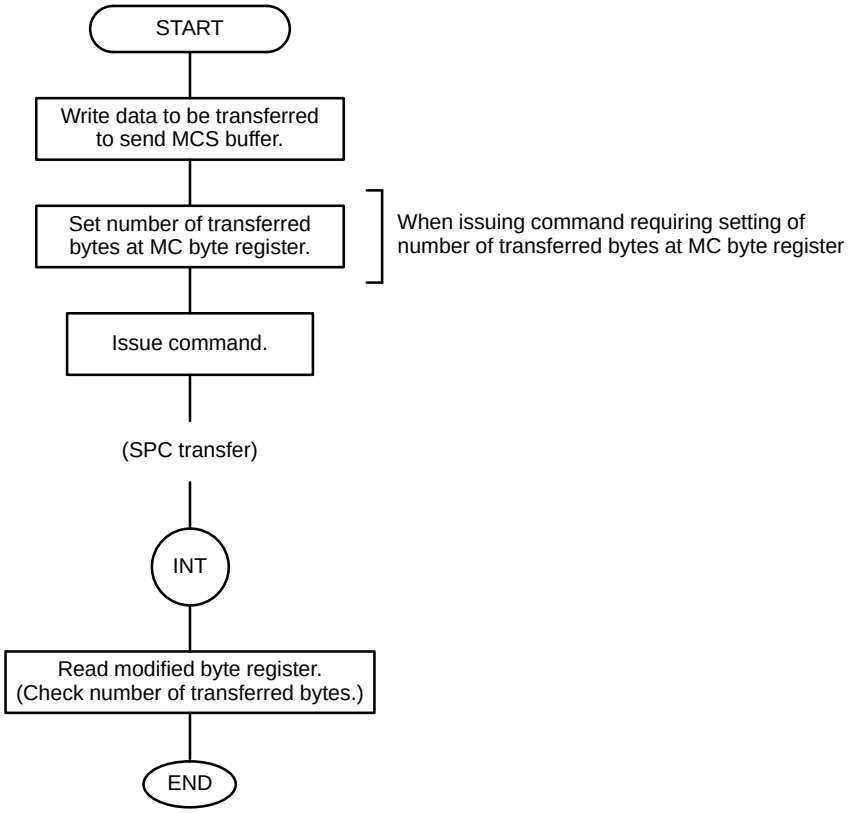
- Write data to be sent from address 0 of the send MCS buffer and issue the command.
- To issue a command requiring setting of the number of bytes transferred, preset the number of bytes transferred at the MC byte register.
- Do not write data to the send MCS buffer until the SPC reports termination after issuing the command.
- After the end of transfer, indicate the number of transferred bytes at the modified byte register. If transfer is terminated with an error, indicate the number of bytes transferred normally.

(2) Receiving

- If previously-executed phases other than the data phase are received, read data from the receive MCS buffer and issue the command.
- To issue a command requiring setting of the number of bytes transferred, preset the number of bytes transferred at the MC byte register.
- Do not read data from the receive MCS buffer until the SPC reports termination after issuing the command.
- After the end of transfer, indicate the number of transferred bytes at the modified byte register. If transfer is terminated with an error, indicate the number of bytes transferred normally.
- After the end of transfer, read the received data from address 0 of the receive MCS buffer.

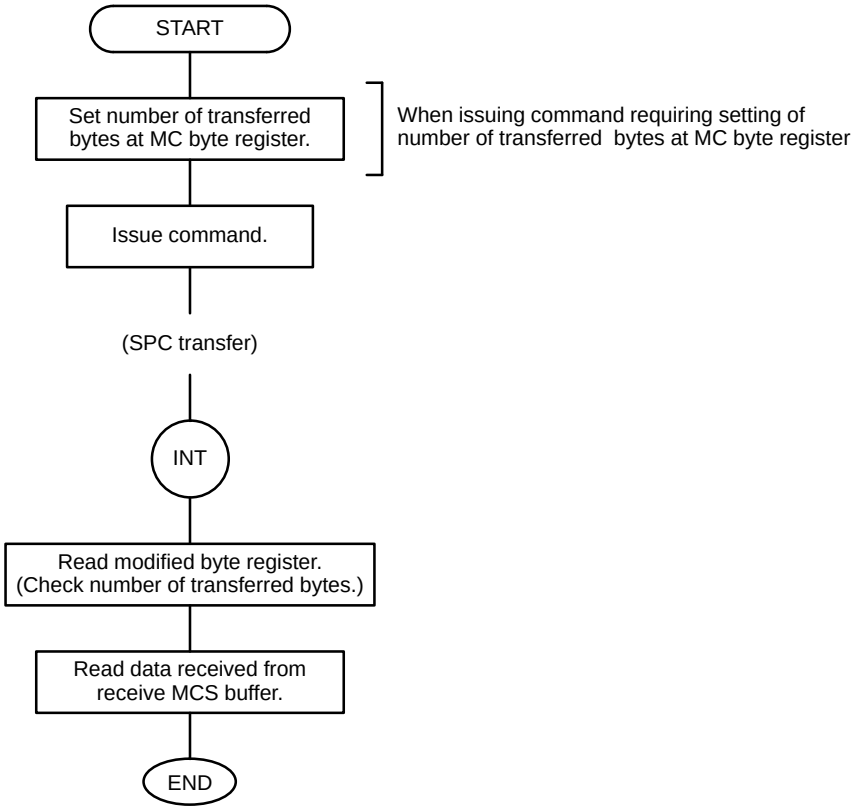
a. Flowchart for sending message, command, and status phases

* Perform the N-byte transfer after setting the number of bytes transferred at the MC byte register.



b. Flowchart for receiving message, command, and status phases

* Perform the N-byte transfer after setting the number of transferred bytes at the MC byte register.



5.2.5. DATA PHASE

There are the following transfer modes for data transfer in the data phase.

Mode	Function
Program transfer mode	The MPU program uses the data register (address 00/01) to set up send data and read receive data. The DMA interface is not used.
DMA transfer mode	In this mode, the SPC executes all the sequences for data transfer. Send data is set up and receive data is read via the DMA interface.

(1) Program transfer mode

When using the program transfer mode, perform the processing while checking the state of the SPC with the SPC status register.

- Do not access the data register until bit 2 (DATA TRANS REQ) of the SPC status register goes to 1.
- Do not access the data register while bit 2 (DATA TRANS REQ) of the SPC status register is 0.
- The termination of transfer can be checked with bit 6 (BUSY state).

(2) DMA transfer mode

In the DMA transfer mode, only burst transfer is supported. In this transfer, the DREQ signal is always active as long as the data register is in the Ready state.

- Do not activate the $\overline{\text{DACK}}$, $\overline{\text{IORD}}$, or $\overline{\text{IOWR}}$ signal until the DREQ signal goes High after issuing the DMA transfer command (while the DREQ signal is Low).
- If the **COMMAND PAUSE** command is issued at output (when the SPC is operating as a target), the DREQ signal goes Low out of synchronization with the $\overline{\text{DACK}}$, $\overline{\text{IORD}}$, or $\overline{\text{IOWR}}$ signal.

Note: When executing the data phase, take care about the following points.

- Always set the transfer parameters (transfer mode, transfer period, and transfer offset) for the synchronous transfer mode before issuing the **SEND-DATA** and **RECEIVE-DATA** commands. Also, it is not necessary to set the transfer parameters for the 2nd or subsequent transfers once it is set at the 1st transfer. The SPC stores the transfer parameters for each partner ID and automatically sets them.
- If the data-in phase is established when the SPC is operating as the initiator and before the data block register and data byte register are set by the host MPU and the command is issued, receive data from the target as follows. Note that the ACK signal is not asserted.

  Asynchronous: 1 byte (2 bytes if the SCSI bus is a 16-bit width)
  Synchronous: Number of bytes for received REQ signal

- In the MB86605, two or more transfer commands can be issued in one data phase. In case of SEND DATA command, use “SEND Chain DATA” command for 1st to (N–1) data, and use “SEND DATA” command for the last Nth data as usual. (No need to use the “Chain” command for RECEIVE operation and when the byte data to be sent is always even numbers.)
- In the data transfer with system side (both in program and DMA transfers), transfer only the numbers of data as set in the Transfer Byte Counter (for fixed length data, the number of transfer block x the number of transfer byte), and do not transfer unnecessary data such as a dummy data. Especially, in case that the number of transfer byte is odd, a byte access is needed for the last data.

- In case of fixed length data transfer (block transfer), the block length can be set in only even number (in word boundary). If the block length is set in odd number, data will be corrupt at the block boundary. In case that the block transfer should be performed in odd number, issue the transfer commands of variable length (odd number) for the numbers of block to be transferred or set the total number of transfer as the variable data transfer since two or more commands can be issued in one data phase in the MB86605. Also, during one data phase, do not issue the fixed length data transfer (block transfer) command after the variable length data transfer command for the odd number byte.

5.3. INTERRUPT CODES

Parameter		INIT or TARG	AUTO1	AUTO2	Interrupt Code								Interrupt Source
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 and MPU parity error
					0	0	1	0	0	1	0	1	SCSI and 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 and 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 and MSG received
			Y		0	1	0	1	0	1	1	0	Initial phase-error and status received
			Y		0	1	0	1	1	1	0	1	Initial Phase-error and MSG not received
			Y		0	1	0	1	1	1	1	0	Initial Phase-error and status not 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 and MSG received)
			Y		0	1	0	0	0	1	1	1	Command stop (ATN condition detected and MSG not 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	1	COMMAND PAUSE
					0	1	1	0	1	0	0	0	Diagnostic Result : “Good”
					0	1	1	0	1	0	0	1	Diagnostic Result : “No-good”
					0	1	1	0	1	1	0	0	Command Rejected (Receive MCS buffer not ready)
		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)

AUTO1: Automatic receive mode

AUTO2: Automatic selection/reselection response mode

Y = Defined, N = Undefined

Parameter		INIT or TARG	AUTO1	AUTO2	Interrupt Code								Interrupt Source
4	REPORT	T	Y		0	1	1	0	0	0	1	0	Command complete (ATN condition detected and MSG received
			Y		0	1	1	0	0	1	1	0	Command complete (ATN condition detected and MSG not 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
		I			1	0	0	1	0	1	0	0	Reselected & Receive MCS buffer not ready
		T		N	1	0	0	0	0	0	0	0	Selected
				N	1	0	0	0	0	0	0	1	Selected with ATN
					1	0	0	0	0	1	0	0	Selected & Receive MCS buffer not ready
					1	0	0	0	0	1	0	1	Selected with ATN & Receive MCS buffer not ready
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 and MSG–received
				Y	1	0	1	0	1	0	0	1	CDB received with identify–MSG and MSG–received
			N	Y	1	0	1	0	1	0	1	0	CDB received (ATN condition detected)

5.3.1 INTERRUPT RELATED TO SCSI RESET

Interrupt Code	Interrupt Source	Explanation
01h	RESET-condition detected	Reset state detected on SCSI bus

5.3.2 INTERRUPT RELATED TO TRANSFER

Interrupt Code	Interrupt Source	Explanation
24h	SCSI parity error	In currently-executing phase, parity error detected in input/output data over SCSI interface
22h	MPU parity error	1. During program transfer (data phase), parity error detected in input/output data over MPU interface 2. Parity error is detected in the register accessed by the host MPU.
21h	DMA parity error	During DMA transfer (data phase), parity error detected in input/output data over DMA interface
26h	SCSI and MPU parity error	During program transfer (data phase), parity error detected in input/output data over MPU and SCSI interfaces
25h	SCSI and DMA parity error	During DMA transfer (data phase), parity error detected in input/output data over DMA and SCSI interfaces
29h	Offset error	REQ/ACK signal greater than offset value set in synchronous transfer received
2Ah	Short transfer period error	SPC could not follow because period of REQ/ACK signal received during synchronous transfer short
2Bh	Offset error and Short transfer period error	**Offset error** and **Short transfer period** error detected during synchronous transfer
2Ch	REQ/ACK timeout	More than specified REQ/ACK timeout time elapsed
31h	Disconnected in transfer progress	Bus Free state detected during phase execution (transfer)
32h	Phase error in transfer progress	Phase error occurred during phase execution (transfer)

5.3.3 INTERRUPT RELATED TO PHASE TRANSITION

Interrupt Code	Interrupt Source	Explanation
54h	Initial phase-error (*1)	Phase for receive command disagreed when first REQ signal received
55h	Initial phase-error and MSG received (* 2)	Initial phase error occurred and one message received in response to message receive request from target
56h	Initial phase-error and STATUS received (* 2)	Initial phase error occurred and one message received in response to status receive request from target
5Dh	Initial phase-error and MSG not received (*2)	Initial phase error occurred, and intends to receive one message in response to message receive request from target but could not receive since data has been stored in Receive MCS buffer. (*4)
5Eh	Initial phase-error and STATUS not rceived(*2)	Initial phase error occurred, and intends to receive one status in response to status receive request from target but could not receive since data has been stored in Receive MCS buffer. (*4)
42h	Command stop (ATN condition detected) (*3)	Attention condition generated by initiator to disallow phase transition. Or, during execution of data phase, transfer stopped at transfer block boundary. (At this time, the data register is empty.)
43h	Command stop (ATN condition detected) and MSG received (*2)	Attention condition generated by initiator to disallow phase transition and one message received in response to message-out phase request from initiator. Or, during execution of data phase, one message received in response to request at transfer block boundary. (At this time, the data register is empty.)
47h	Command Stop (ATN condition detected) and MSG not received (*2)	Attention condition generated by initiator to disallow phase transition and intends to receive one message in response to message-out phase request from initiator, or during execution of data phase, intends to receive one message in response to request at transfer block boundary. But could not receive since data has been stored in the Receive MCS buffer. (At this time, the data register is empty.)

* Notes:

1. If a phase error occurs and the data-in phase is requested, the SPC operates as follows:
   (1) When nexused target in asynchronous mode
   Receive one-byte data.
   (2) When nexused target in synchronous mode
   Receive number of bytes for received REQ signal (ACK signal not asserted)
2. Only when automatic receive mode set
3. Only when automatic receive mode not set
4. Modified-byte counter is automatically cleared.

5.3.4 INTERRUPT RELATED TO REPORT

Interrupt Code	Interrupt Source	Explanation
60h	Command complete	Execution of received command terminated normally
64h	Command rejected	(1) Command received when SPC in automatic operation mode (auto selection/reselection response mode) (2) New command received during execution of command (3) Message of 33 bytes or more when **SEND MSG** command for extended message received (Note 1).
65h	Command invalid	(1) Command from target received when operating as initiator (2) Command from initiator received when operating as target (3) Sequential command in user program memory (4) Received command code undefined (5) Register value 0 although command requiring setting of data block register/data byte register (MC byte counter) received (6) Transfer impossible since length of group 6/7 CDB set to 0 (Note 2) (7) Group 3/4 CDB or transfer command was received. (Note 2) (8) Command starting at transfer phase received in unnexused state (9) Command starting at selection/reselection phase received in nexused state
67h	Command Pause	(1) **COMMAND PAUSE** command received and terminated (2) Received message extended message of 33 bytes or more (Note 3)
68h	Result of self-diagnosis **Good**	Self-diagnosis performed with good result
69h	Result of self-diagnosis **No Good**	Self-diagnosis performed with no-good result
6Ch	Command Rejected (Receive MCS buffer not ready)	Message, command status Receive Command received or intended to start the user program operation while previous data is still in Receive MCS buffer. (Receive operation or user program operation can not be done while previous data is in MCS buffer.)
70h	Disconnected	Bus free state detected during nexus as initiator and detected at other than during transfer
71h	REQ Asserted (Note 4)	Request for execution of next phase made from target. This interrupt is reported only when the command is not received during nexus.
61h	Command complete (ATN condition detected) (Note 5)	Command terminated normally and attention condition generated by initiator detected. The ATN signal is detected only when it is not asserted with the ACK signal in the last phase for the asserted executed command.
62h	Command complete (ATN condition detected) and MSG received (Note 6)	Command terminated normally and one message received in response to attention condition generated by initiator. The ATN signal is detected only when it is not asserted with the ACK signal in the last phase for the asserted executed command.

Interrupt Code	Interrupt Source	Explanation
66h	Command complete (ATN condition detected) and MSG not received (Note 6)	Command completed normally and intended to receive one message in response to attention condition generated by initiator. But, could not receive since data is still in Receive MCS buffer. The ATN signal is detected only when it is not asserted with the ACK signal in the last phase for the asserted executed command. (Note 7)

Notes:

1. Transfer a message of 33 bytes or more in length as follows:
   - Initiator

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

     (COMMAND COMPLETE) ←
   - Target

     SEND-N-Byte-MSG → SEND N-Byte-MSG
2. Transfer is not performed at issuing.
   At receiving, this interrupt is reported after writing the first byte to the receive MCS buffer. At user program operation, write the first byte at the specified user program address.
3. This interrupt occurs in receiving.
   The interrupt is reported to the receive MCS buffer after the first/second byte of the received message. At user program operation, write the first/second byte at the specified user program address. The remaining bytes should be received by the **`RECEIVE N-Byte-MSG`** command.
4. If the data-in phase is required, the SPC operates as follows:
   (1) When nexused target in asynchronous mode
   Receive one-byte data.
   (2) When nexused target in synchronous mode
   Receive number of bytes for received REQ signal.
5. Only when automatic receive mode not set
6. Only when automatic receive mode set
   When the attention condition is continuously detected, for a command that is terminated in the message-out phase, the Command Complete (ATN Condition Detected) interrupt is reported instead of this interrupt (without receiving message).
7. Modified-byte counter is automatically cleared.

5.3.5 INTERRUPT RELATED TO SELECTION/RESELECTION

Interrupt Code	Interrupt Source	Explanation
82h	Selection/Reselection timeout	More than specified selection/reselection timeout time elapsed
90h	Reselected (Note 1)	SPC reselected as initiator by target
94h	Reselected and Receive MCS buffer not ready	Reselected at the reselection response mode by the auto reselection response mode or user program operation. But, could not operate response since data is still in Receive MCS buffer. (Note 3)

Interrupt Code	Interrupt Source	Explanation
80h	Selected (Note 2)	SPC selected as target by initiator
81h	Selected with ATN (Note 2)	SPC selected as target by initiator and attention condition generated by initiator detected
84h	Selected and Receive MCS buffer not ready	Selected at the selection response mode by the auto selection response mode or user program operation. But, could not operate response since data is still in Receive MCS buffer (Note 3)
85h	Selected with ATN and Receive MCS buffer not ready	Selected and detected attention condition at the selection response mode by the auto selection response mode or user program operation. But, could not operate response since data is still in Receive MCS buffer (Note 3)

Notes:

1. Only when automatic reselection response mode not set
2. Only when automatic selection response mode not set
3. Modified-byte counter is automatically cleared.

5.3.6 INTERRUPT RELATED TO AUTOMATIC SELECTION/RESELECTION RESPONSE

Interrupt Code	Interrupt Source	Explanation
B0h	AUTO mode phase error	After reselection phase, SPC switched to one other than message-in phase
B1h	MSG received	After reselection phase, one message received
A0h	AUTO mode phase error	ATN signal not asserted in selection phase. This interrupt is reported only when the message-out phase is set after the selection phase. (Note 1)
A1h	MSG received	After selection phase, message other than **identify-MSG** received
A2h	Illegal Identify-MSG received	After selection phase, **identify-MSG** received but 1 set at bits 5, 4, and 3
A3h	MSG received with Identify-MSG	After selection phase, **identify-MSG** and one message received
A4h	Unknown-CDB-received	After selection phase, attempt made to receive one command but only one byte received because length of group 6/7 CDB unknown. (Note 2) This interrupt is reported only when the message phase or command phase is set after the selection phase. (Note 1)
A5h	Unknown-CDB-received with Identify-MSG	After selection phase, attempt made to receive **identify-MSG** and one command but only one byte received because length of group 6/7 CDB unknown (Note 2)
A6h	CMD-received	After selection phase, one command received. This interrupt is reported only when the message phase or command phase is set after the selection phase. (Note 1)
A7h	CMD-received with Identify-MSG	After selection phase, **identify-MSG** and one command received
A8h	CMD received and MSG received	After selection phase, one command received, and then one message received in response to attention condition. This interrupt is reported only when the message phase or command phase is set after the selection phase. (Note 1)
A9h	CMD received with Identify-MSG and MSG received	After selection phase, **identify-MSG** and one command received, and then one message received in response to attention condition
AAh	CMD received (ATN condition detected)	At selection phase, one command received since ATN signal not asserted. After that, detected attention condition. This interrupt is for only the setting message or command phase and not setting auto receive mode. (Note 1.)

Notes:

1. Set the phase transition after the selection phase to the SEL/RESEL mode setting register.
2. Set the length of group 6/7 CDB to the group 6/7 command length setting register.

5.4. INTERRUPT CODE AND COMMAND STEP

When reporting an interrupt, the SPC indicates the interrupt code representing the interrupt status and the command step representing the operating condition of the SPC when the interrupt occurred. The command step varies with the SPC operation mode.

There are the following four SPC operating conditions.

(1) Execution of discrete/sequential command
(2) User program operation
(3) Automatic selection response
(4) SPC in ready state (before receiving command)

When analyzing the interrupt, reference the interrupt status register (address 04) and command step register (address 05).

(1) Execution of discrete/sequential command

The command step specified for each command is indicated at bits 3 to 0 of the command step register.

(Interrupt status) (Command step)

I7	I6	I5	I4	I3	I2	I1	I0	0	0	0	0	S3	S2	S1	S0

(2) User program operation

The number of discrete commands executed (terminated command) is indicated at bits 7 to 4 of the command step register and the command step where the interrupt occurs is indicated at bits 3 to 0.

(Interrupt status) (Command step)

I7	I6	I5	I4	I3	I2	I1	I0	S7	S6	S5	S4	S3	S2	S1	S0

Number of commands executed

(3) Automatic selection response

The response sequence type is indicated at bits 7 to 4 of the command step register and the sequence step where the interrupt occurs is indicated at bits 3 to 0.

(Interrupt status) (Command step)

I7	I6	I5	I4	I3	I2	I1	I0	S7	S6	S5	S4	S3	S2	S1	S0

Sequence code

(4) SPC in ready state

The command step indicates 00H.

(Interrupt status) (Command step)

I7	I6	I5	I4	I3	I2	I1	I0	0	0	0	0	0	0	0	0

The corresponding interrupts are as follows:

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

5.5. OPERATION WHEN ERROR DETECTED

5.5.1. OFFSET ERROR DETECTED

(1) Initiator (detected in Data-In/-Out Phase)

Assert the ATN signal concurrently with the detection of the offset error. Transfer is continued but normal operation cannot be assured.

A termination report is made as follows:

(1) When the transfer is terminated or the target switched to the message-out phase before the transfer termination, report the Offset Error interrupt.
(2) When the target is switched to a phase other than the message-out phase before the transfer termination, report the Offset Error and Phase Error in Transfer Progress interrupts.

(2) Target (detected in Data-In/-Out Phase)

Perform the exit processing to report the Offset Error interrupt concurrently with the detection of the offset error. For the exit processing, see section 5.7.

Notes on Offset Error Detected :

Please note that the transfer is not terminated when the following phenomenon occurs in Data-In Phase at Initiator operation :

(Phenomenon) :

Data are received by the SPC whereas the internal data register has been in Full state because a Target sent the REQ signals exceeding the specified offset value. This time, the SPC detects the offset error and asserts the ATN signal.

(Countermeasure) :

Please recognize that the above phenomenon occurred with the description (1) below, and then take the (2) to (5) steps as the countermeasure :

(1) Wait for Twait (*). Although Twait has passed, Data Request bit (Bit 2) of the SPC Status Register (addr 02h) did not become “1” in the Program transfer mode or DREQ signal was not asserted in the DMA transfer mode. This case, please recognize that the above phenomenon is occurring.

(*) Twait = (Transfer Cycle) x (Block Length) + (Internal Operating Cycle) x (Block #) x 60

Example :

Data transfer rate : 10MByte/sec (i.e. 1 cycle = 100ns), block length = 512 bytes, internal operating frequency : 10MHz (i.e. 1 cycle = 100ns), and block # = 10 :

Twait = 100 x 512 x + 100 x 10 x 60 = 111200[ns] = 111.2[µs].

(2) Stop the data transfer with Command Pause command.

(3) This time, Data Request bit may become “1” or DREQ signal may be asserted again. Perform the data transfer.

(4) Read out the interrupt. “Command Pause” (67h) and “Offset Error”(29h) will be read out.

(5) Issue the SET ACK and RESET ACK command repeatedly until the Target changes the phase.

5.5.2. SHORT TRANSFER PERIOD DETECTED

(1) Initiator (detected in Data-In/-Out Phase)

Assert the ATN signal concurrently with the detection of the short transfer period error. Transfer is continued but normal operation cannot be assured.

A termination report is made as follows:

(1) When transfer is terminated or the target is switched to the message-out phase before the transfer termination, report the Short Transfer Period Error interrupt.
(2) When the target is switched to a phase other than the message-out phase before the transfer termination, report the Short Transfer Period Error and Phase Error in Transfer Progress interrupts.

(2) Target (detected in Data-In/-Out Phase)

Perform the exit processing to report the Short Transfer Period Error interrupt concurrently with the detection of the short transfer period error. For the exit processing, see 5.7.

5.5.3. REQ/ACK TIMEOUT ERROR DETECTED

(1) Initiator (detected in Data-In/-Out Phase)

Stop transfer to report the REQ/ACK Timeout Error interrupt concurrently with the detection of the REQ/ACK timeout error. Upon receipt of the REQ/ACK Timeout Error interrupt, issue the **SCSI RESET** command.

(2) Target

Stop transfer to report the REQ/ACK Timeout Error interrupt concurrently with the detection of the REQ/ACK timeout error. Upon receipt of the REQ/ACK Timeout Error interrupt, issue the **SCSI RESET** or **DISCONNECT** command.

Note : When REQ/ACK Timeout Error is detected while data are being received, the SPC may receive an excess data (1 byte) in the MCS buffer compared with the modified-byte counter value. So, please read out the MCS buffer data when REQ/ACK Timeout Error is detected while data are received.

5.5.4. SCSI PARITY ERROR DETECTED

When the SCSI parity error is detected, the SPC operates differently according to the detected phase.

(1) Initiator

Phase detected	SPC Operation
Message in	Assert the ATN signal concurrently with the detection of a parity error. However, varies varies with the detect point. (1) When the received message is a one-byte or two-byte message, report the SCSI Parity Error interrupt after transfer. (2) When the received message is an extended message, report the SCSI Parity Error interrupt after receiving two bytes when detected at the first or second byte, and after receiving all bytes when detected at the third and later bytes.
Status	Assert the ATN signal to report the SCSI Parity Error interrupt after transfer concurrently with the detection of a parity error.
Data in/out	Assert the ATN signal to report as follows concurrently with the detection of a parity error. (1) When transfer is terminated or the target is switched to the message-out phase before transfer, report the SCSI Parity Error interrupt. (2) When the target is switched to a phase other than the message-out phase before transfer, report the SCSI Parity Error and Phase Error in Transfer Progress interrupts.

(2) Target

Phase detected	SPC Operation
Message out	Stop transfer to report the SCSI Parity Error interrupt concurrently with the detection of a parity error
Status	Stop transfer to report the SCSI Parity Error interrupt concurrently with the detection of a parity error.
Data in/out	The SPC operation after the SCSI parity error depends on the Response Operation Mode Setting Register (Addr 12h). (1) When set to byte-by-byte stop Perform exit process (*1) to report the SCSI Parity Error interrupt concurrently with the detection of a parity error. (2) When set to block-by-block stop As soon as transfer of the block where a parity error is detected is terminated, stop transfer to report the SCSI Parity Error interrupt (terminated with the data register empty).

(*1) : See Section 5.6 Exit Processing.

5.5.5. MPU PARITY ERROR DETECTED

(1) Initiator

Phase detected	SPC Operation
Before execution (write to register)	Report the MPU Parity DETECTED interrupt concurrently with the SPC transfer start (with the receipt of command) after the detection of a parity error.
Data in/out (program transfer)	Assert the ATN signal to report as follows concurrently with the detection of a parity error. (1) When transfer is terminated or the target is switched to the message-out phase before transfer, report the MPU Parity Error interrupt. (2) When the target is switched to a phase other than the message-out phase before transfer, report the MPU Parity Error and Phase Error in Transfer Progress interrupts.

(2) Target

Phase detected	SPC Operation
Before execution (write to register)	Report the MPU Parity Error DETECTED interrupt concurrently with the detection of a parity error. This interrupt occurs when the SPC starts operation (when the command is received to reference the register to which a MPU writes data).
Data in/out (program transfer)	When a parity error is detected, there are the following two types of operations. (1) When set to byte-by-byte stop Perform the exit process to report the MPU Parity Error interrupt concurrently with the detection of a parity error. (2) When set to block-by-block stop As soon as the transfer of the block where a parity error is detected is terminated, stop transfer to report the MPU Parity Error interrupt (terminated with the data register empty).

Note: For the setting of byte-by-byte stop and block-by-block stop, see description of Bit 4 of Response Operation Mode Setting Register. Also, for the exit process, see section 5.6.

5.5.6. DMA PARITY ERROR DETECTED

(1) Initiator

Phase detected	SPC Operation
Data in/out (DMA transfer)	Assert the ATN signal as follows concurrently with the detection of a parity error. (1) When transfer is terminated or the target is switched to the message-out phase before transfer, report the DMA Parity Error interrupt. (2) When the target is switched to a phase other than the message-out phase before transfer, report the DMA Parity Error and Phase Error in Transfer Progress interrupts.

(2) Target

Phase detected	SPC Operation
Data in/out (DMA transfer)	When a parity error is detected, there are the following two types of operations. (1) When set to byte-by-byte stop Perform the exit process to report the DMA Parity Error interrupt concurrently with the detection of a parity error. (2) When set to block-by-block stop As soon as the transfer of the block at which a parity error is detected is terminated, stop transfer to report the DMA Parity Error interrupt (terminated with the data register empty).

Note: For the setting of byte-by-byte stop and block-by-block stop, see description of Bit 4 of Response Operation Mode Setting Register. Also, for the exit process, see section 5.6.

5.6. EXIT PROCESSING

This section describes the SPC exit processing for error involving the exit processing.

5.6.1. WHEN ERROR IS DETECTED IN THE SCSI INTERFACE

- System I/F —In program transfer, keep the transfer until DATA TRANS REQ bit (Bit 2 of SPC Status Register) indicates '0'. In DMA transfer, keep the transfer until DREQ signal falls to "L".

  In the SCSI output operation, terminate storing data in the data register. In the SCSI input operation, keep the transfer until data register becomes empty.

- SCSI I/F —Just after detecting an error, stop the output of ACK or REQ signal.

The following are the error occurred in the SCSI interface, as described in Section 5.6. :
"Offset Error", "Short Transfer Period Error", "SCSI Parity Error". In addition, the following error also involve the exit process regardless of the operation mode and setting : "Phase Error in Transfer Progress", "Disconnect in Transfer Progress", "RST Condition Detected"

5.6.2. WHEN ERROR IS DETECTED IN THE SYSTEM INTERFACE

- System I/F —Just after detecting an error, in program transfer, set '0' to DATA TRANS REQ bit (Bit 2 of SPC Status Register), and in DMA transfer, fall the DREQ signal to "L".

- SCSI I/F— In the SCSI input operation, stop the output of ACK or REQ signal immediately. (In this case, terminate storing data in the data register.) In the SCSI output operation, keep the transfer until the data register becomes empty.

The following are the error occurred in the System interface:
"MPU Parity Error", "DMA Parity Error"

5.6.3. WHEN "REQ/ACK TIMEOUT" IS DETECTED

- System I/F —Just after detecting an error, in program transfer, set '0' to DATA TRANS REQ bit (Bit 2 of SPC Status Register), and in DMA transfer, fall the DREQ signal to "L".

- SCSI I/F Stop the output of ACK or REQ signal immediately.

5.6.4. WHEN "COMMAND PAUSE" IS DETECTED

- System I/F —In program transfer, maintain the transfer until DATA TRANS REQ bit (Bit 2 of SPC Status Register) indicates '0'. In DMA transfer, keep the transfer until DREQ signal falls to "L".

  In the SCSI input operation, maintain the transfer until data register becomes empty. In the SCSI output operation, indicate 0 in DATA TRANS REQ bit immediately or set the DREQ signal to "L".

- SCSI I/F —In the SCSI input operation, stop the output of ACK or REQ signal. In the SCSI output operation, maintain the transfer until data register becomes empty.

5.6.5. CALCULATION OFABANDONED BYTES IN DATA REGISTER

When an error occurs, data transfer may stop leaving data in the data register and on the SCSI bus. For these cases, the bytes that will be abandoned can be calculated as follows:

(1) SCSI-INPUT

Nin = (Value of lower 7 bits of modified byte register) - (Value of lower 7 bits of data byte register)

When Nin ≥ 0, remaining byte count N_{DREG}=Nin
When Nin < 0, remaining byte count N_{DREG}= Nin+128

(2) SCSI-OUTPUT

Nout = (Value of lower 7 bits of data byte register) - (Value of lower 7 bits of modified byte register)

When Nout ≥ 0, remaining byte count N_{DREG}= Nout
When Nin < 0, remaining byte count N_{DREG}= Nout+128

Note :

When SPC terminates on condition that data register stores odd byte data, the SPC outputs 1byte invalid data at the last word access. In this case, the relationship between values for data byte register and modified byte register is as follows:

(Value for data byte register) + 1 = (Value for modified byte register)

When SPC terminates in the PCI bus interface mode, it may cease to transfer the remaining data in the burst transfer buffer of the PCI interface block. In such a case, the bytes remaining in the burst transfer buffer can be calculated as follows:

(1) SCSI-INPUT

Nin = (Value of PCI modified byte register) - (Value of lower 8 bits of data byte register)

When Nin ≥ 0, remaining byte count N_{PCIB}=Nin–N_{DREG}
When Nin < 0, remaining byte count N_{PCIB}= Nin+256–N_{DREG}

(2) SCSI-OUTPUT

Nout = (Value of lower 8 bits of data byte register) - (Value of PCI modified byte register)

When Nout ≥ 0, remaining byte count N_{PCIB}=Nout–N_{DREG}
When Nout < 0, remaining byte count N_{PCIB}= Nout+256–N_{DREG}

5.7. DIAGNOSTIC MODE

When the **DIAG START** command is issued, the SPC automatically starts self-diagnosis and enters the diagnostic mode. The SPC diagnoses the following conditions and reports the results upon completion.

- Self-diagnosed items (1) Arbitration sequence
  (2) Selection/reselection sequence

  After the interrupt is reported as the result of diagnosis, the SPC enters the diagnostic mode.

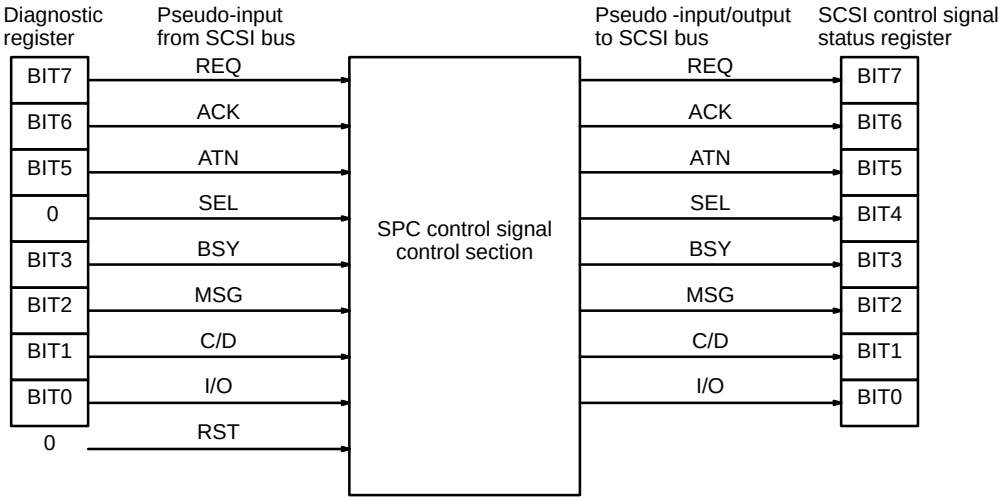
In the self-diagnostic mode, the SPC is disconnected from the SCSI bus so it does not send signals to the SCSI bus nor does it receive control signals from the SCSI bus. However, the SPC is ready for internal control so that it enables pseudo-interface with the SCSI bus.

To terminate the diagnostic mode, issue the **DIAG END** command.

5.7.1. SCSI Control Signals

In the diagnostic mode, each bit of the diagnostic register (address 11) is regarded as input signals to the SPC. Output signals from the SPC are indicated at the SCSI control signal status register (address 11).

Pseudo-input from the SCSI bus and pseudo-output to the SCSI bus are indicated at the SCSI control signal status register.



5.7.2. SCSI DATA BUS SIGNALS

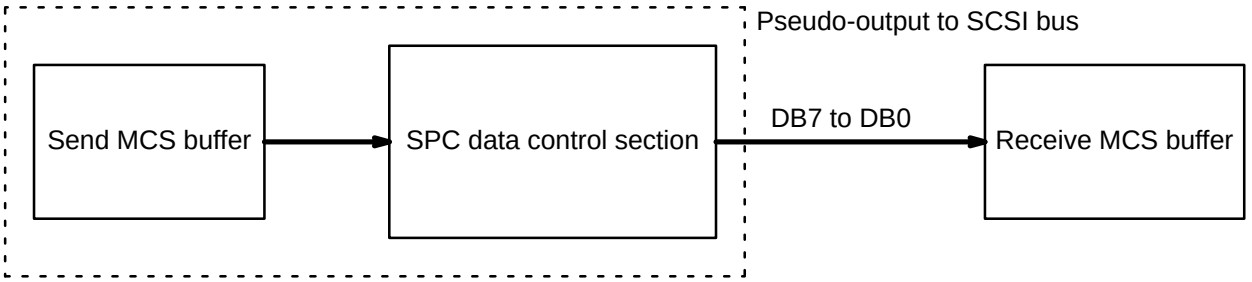
In the diagnostic mode, the pseudo-input/output data to/from the SCSI data bus can be set. The setting of pseudo-input/output data varies with the phase to be executed.

In the diagnostic mode, parity-bit input from the SCSI bus is always regarded as 0.

(1) Sending in message, command, and status phases

As in the normal mode, write the send data to the send MCS buffer and then issue the command.

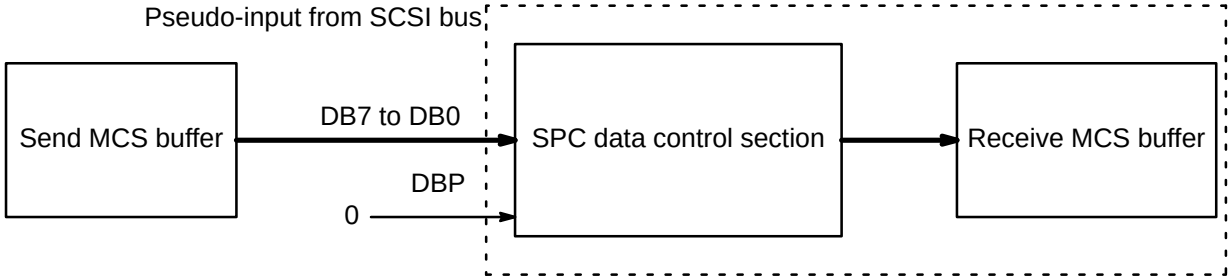
Store the pseudo-output data to the SCSI bus in the receive MCS buffer.



(2) Receiving in message, command, and status phases

Write the pseudo-input data from the SCSI bus to the send MCS buffer and then issue the command.

As in the normal mode, store the receive data in the receive MCS buffer.



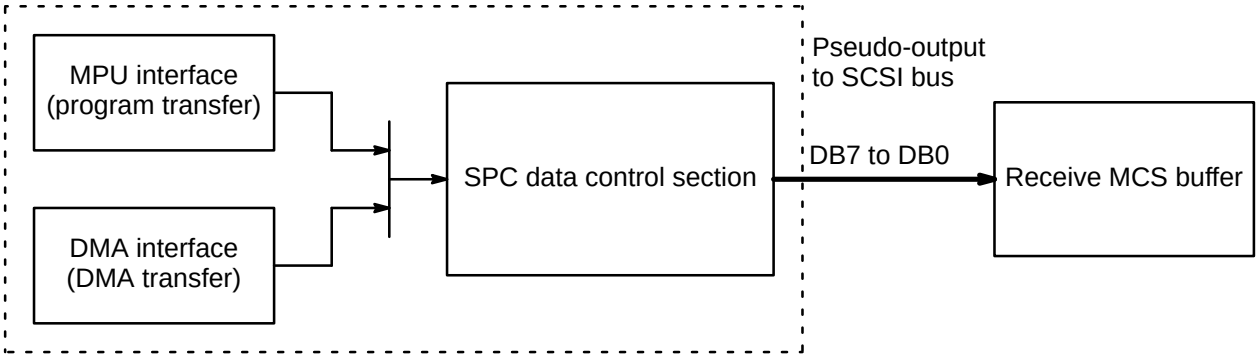
Note: In order to avoid the Parity Error, make sure the SCSI parity setting as follows:

SCSI Parity Generate : Enable
SCSI Parity Check : Disable

(3) Sending in data phase

Store the pseudo-output data in the receive MCS buffer. If the transfer data is 33 bytes or more, storing data in the receive MCS buffer is repeated.

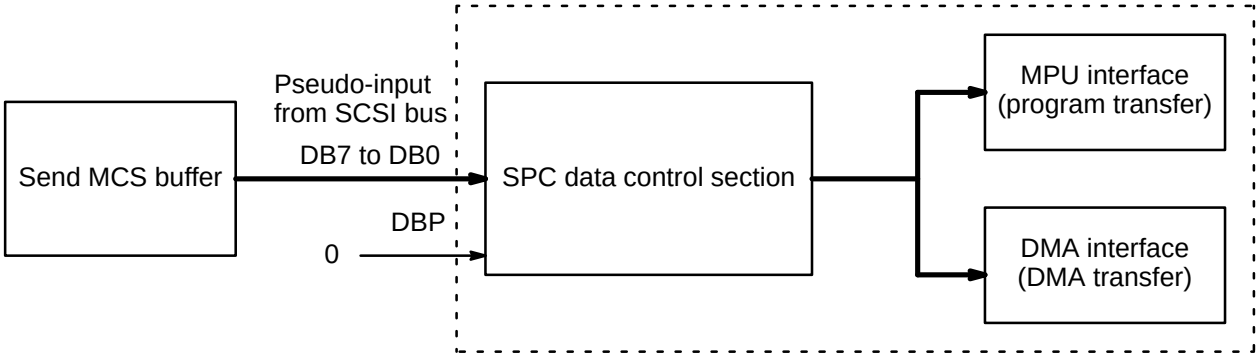
The data phase can be executed in the synchronous mode. However, the transfer parameters correspond to the values set at the self-ID. Preset the self-ID at the SEL/RESEL ID register before specifying the transfer parameters.



(4) Receiving in data phase

Write the pseudo-input data from the SCSI bus to the send MCS buffer and then issue the command. If the transfer data is 33 bytes or more, data in the send MCS buffer is regarded as repetitive pseudo-input data.

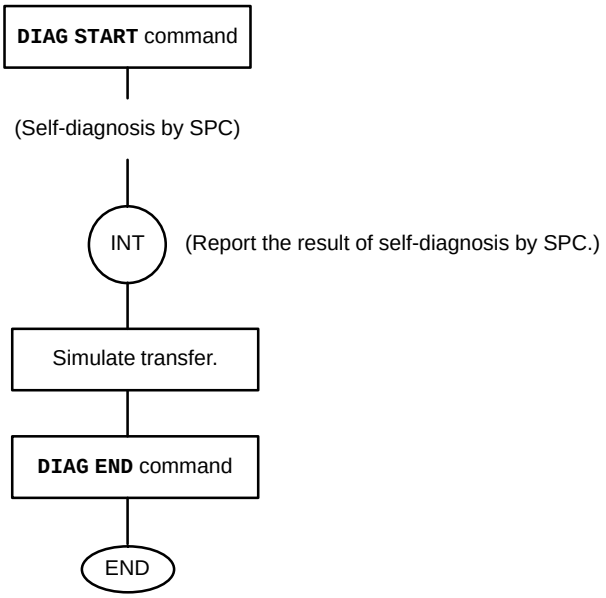
The data phase can be executed in the synchronous mode. However, the transfer parameters correspond to the values set at the self-ID. Preset the self-ID at the SEL/RESEL ID register before specifying the transfer parameters.



Note: In order to avoid the Parity Error, make sure the SCSI parity setting as follows:

SCSI Parity Generate : Enable
SCSI Parity Check : Disable

5.7.3. EXECUTION OF DIAGNOSTIC MODE



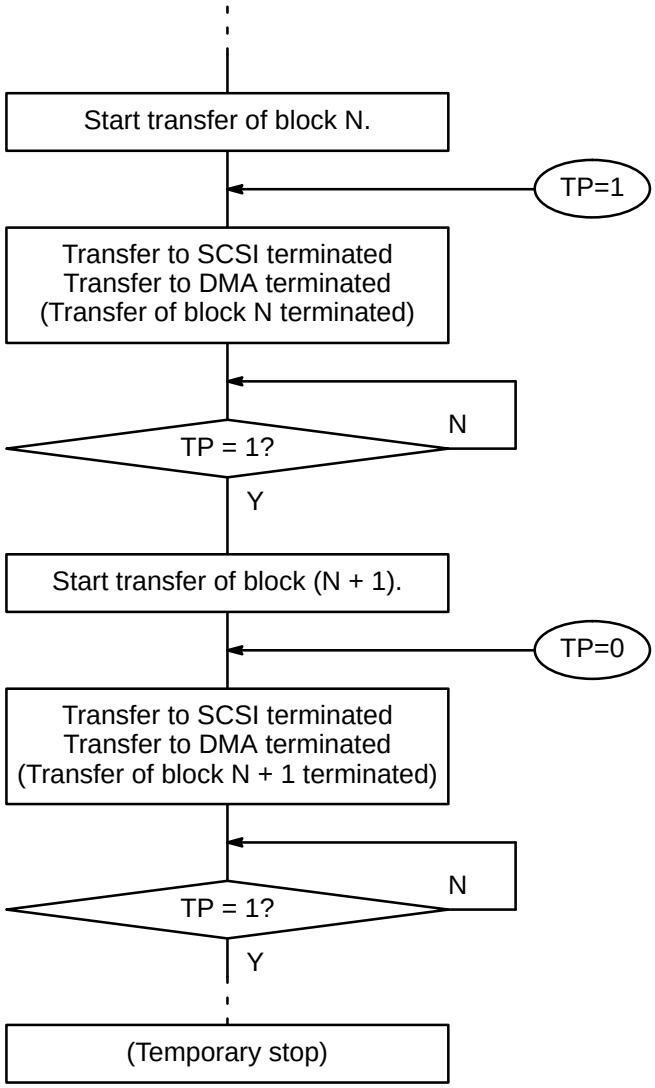
Note: Prior to execution of the **DIAG START** command, set the Bits 6 and 7 of Response Operation Mode Setting Register “disable”. Do not execute the **DIAG START** and **DIAG END** command with the bits 6 and 7 enabled or with nexus with the other device established.

5.8. TP SIGNAL CONTROL

The TP (Transfer Permission) signal is used to stop DAM transfer temporarily. The SPC controls transfer every block, checks the TP signal for each block, and determines whether to transfer the next block.

To stop DMA transfer temporarily, input 0 during transfer of the block to be stopped.

Example:
Temporarily stop DMA transfer at completion of block (N + 1) transfer.



* Temporarily stopped state

- Initiator — Keep the last ACK signal asserted just before the temporary stop. The DREQ signal goes Low. In the data-in phase, receive the data as numbers of REQ signal received.
- Target — Do not assert the REQ signal. The DREQ signal goes Low (the data register is empty).

5.9. DMA MISALIGNMENT

DMA misalignment aligns odd byte data to word boundary and connects data between transfer commands, and data phases.

The MB86605 can also perform the misalignment automatically when two or more transfer commands are issued in one data phase (automatic misalignment mode), in addition to the manual misalignment mode on.

The MB86605 actually performs the DMA misalignment by swapping the connection between external DMA bus and internal data register FIFO in byte unit as follows:

The Bit 4 of Nexus Status Register indicates the current status of DMA misalignment.

At 80 Series MPU	DMA Bus	Data Register
Misalignment OFF (Bit 4 of Nexus status reg = ‘0’.)	MSB (DMD15–8) LSB (DMD7 –0)	MSB (Second data) LSB (First data)
Misalignment ON (Bit 4 of Nexus status reg=‘1’.)	MSB (DMD15–8) LSB (DMD7–0)	MSB (Second data) LSB (First data)

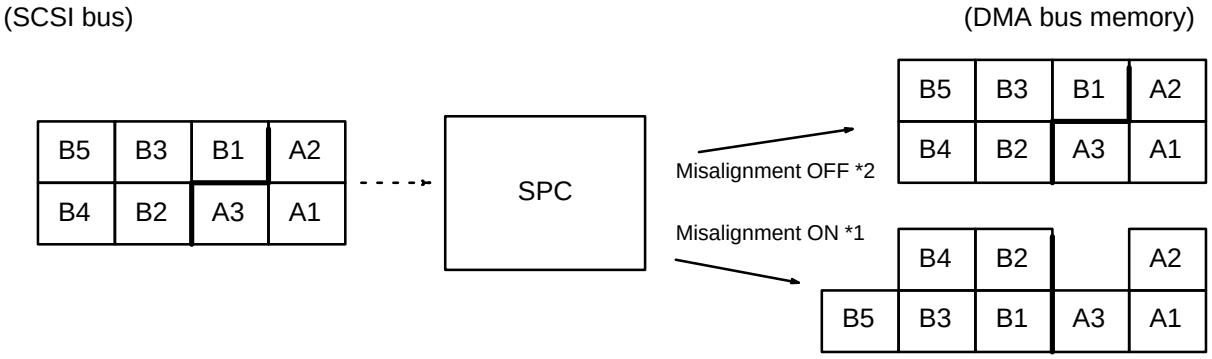
At 68 Series MPU	DMA Bus	Data Register
Misalignment OFF (Bit 4 of Nexus status reg = ‘0’)	MSB (DMD15–8) LSB (DMD7 –0)	MSB (Second data) LSB (First data)
Misalignment ON (Bit 4 of Nexus status reg=‘1’.)	MSB (DMD15–8) LSB (DMD7–0)	MSB (Second data) LSB (First data)

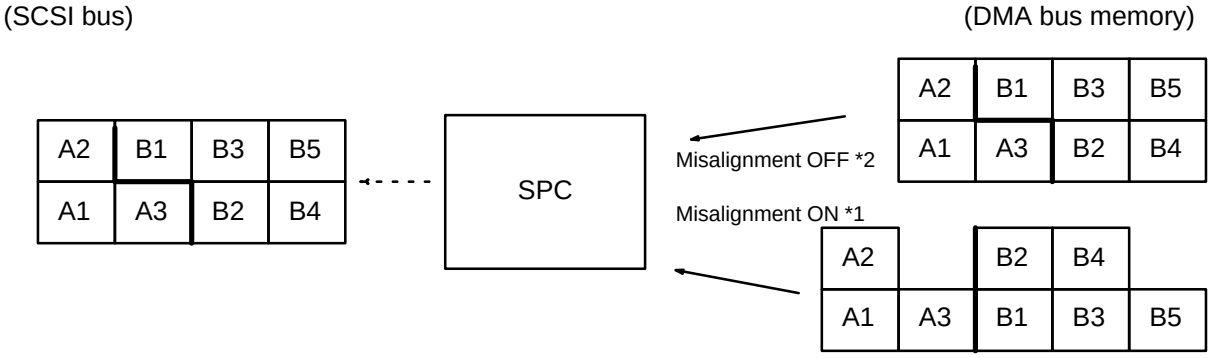
5.9.1. MANUAL MISALIGNMENT MODE

Alignment ON/OFF can be set by the bit 4 of I/O Control Register. The following are descriptions how to use the misalignment.

(1) How to Use in One Data Phase :

In the MB86605, two or more commands can be issued in one data phase. Since SPC data register pointer continues to operate, the SPC consecutively writes/reads to/from the data register as if operating in response to only one command. Therefore, even if the byte data to be transferred with a previous transfer command is an odd number, the next data transfer can be performed from word boundary if the next transfer command is issued after the misalignment is set to ON. (When the misalignment is set to OFF, swapping the byte data is needed by the system side for the next transfer command.)

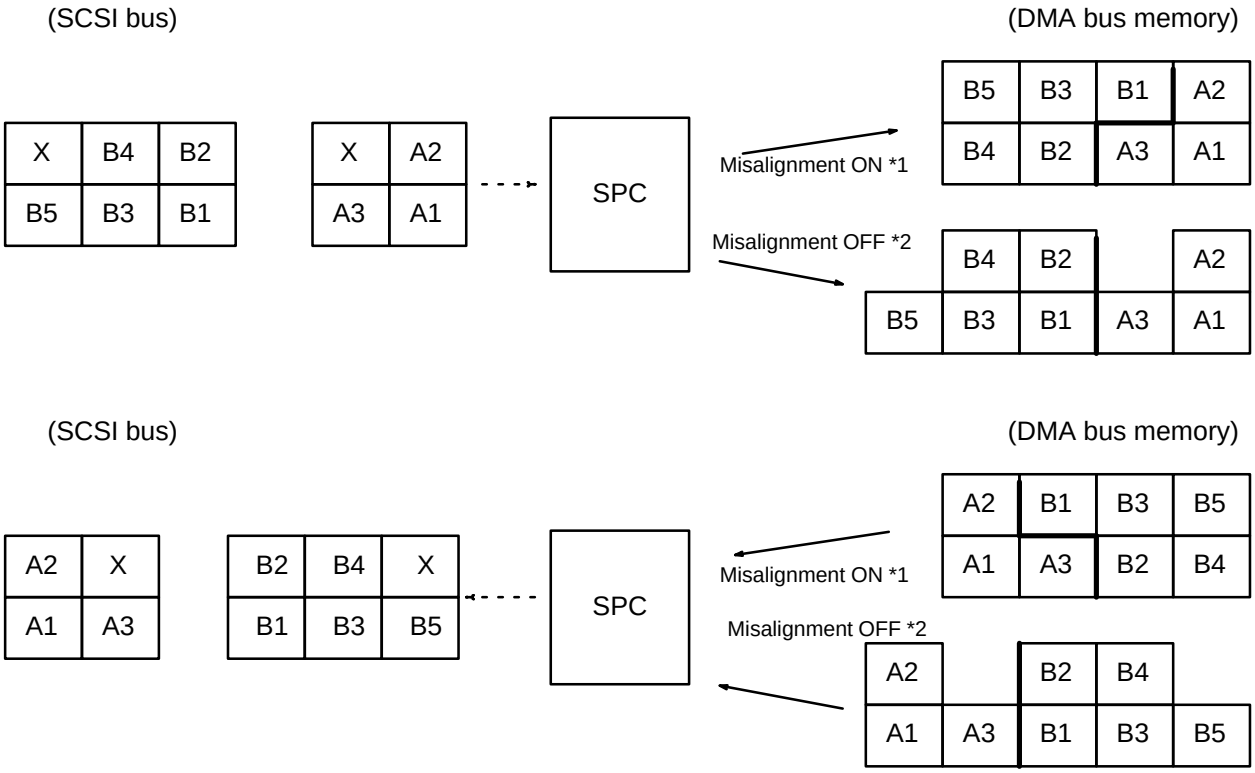




* 1 : A1 to A3 are transferred by the misalignment OFF and B1 to B5 are by the misalignment ON.
* 2 : Both A1 to A3 and B1 to B5 are transferred by the misalignment OFF.

(1) How to Use between Different Data Phases :

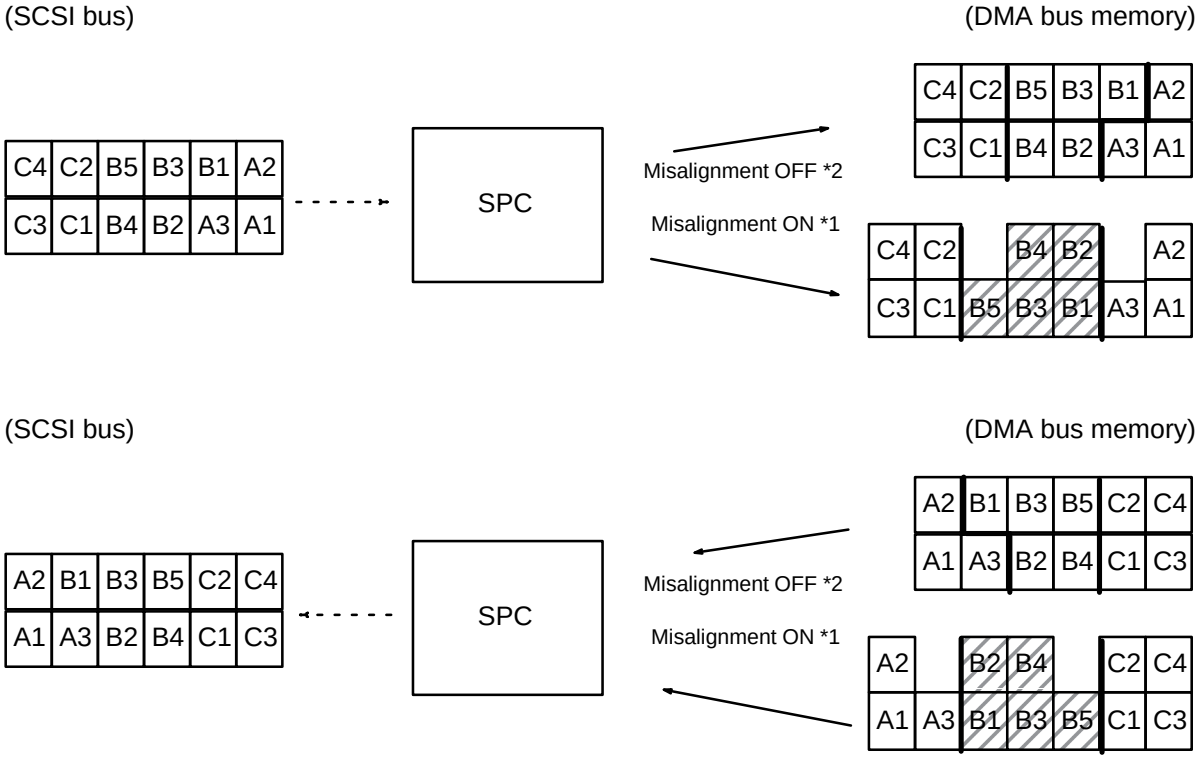
When data phase is complete, the SPC data register pointer returns to initial state. So, when the misalignment is set to ON, data in the next data phase can be transferred to the system side connecting with data in the previous data phase.



* 1 : A1 to A3 are transferred by the misalignment OFF and B1 to B5 are by the misalignment ON.
* 2 : Both A1 to A3 and B1 to B5 are transferred by the misalignment OFF.

5.9.2. AUTOMATIC MISALIGNMENT MODE

By setting the Bit 3 of I/O Control Register, the automatic misalignment can be used. When two or more commands are issued in one data phase in the automatic misalignment mode, the misalignment mode is automatically switched to ON/OFF depending on whether total transfer byte in the data phase is odd or even number.



* 1 : Transferred by setting Bit 3 of I/O Control register to '1'.
* 2 : Transferred by setting Bit 3 of I/O Control register to '0'.
* 3 : B1 to B5 are automatically transferred by the misalignment ON.

5.10. MULTISELECTION/RESELECTION RESPONSE

Multiselection/reselection response performs response operation to selection/reselection requests from other SCSI devices with two or more IDs. This feature realizes multitarget function for disk array applications.

5.10.1. INTERNAL REGISTERS FOR MULTISELECTION/RESELECTION RESPONSE

(1) Response Own ID Setting Register (MSB)

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
11	1Ch		2Ch	R/W	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

(2) Response Own ID Setting Register (LSB)

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
11	1Dh		2Dh	R/W	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

These registers are used to set the own ID to respond to the selection/reselection request from the partner device. The SPC responds for the ID where '1' is set to the corresponding bit. When the all bit value is 0, the SPC does not responds to the selection/reselection.

(3) Response Own ID Display Register

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
11	1Bh		2Bh	R/W	'0'	'0'	'0'	'0'	Response Own ID			
									RO3	RO2	RO1	RO0
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

This register indicates the own ID in hexadecimal which responds to the selection/reselection request from the partner device. By reading out this register, you can recognize the ID which responds to the selection/reselection request among IDs specified in the Response Own ID Setting Register.

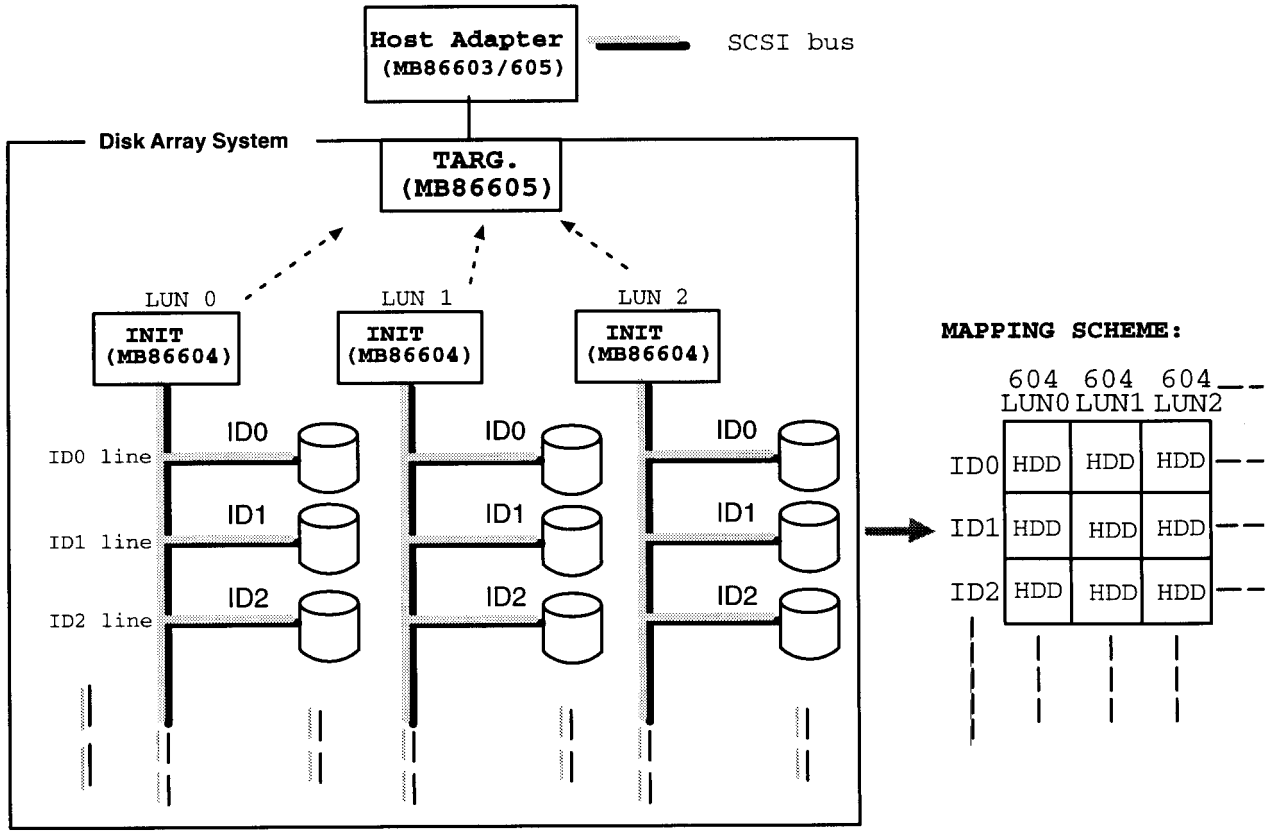
5.10.2. OTHER RELATED REGISTERS

(1) Own ID Setting Register

Bank	Address			R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	MT	In	PCI									
11	11h		21h	R/W	'0'	'0'	'0'	'0'	Own ID OI3	OI2	OI1	OI0
Initial Value					'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'

Set the bus device own ID number of the SPC that outputs when the arbitration and selection/reselection request are output.

Example of Multi-Selection Response:



1. IN RAID 5 a block of data is written into one HDD:

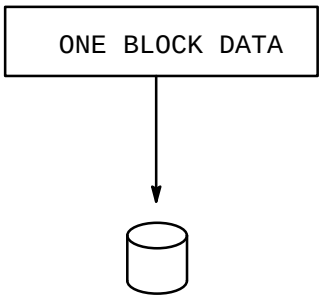


Fig. 1

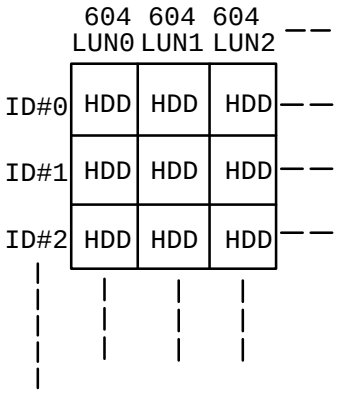


Fig. 2 HDD Array Mapping Scheme

2. When the Target in the disk array system (MB86605) responds to any of ID #1 HDD and then LUN2 INIT (MB86604) was specified in command phase, LUN2 INIT (MB86604) will nexus with the ID #1 HDD connected with the LUN2 INIT and write data received from host chip into the ID#1 HDD. (See Fig. 2.)

3. As well as the process 2 above, some of disk array systems have a possibility to respond with other IDs (HDDs), not only with a single HDD. In that case, in fact, the target (MB86005) will need to respond to the selection for multiple INIT devices (MB86604) up to 15.

The unique feature of this scheme is that each array-disk (HDD) is unlikely handled as “LUN”, but each INIT device (MB86604) is handled as “LUN”. Therefore, in RAID 5, two-dimensional HDD selection is possible using LUN and ID. For such an application, MB86605 supports multi-selection responses that can respond with selection for multiple LUNs (INIT devices) when writing/reading data into/from any desired HDDs.

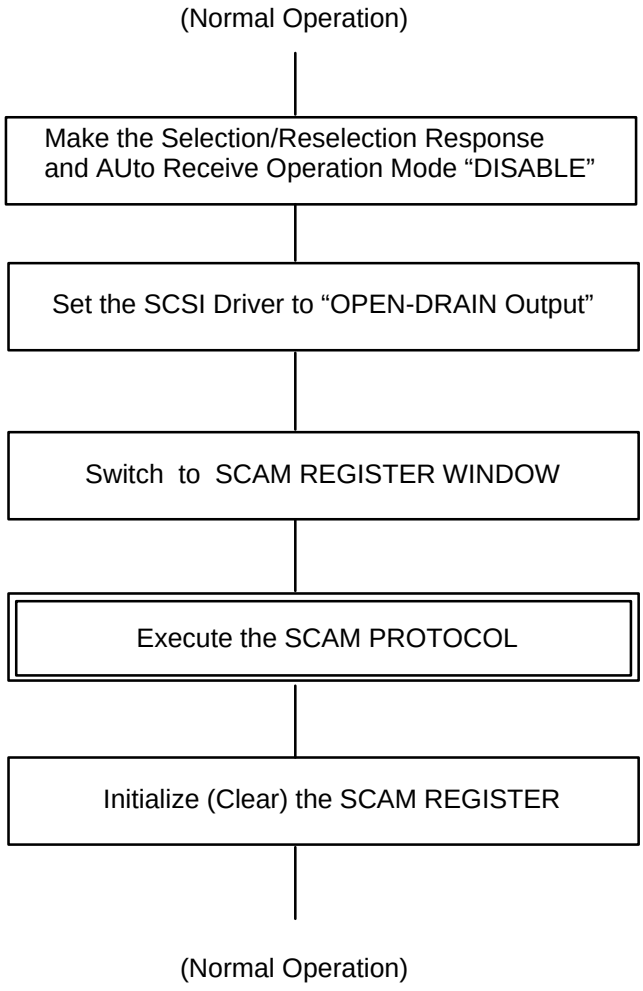
5.11. SCAM FUNCTION

The MB86605 supports the SCAM function that allows the SCAM 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.

Note: Please refer to the SCAM document for details on the operation of this protocol.

5.11.1. SCAM Protocol

To execute the SCAM protocol, take the following steps and set the MB86605.



5.11.2. 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 and make Hi–Z state) due to the open-drain mode.

APPENDIX

APPENDIX A. USER PROGRAM OPERATION

A-1. EXAMPLE OF INITIATOR USER PROGRAM

Two kinds of program example in executing a series of phases for command queuing are given below. Example 1 gives program examples that are compatible with earlier Fujitsu SPCs MB86601A/MB86602C/MB86603. Example 2 gives an example when using various automatic operation modes which are supported in the MB86604A and MB86605 (by setting the Auto Operation Mode Setting Register (addr 1Eh).)

(1) Program prerequisites

- When “Initial Phase Error” occurs, check for the phase and stops after proceeding to the receive operation.
- Store the execution results of each phase (interrupt status, command step) sequentially from address 90H. (For Examples 1.)
- Set the ACK Reset Mode to ‘automatic’ and the Level-2 Interrupt Report Mode to ‘not report’ in the Auto Operation Mode Setting Register (addr 1Eh). (For Example 2.)

(2) Outline program flow

Figure A-1 shows an outline of flow chart for Initiator user program. Figure A-2 shows the detailed flow chart.

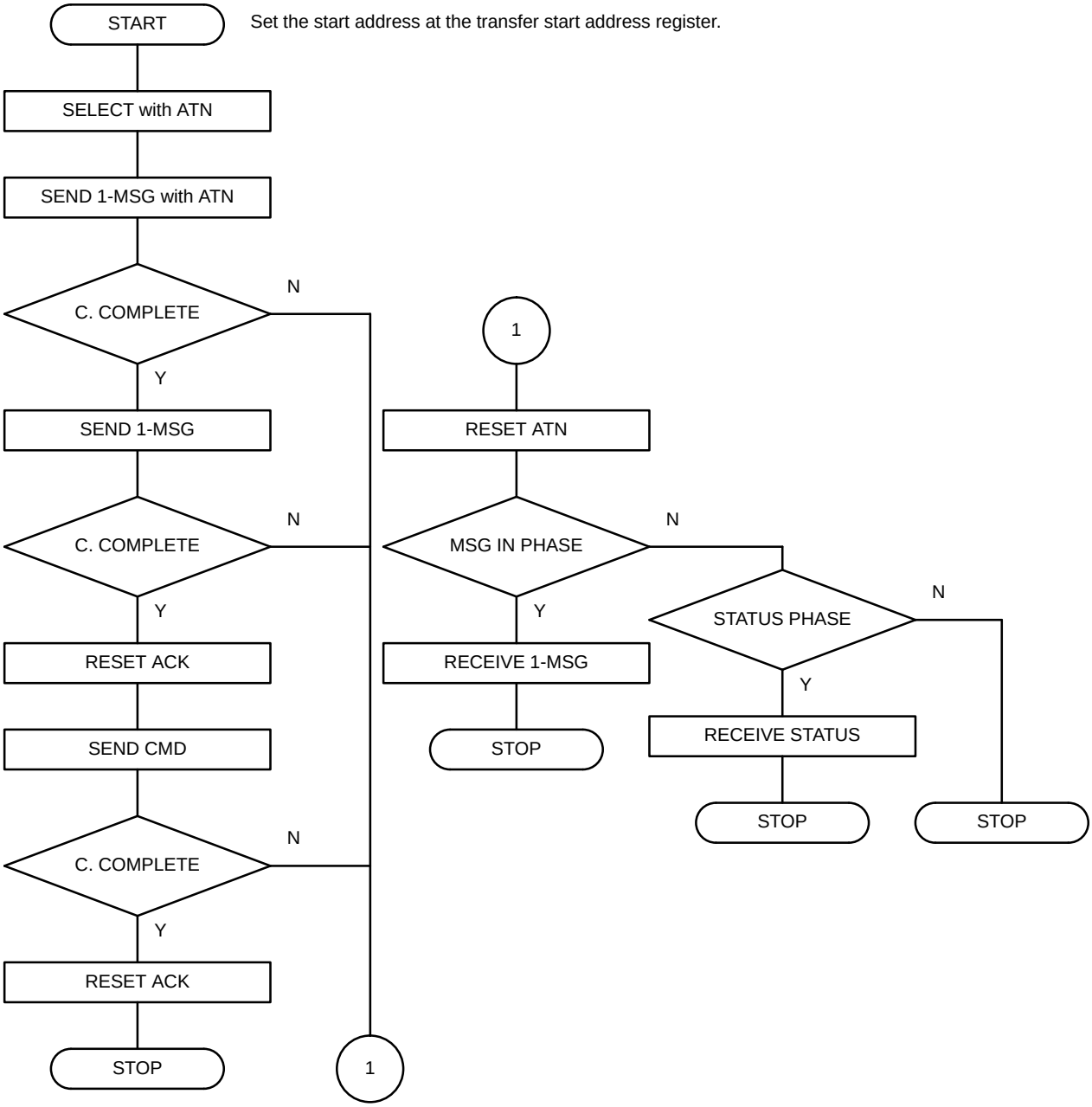


Figure A-1. Outline of Initiator User Program Example (Example 1)

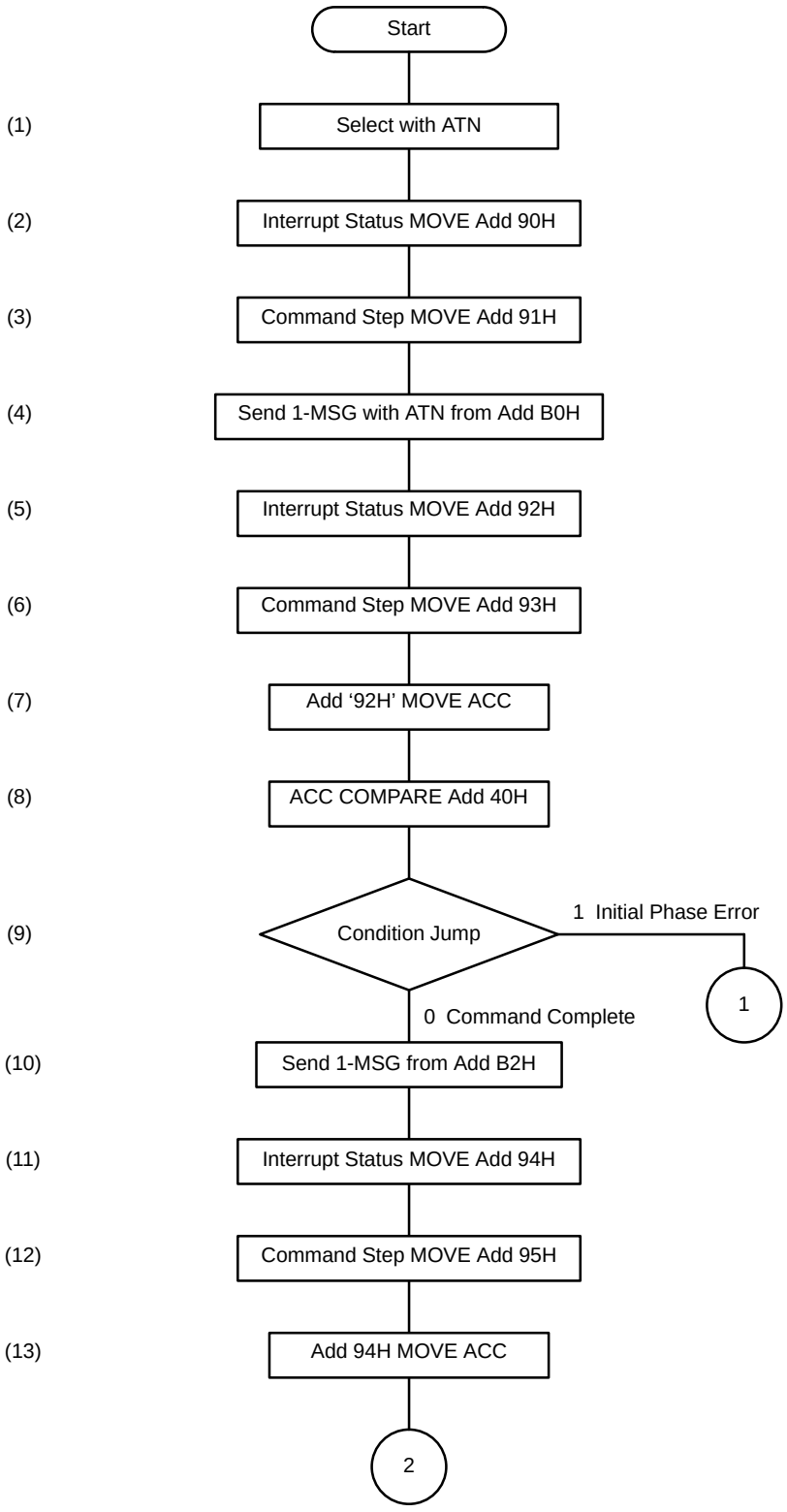
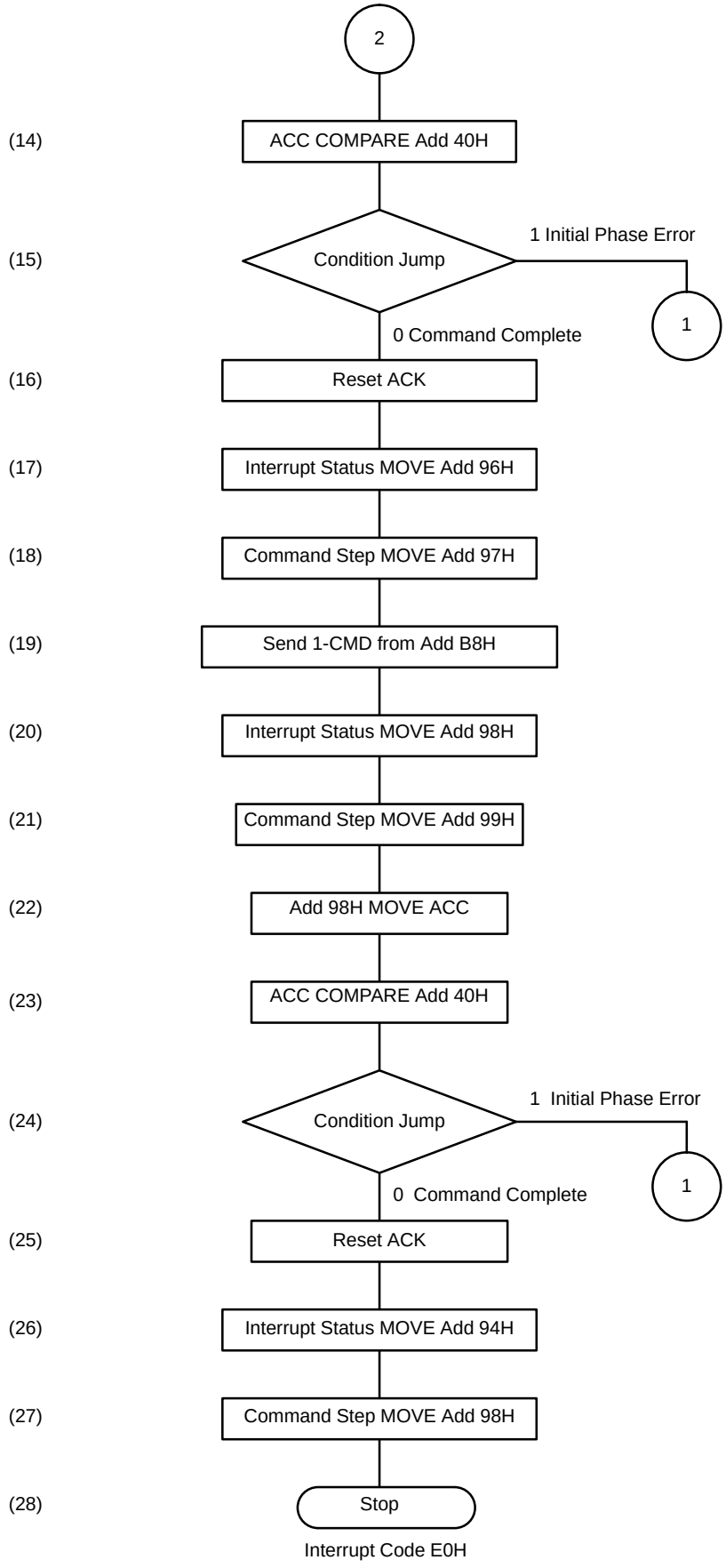
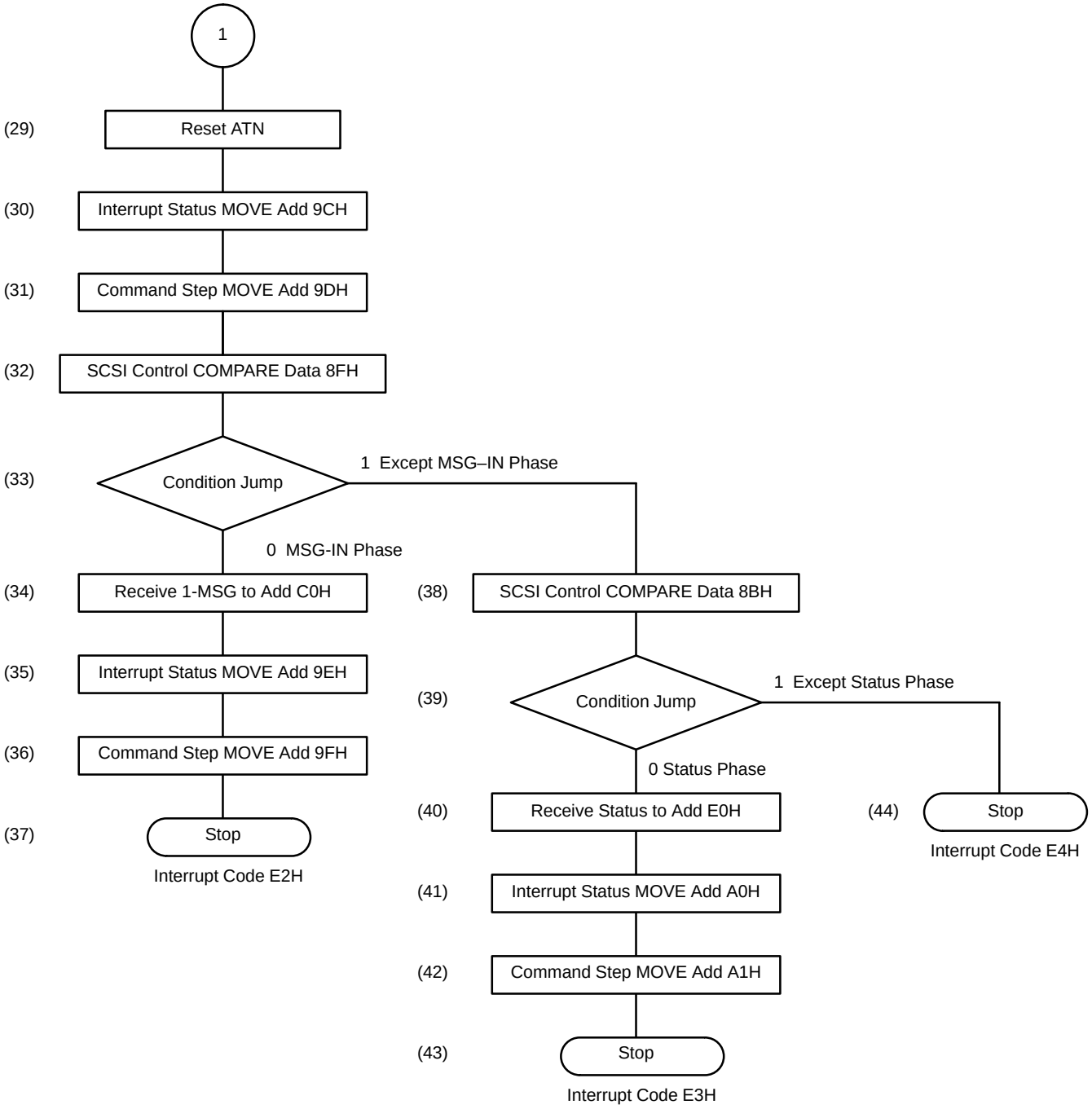


Figure A-2. Detailed Initiator User Program Flow (Example 1)


2
(14)
ACC COMPARE Add 40H
(15)
Condition Jump
1 Initial Phase Error
1
0 Command Complete
(16)
Reset ACK
(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
(24)
Condition Jump
1 Initial Phase Error
1
0 Command Complete
(25)
Reset ACK
(26)
Interrupt Status MOVE Add 94H
(27)
Command Step MOVE Add 98H
(28)
Stop
Interrupt Code E0H


1
(29)
Reset ATN
(30)
Interrupt Status MOVE Add 9CH
(31)
Command Step MOVE Add 9DH
(32)
SCSI Control COMPARE Data 8FH
(33)
Condition Jump
1 Except MSG–IN Phase
0 MSG-IN Phase
(34)
Receive 1-MSG to Add C0H
(35)
Interrupt Status MOVE Add 9EH
(36)
Command Step MOVE Add 9FH
(37)
Stop
Interrupt Code E2H
(38)
SCSI Control COMPARE Data 8BH
(39)
Condition Jump
1 Except Status Phase
0 Status Phase
(40)
Receive Status to Add E0H
(41)
Interrupt Status MOVE Add A0H
(42)
Command Step MOVE Add A1H
(43)
Stop
Interrupt Code E3H
(44)
Stop
Interrupt Code E4H

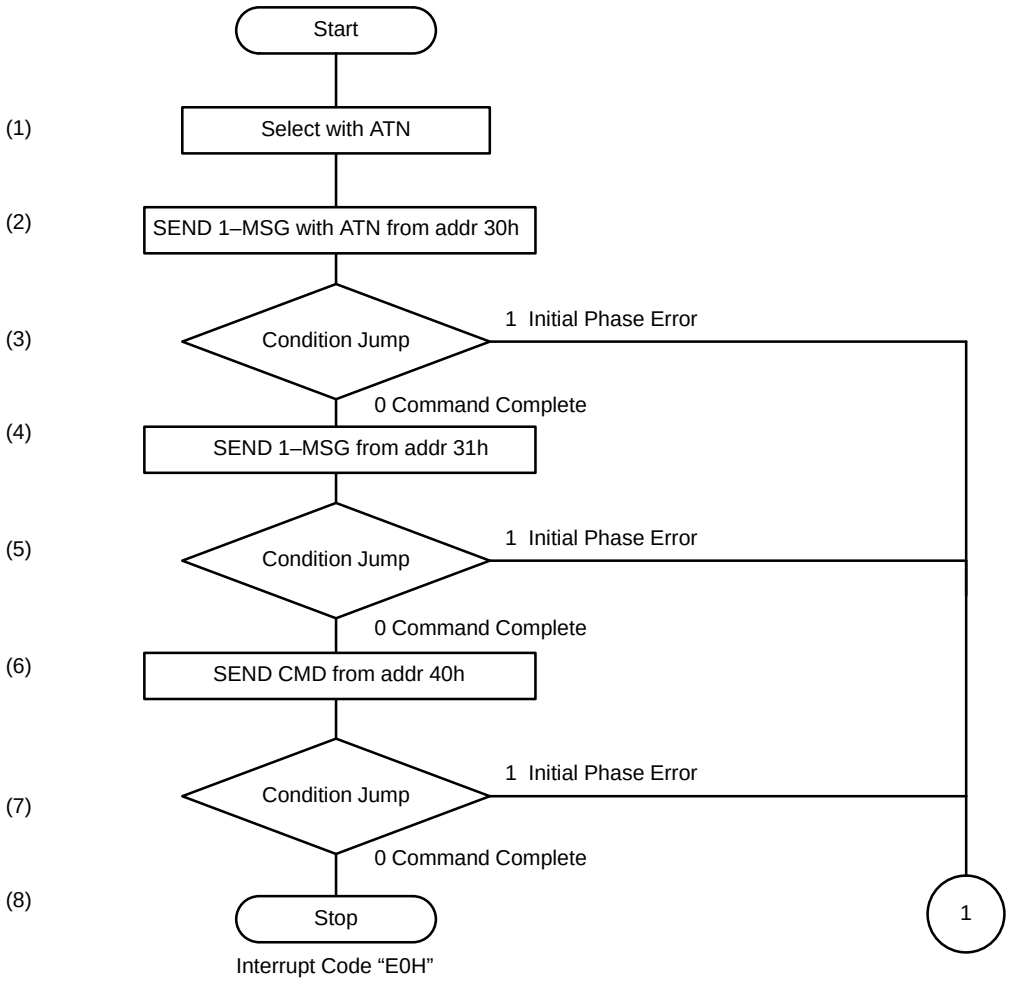
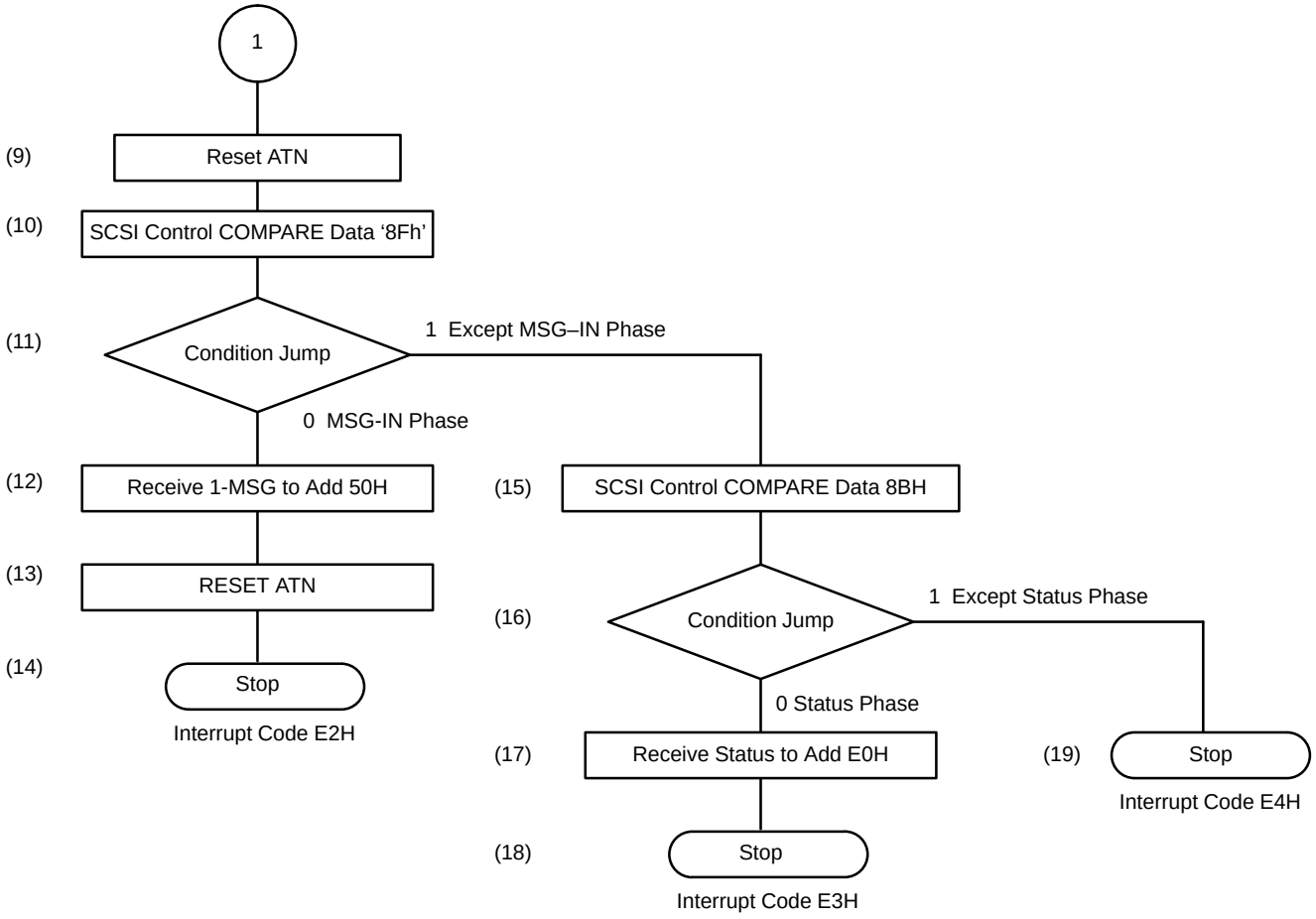


Figure A-3 Initiator User Program Example (Example 2)


1
(9)
Reset ATN
(10)
SCSI Control COMPARE Data '8Fh'
(11)
Condition Jump
1 Except MSG–IN Phase
0 MSG-IN Phase
(12)
Receive 1-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 E0H
(18)
Stop
Interrupt Code E3H
(19)
Stop
Interrupt Code E4H

Explanation of Initiator User Program Example 1(Addresses 00h to 34h)

No.	Address	Code	Explanation
1	00	09	Issue **SELECT with ATN** command.
2	01 02	EC 90	Move value of interrupt status register to address 90H.
3	03 04	ED 91	Move value of command step register to address 91H.
4	05 06	19 B0	Issue **SEND 1-MSG with ATN** command. Store 1-byte send data from address B0H.
5	07 08	EC 92	Move value of interrupt status register to address 92H.
6	09 0A	ED 93	Move value of command step register to address 93H.
7	0B 0C	E0 92	Move data stored at address 92H to accumulator.
8	0D 0E	CF 40	Compare value of accumulator with data stored at address 40H (**COMPARE**).
9	0F 10	D1 50	If operation result ≠ 0, branch to address 50H.
10	11 12	18 B2	Issue **SEND 1-MSG** command. Store 2-byte send data from address B2H.
11	13 14	EC 94	Move value of interrupt status register to address 94H.
12	15 16	ED 95	Move value of command step register to address 95H.
13	17 18	E0 94	Move data stored at address 94H to accumulator.
14	19 1A	CF 40	Compare value of accumulator with data stored at address 40H (**COMPARE**).
15	1B 1C	D1 50	If operation result ≠ 0, branch to address 50H.

Explanation of Initiator User Program Example 1 (Addresses 00h to 34h) (Continued)

No.	Address	Code	Explanation
16	1D	0D	Issue **RESET ACK** command.
17	1E 1F	EC 96	Move value of interrupt status register to address 96H.
18	20 21	ED 97	Move value of command step register to address 97H.
19	22 23	1B B8	Issue **SEND CMD** command. Store 6-byte send data from address B8H.
20	24 25	EC 98	Move value of interrupt status register to address 98H.
21	26 27	ED 99	Move value of command step register to address 99H.
22	28 29	E0 98	Move data stored at address 98H to accumulator.
23	2A 2B	CF 40	Compare value of accumulator with data stored at address 40H (**COMPARE**).
24	2C 2D	D1 50	If operation result ≠ 0, branch to address 50H.
25	2E	0D	Issue **RESET ACK** command.
26	2F 30	EC 9A	Move value of interrupt status register to address 9AH.
27	31 32	ED 9B	Move value of command step register to address 9BH.
28	33 34	91 E0	Issue **STOP** command. The interrupt code E0H (normal termination) is reported.

Explanation of Initiator User Program Example 1 (Addresses 40h to E0h)

No.	Address	Code	Explanation
	40	60	Store value at address 60H that will be compared with value of interrupt status register.
29	50	0B	Issue **RESET ATN** command.
30	51 52	EC 9C	Move value of interrupt status register to address 9CH.
31	53 54	ED 9D	Move value of command step register to address 9DH.
32	55 56	C6 8F	Compare value of SCSI control register data stored at address 8FH (**COMPARE**).
33	57 58	D1 70	If operation result ≠ 0, branch to address 70H.
34	59 5A	1A C0	Issue **RECEIVE 1-MSG** command. Store received data from address C0H.
35	5B 5C	EC 9E	Move value of interrupt status register to address 9EH.
36	5D 5E	ED 9F	Move value of command step register to address 9FH.
37	5F 60	91 E2	Issue **STOP** command. The interrupt code E2H (one message was received after the **Initial Phase Error** occurred) is reported.
38	70 71	C6 8B	Compare value of SCSI control register with data stored at address 8BH (**COMPARE**).
39	72 73	D1 7C	If operation result ≠ 0, branch to address 7CH.
40	74 75	1C E0	Issue **RECEIVE STATUS** command. Store received data from address E0H.
41	76 77	EC A0	Move value of interrupt status register to address A0H.
42	78 79	ED A1	Move value of command step register to address A1H.
43	7A 7B	91 E3	Issue **STOP** command. The interrupt code E3H (status was received after the **Initial Phase Error** occurred) is reported.
44	7C 7D	91 E4	Issue **STOP** command. The interrupt code E4H (the next phase was not RECEIVE MCS after the **Initial Phase Error** occurred) is reported.

Explanation of Initiator User Program Example 1 (Addresses 40h to E0h) (Continued)

No.	Address	Code	Explanation
	90 91		Store result of executing **SELECT with ATN** command (interrupt status and command step; to be repeated in the following).
	92 93		Store result of executing **SEND I-MSG with ATN** command.
	94 95		Store result of executing **SEND 1-MSG** command.
	96 97		Store result of executing **RESET ACK** command.
	98 99		Store result of executing **SEND CMD** command.
	9A 9B		Store result of executing **RESET ACK** command.
	9C 9D		Store result of executing **RESET ATN** command.
	9E 9F		Store result of executing **RECEIVE 1-MSG** command.
	A0 A1		Store result of executing **RECEIVE STATUS** command.
	B0	XX	Store 1-byte data to be transferred by **SEND 1-MSG with ATN** command.
	B2 B3	XX XX	Store 2-byte data to be transferred by **SEND 1-MSG** command.
	B8 B9 BA BB BC BD	XX XX XX XX XX XX	Store 6-byte data to be transferred by **SEND CMD** command.
	C0 to DF		When receiving one message after **Initial Phase Error** occurs, store data.
	E0		When receiving one status after **Initial Phase Error** occurs, store data.

Explanation of Initiator User Program Example 2 (Addresses 00h to 70h)

No.	Address	Code	Explanation
1	00	09	Issue **SELECT with ATN** command.
2	01 02	19 30	Issue **SEND 1-MSG with ATN** command. The data to be sent is 1- byte from address 30h.
3	03 04	D1 10	Branch to address 10h if “Initial Phase Error” occurred.
4	05 06	18 31	Issue **SEND 1-MSG** command. The data to be sent is 2-byte from address 32h.
5	07 08	D1 10	Branch to address 10h if “Initial Phase Error” occurred.
6	09 0A	1B 40	Issue **SEND CMD** command. The data to be sent is 6-byte from address 40h.
7	0B 0C	D1 10	Branch to address 10h if “Initial Phase Error” occurred.
8	0D 0E	91 E0	Issue **STOP** command. The interrupt code to be reported is “E0h” (normal termination)
–	0F	–	–
9	10	0B	Issue **RESET ATN** command.
10	11 12	C6 8F	Compare the SCSI control register with 8Fh.
11	13 14	D1 20	If operation result ≠ 0, branch to address 20h.
12	15 16	1A 50	Issue **RECEIVE MSG** command. Store the received data from address 50h.
13	17	0D	Issue **RESET ACK** command.
14	18 19	91 E2	Issue **STOP** command. The interrupt code is E2h (1 message was received after “Initial Phase Error” occurred.)

Explanation of Initiator User Program Example 2 (Addresses 00h to 70h) (Continued)

No.	Address	Code	Explanation
15	20 21	C6 8B	Compare the SCSI control register with 8Bh. (**COMPARE**)
16	22 23	D1 28	If operation result ≠ 0, branch to address 28H.
17	24 25	1C 70	Issue **RECEIVE STATUS** command. Store the received data from address 70h.
18	26 27	91 E3	Issue **STOP** command. The interrupt code is E3h (Status was received after "Initial Phase Error" occurred.)
19	28 29	91 E4	Issue **STOP** command. The interrupt code is E4h (The next phase was not RECEIVE MCS after "Initial Phase Error" occurred.)
–	30	XX	Store the data to be sent by **SEND 1–MSG with ATN** command. (1-byte)
–	31 32	XX XX	Store the data to be sent by **SEND 1–MSG** command. (2-byte)
–	40 to 4F	XX to XX	Store the data to be sent by **SEND CMD** command.
–	50 to 6F	–	Store the data if 1 message is received after "Initial Phase Error" occurred.
–	70	–	Store the data if 1 status is received after "Initial Phase Error" occurred.

A-2. EXAMPLE OF TARGET USER PROGRAM

Two kinds of program example in executing a series of phases for command queuing are given below. Example 1 gives program examples compatible with earlier Fujitsu SPCs MB86601A/MB86602C/MB86603. Example 2 gives an example when using various automatic operation modes which are supported in the MB86604A and MB86605 (by setting the Auto Operation Mode Setting Register (addr : 1Eh).)

(1) Program prerequisites

- If the **COMMAND COMPLETE** (**ATN CONDITION DETECTED**) occurs, stop the operation without receiving.
- Store the execution results of each phase (interrupt status, command step) sequentially from address 50H. (For Example 1.)
- Set the ACK Reset Mode to 'automatic' and the Level-2 Interrupt Report Mode to 'not report' in the Auto Operation Mode Setting Register (addr : 1Eh). (For Examples 2.)

(2) Outlined program flow

Figure A-4 shows an outline of flow chart for Target user program. Figure A-5 shows the detailed flow chart.

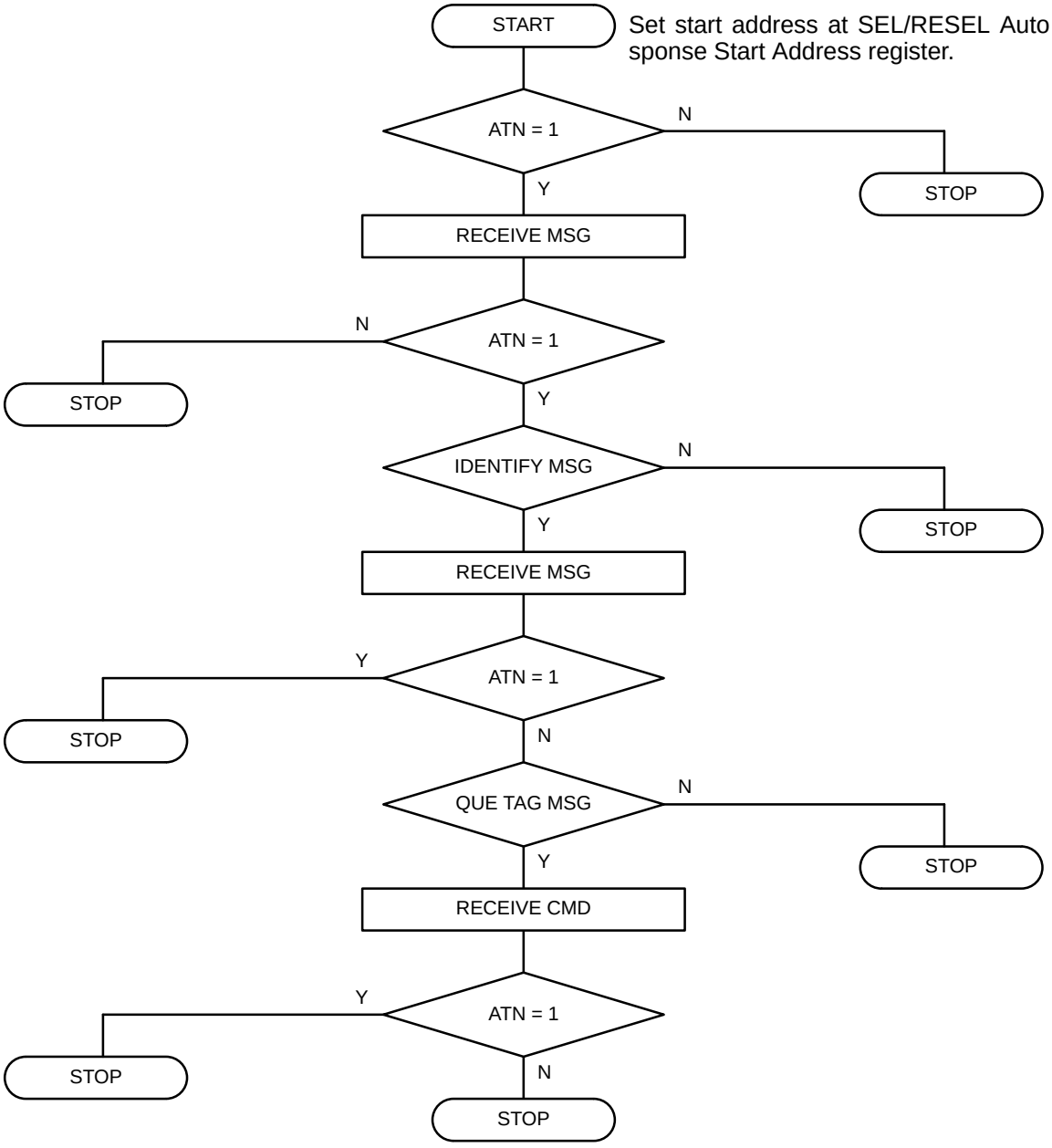


Figure A-4. Outline of Target User Program (Example 1)

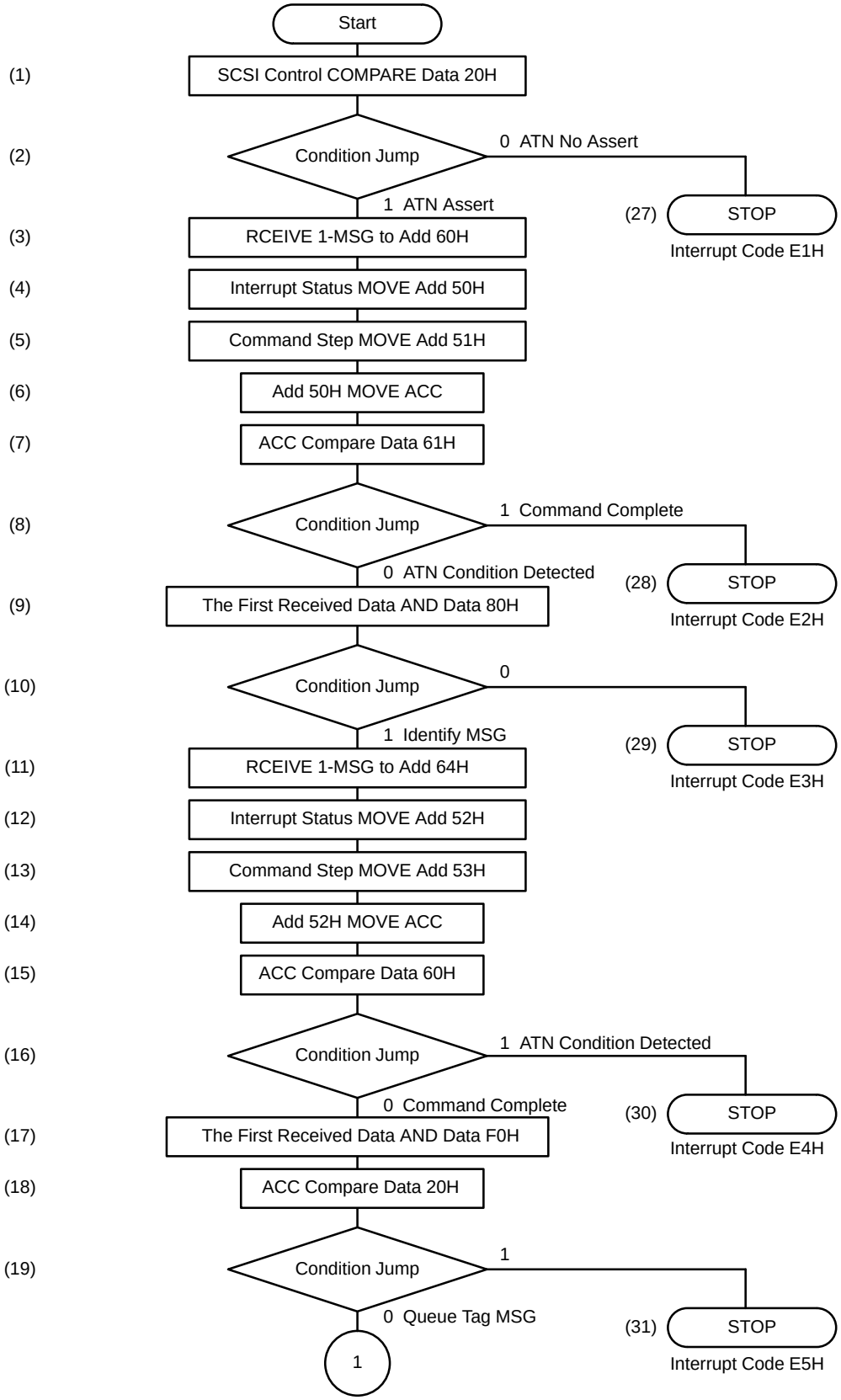
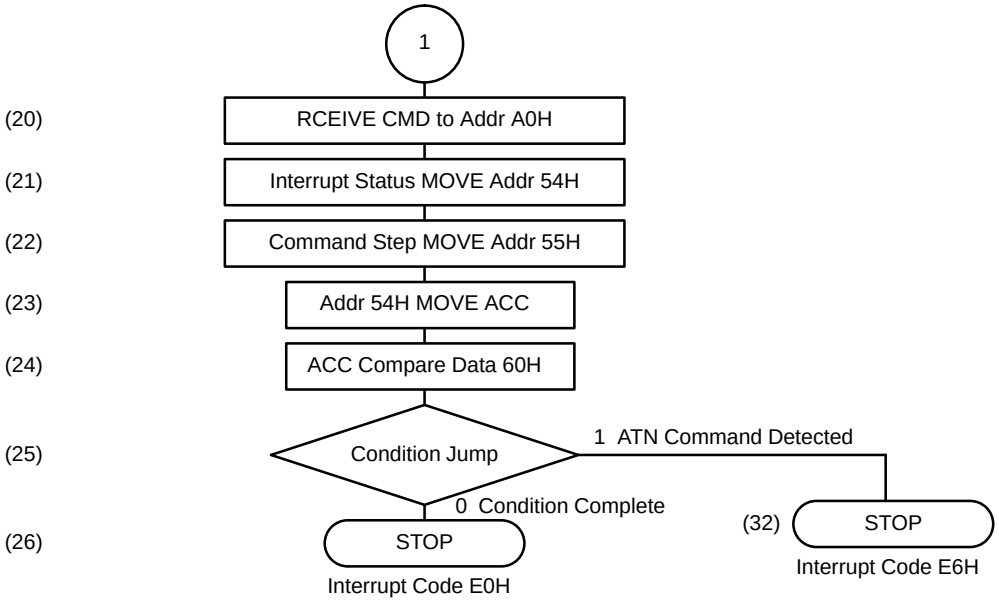


Figure A-5. Detailed Target User Program Flow (Example 1)


1
(20)
RCEIVE CMD to Addr A0H
(21)
Interrupt Status MOVE Addr 54H
(22)
Command Step MOVE Addr 55H
(23)
Addr 54H MOVE ACC
(24)
ACC Compare Data 60H
(25)
Condition Jump
1 ATN Command Detected
0 Condition Complete
(26)
STOP
Interrupt Code E0H
(32)
STOP
Interrupt Code E6H

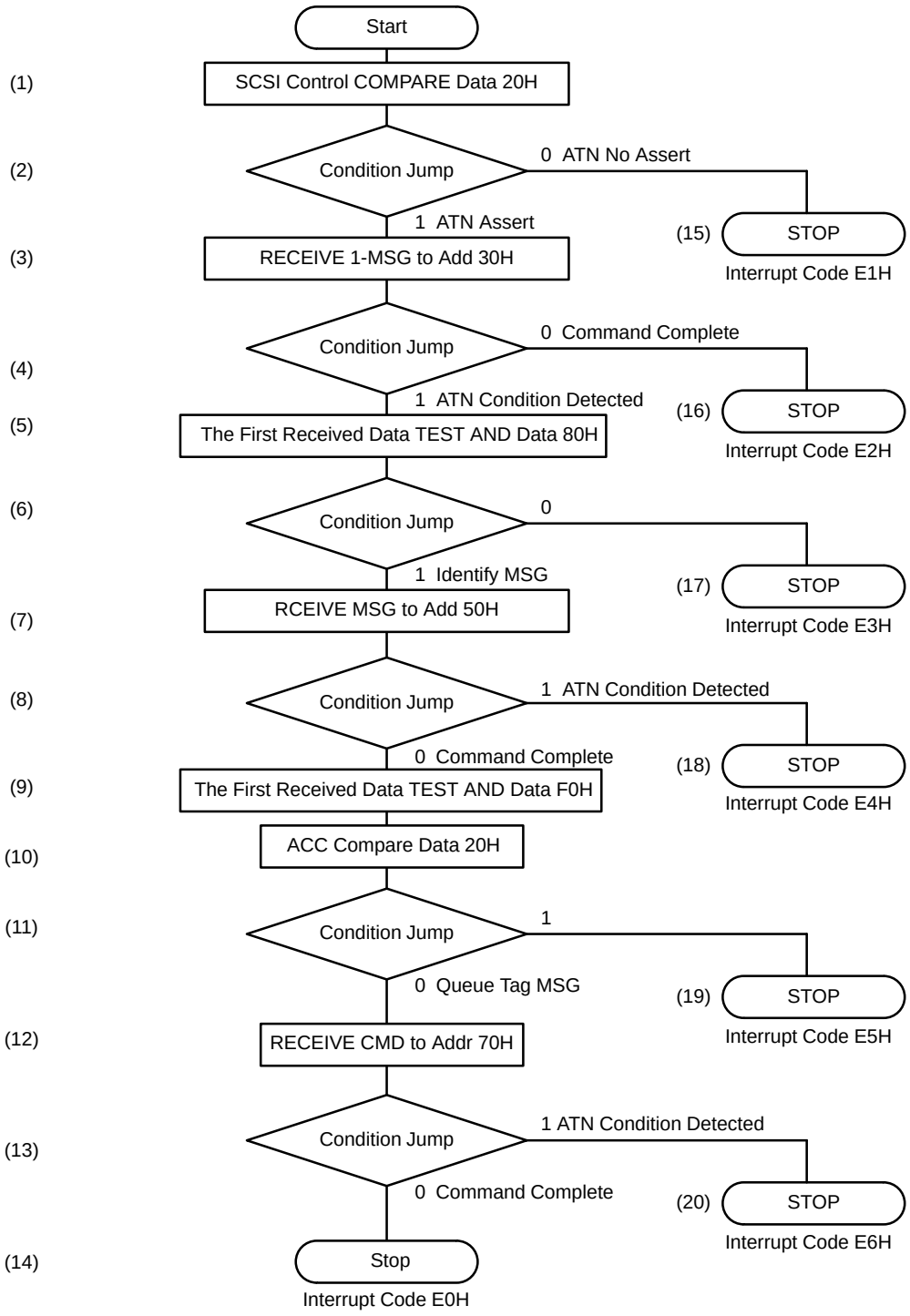


Figure A-6. Detailed Target User Program Flow (Example 2)

Explanation of Target User Program Example 1 (Addresses 00h to 21h)

No.	Address	Code	Explanation
1	00 01	B6 20	Carry out logical AND between the value of SCSI control signal status register and data stored at address 20H (**TEST AND**).
2	02 03	D2 40	If operation result = 0, branch to address 40H.
3	04 05	39 60	Issue **RECEIVE MSG** command. Store received data at address 60H.
4	06 07	EC 50	Move value of interrupt status register to address 50H.
5	08 09	ED 51	Move value of command step register to address 51H.
6	0A 0B	E0 50	Move data stored at address 50H to accumulator.
7	0C 0D	C7 61	Compare value of accumulator with data stored at address 61H (**COMPARE**).
8	0E 0F	D1 42	If operation result ≠ 0, branch to address 42H.
9	10 11	B0 80	Carry out logical AND between first byte of received message with data stored at address 80H (**TEST AND**).
10	12 13	D2 44	If operation result = 0, branch to address 44H.
11	14 15	39 80	Issue **RECEIVE MSG** command. Store received data from address 80H.
12	16 17	EC 52	Move value of interrupt status register to address 52H.
13	18 19	ED 53	Move value of command step register to address 53H.
14	1A 1B	E0 52	Move data stored at address 52H to accumulator.
15	1C 1D	C7 60	Compare value of accumulator with data stored at address 60H (**COMPARE**).
16	1E 1F	D1 46	If operation result ≠ 0, branch to address 46H.
17	20 21	A0 F0	Carry out logical AND between first byte of received message and data stored at address F0H (**AND**).

Explanation of Target User Program Example 1 (Addresses 22h to BFh)

No.	Address	Code	Explanation
18	22 23	C7 20	Compare value of accumulator with data stored at address 20H (**COMPARE**).
19	24 25	D1 48	If operation result ≠ 0, branch to address 48H.
20	26 27	3B A0	Issue **RECEIVE CMD** command. Store received data from address A0H.
21	28 29	EC 54	Move value of interrupt status register to address 54H.
22	2A 2B	ED 55	Move value of command step register to address 55H.
23	2C 2D	E0 54	Move data stored at address 54H to accumulator.
24	2E 2F	C7 60	Compare value of accumulator with data stored at address 60H (**COMPARE**).
25	30 31	D1 4A	If operation result ≠ 0, branch to address 4AH.
26	32 33	91 E0	Issue **STOP** command. The interrupt code E0H (normal termination) is reported.
27	40 41	91 E1	Issue **STOP** command. The interrupt code E1H (the ATN signal was not asserted in the selection phase) is reported.
28	42 43	91 E2	Issue **STOP** command. The interrupt code E2H (the ATN signal was not asserted after the first message was received) is reported.
29	44 45	91 E3	Issue **STOP** command. The interrupt code E3H (the first received message was not **Identify-MSG**) is reported.
30	46 47	91 E4	Issue **STOP** command. The interrupt code E4H (the attention condition was detected after the second message was received) is reported.
31	48 49	91 E5	Issue **STOP** command. The interrupt code E5H (the second received message was not **Queue Tag MSG**) is reported.
32	4A 4B	91 E6	Issue **STOP** command. The interrupt code E6H (the attention condition was detected after the command was received) is reported.

Explanation of Target User Program Example 1 (Addresses 22h to BFh) (Continued)

No.	Address	Code	Explanation
	50 51		Store result of executing first **RECEIVE MSG** command (interrupt status and command step; to be repeated in the following).
	52 53		Store result of executing second **RECEIVE MSG** command.
	54 55		Store result of executing **RECEIVE CMD** command.
	60 to 7F		Store result of executing first **RECEIVE MSG** command.
	80 to 9F		Store result of executing second **RECEIVE MSG** command.
	A0 to BF		Store result of executing **RECEIVE CMD** command.

Explanation of Target User Program Example (Addresses 00h to 8Fh)

Explanation of Target User Program Example 2 (Addresses 00h to 8Fh)

No.	Address	Code	Explanation
1	00 01	B6 20	Carry out logical AND between the value of SCSI control signal status register and data stored at address 20H (**TEST AND**).
2	02 03	D2 20	If operation result = 0, branch to address 20H.
3	04 05	39 30	Issue **RECEIVE MSG** command. Store received data at address 30H.
4	06 07	D2 22	If execution result of command is 'Command Complete', branch to address 22H.
5	08 09	B0 80	Carry out logical AND between first byte of received message with data stored at address 80H (**TEST AND**).
6	0A 0B	D2 24	If operation result = 0, branch to address 24H.
7	0C 0D	39 50	Issue **RECEIVE MSG** command. Store received data at address 50H.
8	0E 0F	D1 26	If execution result of command is 'Command Complete (ATN Condition Detected)', branch to address 26H.
9	10 11	A0 F0	Carry out logical AND between first byte of received message with data stored at address F0H (**AND**).
10	12 13	C7 20	Compare value of accumulator with data stored at address 20H (**COMPARE**).
11	14 15	D1 28	If operation result ≠ 0, branch to address 28H.
12	16 17	3B 70	Issue **RECEIVE MSG** command. Store received data at address 70H.
13	18 19	D1 2A	If operation result ≠ 0, branch to address 2AH.
14	1A 1B	91 E0	Issue **STOP** command. The interrupt code is E0H (normal termination).

Explanation of Target User Program Example 2 (Addresses 00h to 8Fh) (Continued)

No.	Address	Code	Explanation
15	20 21	91 E1	Issue **STOP** command. The interrupt code is E1H. (ATN signal was not asserted in the selection phase.)
16	22 23	91 E2	Issue **STOP** command. The interrupt code is E2H. (ATN signal was not asserted after the first message received.)
17	24 25	91 E3	Issue **STOP** command. The interrupt code is E3H. (The first message received was not the Identify MSG.)
18	26 27	91 E4	Issue **STOP** command. The interrupt code is E4H. (Attention condition was detected after the second message received.)
19	28 29	92 E5	Issue **STOP** command. The interrupt code is E5H. (The second message received was not the Queue Tag MSG.)
20	2A 2B	91 E6	Issue **STOP** command. The interrupt code is E6H. (Attention condition was detected after the command received.)
–	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.

APPENDIX B. LIST OF INTERRUPT CODE & COMMAND STEP

Interrupt codes and command steps are listed in this section, including the contents of each code/step and SPC operation when the interupt is detected. "Notes" in the list describe the SPC operation or SPC setting at the detection of interrupt.

When interrupt/step codes not listed in this section are reported, please perform the following operation :

1) When nexus is not established, issue the "SOFTWARE RESET" command.
2) When nexus is established, issue the "SCSI RESET" command or issue the "SOFTWARE RESET" command after executing the "DISCONNECT" command.

B.1. Initiator Commands

a. Sequential Commands

(1) All commands

CODE	STEP	Interrupt Source and SPC Operation
31H	00H	After the transfer completed, bus free state is detected asserting the ACK signal (at the Manual Reset ACK Mode.)
64H	00H	(1) New command received during execution of command (double receiving). This interrupt is reported after reporting the execution result of the first received command. (2) When command received, interrupt source that occurs when SPC is ready was already detected and precedes the interrupt report. This interrupt is reported after reporting the interrupt source that is reported when the SPC is ready (except "REQ Asserted").
65H	00H	(1) Command from target received when operating as initiator (2) Received command undefined (3) Commands issued for starting transfer before establishing nexus and for starting selection after establishing nexus
22H	00H	Parity error detected in register set by host MPU

(2) SELECT & CMD

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected while waiting bus free state
	01H	Reset condition detected during execution of arbitration phase
	02H	Reset condition detected during execution of selection phase
	03H	Reset condition detected during execution of command phase (Note 1)
2CH	03H	More than specified REQ/ACK timeout time elapsed during execution of command phase (Note 1)
31H	03H	Bus free state detected during execution of command phase (Note 1)
32H	03H	Target changed phase during execution of command phase (Note 1)
54H	03H	Phase when REQ signal first received not command phase
55H	03H	Phase when REQ signal first received not command phase and one message received in response to a message receive request from target (Notes 2 & 5)
56H	03H	Phase when REQ signal first received not command phase and one status received in response to status receive request from target (Notes 2 & 5)
5DH	03H	Phase when REQ signal first received not command phase, and attempted to receive one message in response to message receive request from target but message not received due to data remaining in Receive MCS Buffer (Note 2)
5EH	03H	Phase when REQ signal first received not command phase, and attempted to receive one status in response to status receive request from target but status not received due to data remaining in Receive MCS Buffer (Note 2)
60H	04H	Terminated normally (Note 3)
	0CH	Terminated normally and negated the last ACK signal (Note 4)
64H	00H	(1) Arbitration failed and reselection made while waiting for the next bus free state After reporting this interrupt, the result of the operation (automatic reselection response mode or user program operation) specified after reselection is reported. (2) Reselection already made and operation specified after reselection executed when command received This interrupt is reported after reporting the result of the specified operation.
65H	03H	(1) Length of group 6/7 CDB not set (2) Data to be sent was group 3/4 CDB This interrupt is reported without transferring the CDB.
67H	01H	Command Pause accepted in arbitration phase
70H	03H	BSY signal response received from target and nexus established but bus free state detected
82H	02H	More than specified selection/reselection timeout time elapsed This interrupt is reported after executing the specified number of retries.

Notes:

Note 1. Terminated concurrently with detection
Note 2. Reported when automatic receive mode set
Note 3. Reported when manual Reset ACK mode set
Note 4. Reported when automatic Reset ACK mode set
Note 5. Step code at the error is the same as one listed in the table.

(3) SELECT & 1-MSG & CMD, SELECT & N-Byte-MSG & CMD

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected while waiting bus free state
	01H	Reset condition detected during execution of arbitration phase
	02H	Reset condition detected during execution of selection phase
	03H	Reset condition detected during execution of the message phase (Note 1)
	04H	Reset condition detected during execution of command phase (Note 1)
2CH	03H	More than specified REQ/ACK timeout time elapsed during execution of message phase (Note 1)
	04H	More than specified REQ/ACK timeout time elapsed during execution of command phase (Note 1)
31H	03H	Bus free state detected during execution of message phase (Note 1)
	04H	Bus free state detected during execution of command phase (Note 1)
32H	03H	Target changed phase during execution of message phase (Note 1)
	04H	Target changed phase during execution of command phase (Note 1)
54H	03H	Phase when REQ signal first received was not message-out phase This interrupt is reported without negating the ATN signal.
	04H	The phase when the REQ signal was first received was not the command phase.
55H	03H	Phase when REQ signal first received other than in the message-out phase and one message received in response to message receive request from target (ATN signal negated) (Notes 2 & 5)
	04H	Phase when REQ signal first received other than in the command phase and one message received in response to message receive request from target (Notes 2 & 5)
56H	03H	Phase when REQ signal first received other than in the message-out phase and one status received in response to status receive request from target (ATN signal negated) (Notes 2 & 5)
	04H	Phase when REQ signal first received other than in the command phase and one status received in response to status receive request from target (Notes 2 & 5)

(Continued)

CODE	STEP	Interrupt Source and SPC Operation
5DH	03H	Phase when REQ signal first received other than in the message-out phase, and attempted to receive one message in response to message receive request from target but the message not received due to the data remaining in Receive MCS Buffer (ATN signal negated) (Note 2)
	04H	Phase when REQ signal first received other than in the command phase, and attempted to receive one message in response to message receive request from target but the message not received due to the data remaining in Receive MCS Buffer (ATN signal negated) (Note 2)
5EH	03H	Phase when REQ signal first received other than in the message-out phase, and attempted to receive one status in response to status receive request from target but the status not received due to the data remaining in Receive MCS Buffer (ATN signal negated) (Note 2)
	04H	Phase when REQ signal first received other than in the command phase, and attempted to receive one status in response to status receive request from target but the status not received due to the data remaining in Receive MCS Buffer (ATN signal negated) (Note 2)
60H	05H	Terminated normally (Note 3)
	0DH	Terminated normally and negated the last ACK signal (Note 4)
64H	00H	(1) Arbitration failed and reselection made while waiting for the next bus free state After reporting this interrupt, the result of the operation (automatic reselection response mode or user program operation) specified after reselection is reported. (2) When command received, reselection already made and operation specified after reselection executed This interrupt is reported after reporting the result of the specified operation.
	03H	Message 33 bytes or more long (only for **SELECT & 1-MSG & CMD**) Not send message.
65H	03H	MC byte register not set (00H) (only for **SELECT & N-Byte-MSG & CMD**)
	04H	(1) Length of group 6/7 CDB not set (2) Data to be sent group 3/4 CDB This interrupt is reported without transferring the CDB.
67H	01H	**Command Pause** accepted in arbitration phase
70H	03H	BSY signal response received from target and nexus established but bus free state detected
82H	02H	More than specified selection/reselection timeout time elapsed This interrupt is reported after executing the specified number of retries.

Notes:

Note 1. Terminated concurrently with detection

Note 2. Reported when automatic receive mode set

Note 3. Reported when manual Reset ACK mode set

Note 4. Reported when automatic Reset ACK mode set

Note 5. Step code at the error is the same as one listed in the table above.

(4) SELECT & 1-MSG, SELECT & N-Byte-MSG

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected while waiting bus free state
	01H	Reset condition detected during execution of arbitration phase
	02H	Reset condition detected during execution of selection phase
	03H	Reset condition detected during execution of message phase (Note 1)
2CH	03H	More than specified REQ/ACK timeout time elapsed (Note 1)
31H	03H	Bus free state was detected during execution of message phase (Note 1)
32H	03H	Target changed phase during execution of message phase (Note 1)
54H	03H	Phase when REQ signal first received not message-out phase This interrupt is reported without negating the ATN signal.
55H	03H	Phase when REQ signal first received other than in the message-out phase and one message received in response to message receive request from target (ATN signal negated) (Notes 2 & 5)
56H	03H	Phase when REQ signal first received other than in the message-out phase and one status received in response to status receive request from target (ATN signal negated) (Notes 2 & 5)
5DH	03H	Phase when REQ signal first received other than in the message-out phase, and attempted to receive one message received in response to message receive request from target but the message not received due to data remaining in Receive MCS Buffer (ATN signal negated) (Note 2)
5EH	03H	Phase when REQ signal first received other than in the message-out phase, and attempted to receive one status received in response to status receive request from target but the status not received due to data remaining in Receive MCS Buffer (ATN signal negated) (Note 2)
60H	04H	Terminated normally (Note 3)
	0CH	Terminated normally and negated the last ACK signal (Note 4)
64H	00H	(1) Arbitration failed and reselection made while waiting for the next bus free state After reporting this interrupt, the result of the operation (automatic reselection response mode or user program operation) specified after reselection is reported. (2) When command received, reselection already made and operation specified after reselection executed This interrupt is reported after reporting the result of the specified operation.
	03H	Message 33 bytes or more long (only for **SELECT & 1-MSG**)
65H	03H	MC byte register not set (00H) (only for **SELECT & N-Byte-MSG**)
67H	01H	**Command Pause** accepted in arbitration phase
70H	03H	BSY signal response received from target and nexus established but bus free state detected
82H	02H	More than specified selection/reselection timeout time elapsed This interrupt is reported after executing the specified number of retries.

Notes:

Note 1. Terminated concurrently with detection

Note 2. Reported when automatic receive mode set

Note 3. Reported when manual Reset ACK mode set

Note 4. Reported when automatic Reset ACK mode set

Note 5. Step code at the error is the same as one listed in the table above.

(5) SEND N-Byte-MSG

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before transfer
	01H	Reset condition detected during transfer (Note 1)
2CH	01H	More than specified REQ/ACK timeout time elapsed (Note 1)
31H	01H	Bus free state was detected during transfer (Note 1)
32H	01H	Target changed phase during transfer (Note 1)
54H	01H	Phase when REQ signal first received not message-out phase This interrupt is reported without negating the ATN signal.
55H	01H	Phase when REQ signal first received other than in the message-out phase and one message received in response to message receive request from target (ATN signal negated) (Notes 2 & 5)
56H	01H	Phase when REQ signal first received other than in the message-out phase and one status received in response to status receive request from target (ATN signal negated) (Notes 2 & 5)
5DH	01H	Phase when REQ signal first received other than in the message-out phase, and attempted to receive one message in response to message receive request from target but the message not received due to data remaining in Receive MCS Buffer (ATN signal negated) (Note 2)
5EH	01H	Phase when REQ signal first received other than in the message-out phase, and attempted to receive one status in response to status receive request from target but the status not received due to data remaining in Receive MCS Buffer (ATN signal negated) (Notes 2)
60H	02H	Terminated normally (Note 3)
	0AH	Terminated normally and negated the last ACK signal (Note 4)
65H	01H	MC byte register not set (00H)
70H	00H	Bus free state detected before transfer

Notes:

Note 1. Terminated concurrently with detection

Note 2. Reported when automatic receive mode set

Note 3. Reported when manual Reset ACK mode set

Note 4. Reported when automatic Reset ACK mode set

Note 5. Step code at the error is the same as one listed in the table above.

(6) SEND N-Byte-CMD

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before transfer
	01H	Reset condition detected during transfer (Note 1)
2CH	01H	More than specified REQ/ACK timeout time elapsed (Note 1)
31H	01H	Bus free state detected during transfer (Note 1)
32H	01H	Target changed phase during transfer (Note 1)
54H	01H	Phase when REQ signal first received not command phase
55H	01H	Phase when REQ signal first received other than in the command phase and one message received in response to message receive request from target (Notes 2 & 5)
56H	01H	Phase when REQ signal first received other than in the command phase and one status received in response to status receive request from target (Notes 2 & 5)
5DH	01H	Phase when REQ signal first received other than in the command phase, and attempted to receive one message in response to message receive request from target but the message not received due to data remaining in Receive MCS Buffer (Note 2)
5EH	01H	Phase when REQ signal first received other than in the command phase, and attempted to receive one status in response to status receive request from target but the status not received due to data remaining in Receive MCS Buffer (Note 2)
60H	02H	Terminated normally (Note 3)
	0AH	Terminated normally and negated the last ACK signal (Note 4)
65H	01H	MC byte register not set (00H)
70H	00H	Bus free state detected before transfer

Notes:

Note 1. Terminated concurrently with detection

Note 2. Reported when automatic receive mode set

Note 3. Reported when manual Reset ACK mode set

Note 4. Reported when automatic Reset ACK mode set

Note 5. Step code at the error is the same as one listed in the table above.

(7) RECEIVE N-Byte-MSG

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before transfer
	01H	Reset condition detected during transfer (Note 1)
24H	01H	Parity error detected in input data from SCSI bus (This INT is reported after receiving all the bytes data.)
2CH	01H	More than specified REQ/ACK timeout time elapsed (Note 1)
31H	01H	Bus free state was detected during transfer (Note 1)
32H	01H	Target changed phase during transfer (Note 1)
54H	01H	Phase when REQ signal first received not message-in phase
56H	01H	Phase when REQ signal first received other than in the message-in phase and one status received in response to status receive request from target (Notes 2 & 3)
5EH	01H	Phase when REQ signal first received other than in the message-in phase, and attempted to receive one status in response to status receive request from target but the status not received due to data remaining in Receive MCS Buffer (Note 2)
60H	02H	Terminated normally
65H	01H	MC byte register not set (00H)
6CH	01H	Message not received due to data remaining in Receive MCS Buffer
70H	00H	Bus free state detected before transfer

Notes:

Note 1. Terminated concurrently with detection

Note 2. Reported when automatic receive mode set

Note 3. Step code at the error is the same as one listed in the table above.

b. Discrete Commands

(1) All commands

CODE	STEP	Interrupt Source and SPC Operation
31H	00H	After the transfer completed, bus free state detected asserting ACK signal (at manual Reset ACK mode)
64H	00H	(1) New command received during execution of command (double receiving) This interrupt is reported after reporting the execution result of the first received command. (2) When command received, the interrupt source that occurs when SPC is ready already detected and preceded the interrupt report This interrupt is reported after reporting interrupt source (excluding 'REQ asserted') that occurs when SPC ready
65H	00H	(1) Command from target received when operating as initiator (2) Received command undefined (3) Commands issued for starting transfer before establishing nexus and for starting with selection establishing nexus
22H	00H	Parity error detected in register set by host MPU

(2) SELECT, SELECT WITH ATN

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected while waiting for bus free
	01H	Reset condition detected during execution of arbitration phase
	02H	Reset condition detected during execution of selection phase
60H	03H	Terminated normally
64H	00H	(1) Arbitration failed and reselection made while waiting for the next bus free state After reporting this interrupt, the result of the operation (automatic reselection response mode or user program operation) specified after reselection is reported. (2) When command received, reselection already made and operation specified after reselection executed This interrupt is reported after reporting the result of the specified operation.
67H	01H	**Command Pause** accepted in arbitration phase
82H	02H	More than specified selection/reselection timeout time elapsed This interrupt is reported after executing the specified number of retries.

(3) SET ATN, RESET ATN, SET ACK, RESET ACK

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before executing command
31H	00H	Bus free state detected before executing command (when this command issued after executing transfer command) (Note 1)
60H	01H	Terminated normally
	09H	Terminated normally and negated the last ACK signal (only for SET ATN command) (Note 2)
70H	00H	Bus free state detected before executing command (when a command issued after executing a command excluding the transfer)

Notes:

Note 1. Reported when manual Reset ACK mode set

Note 2. Reported when automatic Reset ACK mode set

(4) SEND DATA from MPU/DMA (with Padding)

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before transfer
	01H	Reset condition detected during transfer (Note 2)
21H	01H	Parity error detected in input/output data over DMA interface during DMA transfer (Note 1)
	09H	Parity error detected in input/output data over DMA interface during DMA transfer, and the last ACK signal negated after the transfer completed (Notes 1 & 6)
22H	01H	Parity error detected when accessing SCSI data register during program transfer (Notes 1& 8)
	09H	Parity error detected in input/output data over MPU interface during program transfer, and the last ACK signal negated after the transfer completed (Notes 1, 6, 8)
24H	01H	Parity error detected in input/output data over SCSI interface (Note 1)
	09H	Parity error detected in input/output data over SCSI interface, and the last ACK signal negated after the transfer completed (Notes 1 & 6)
25H	01H	Parity error detected in input/output data over DMA and SCSI interfaces during DMA transfer (Note 1)
	09H	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 (Notes 1 & 6)
26H	01H	Parity error detected in input/output data over MPU and SCSI interfaces during program transfer (Note 1)
	09H	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 (Notes 1 & 6)
29H	01H	REQ signal greater than specified offset value received in synchronous transfer (Note 1)
	09H	REQ signal greater than specified offset value received in synchronous transfer, and the last ACK signal negated after the transfer completed (Notes 1 & 6)
2AH	01H	SPC could not follow because period of received REQ signal short (Note 1) This interrupt may be reported even in asynchronous transfer.
	09H	SPC could not follow because period of received REQ signal short, and the last ACK signal negated after the transfer completed (Notes 1 & 6) This interrupt may be reported even in asynchronous transfer.
2BH	01H	REQ signal greater than specified offset value received in synchronous transfer and SPC could not follow because period of received REQ signal short (Note 1)
	09H	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 (Notes 1 & 6)

(Continued)

(4) SEND DATA from MPU/DMA (with Padding) (Continued)

CODE	STEP	Interrupt Source and SPC Operation
2CH	01H	More than specified REQ/ACK timeout time elapsed (only in asynchronous mode) (Note 2)
31H	01H	Bus free state detected during data transfer (Note 2)
32H	01H	Target changed phase during data transfer (Note 2)
54H	01H	Phase when REQ signal first received not data-out phase
55H	01H	Phase when REQ signal first received other than in the data-out phase and one message received in response to message receive request from target (Notes 3 & 7)
56H	01H	Phase when REQ signal first received other than in the data-out phase and one status received in response to status receive request from target (Notes 3 & 7)
5DH	01H	Phase when REQ signal first received other than in the data-out phase, and attempted to receive one message in response to message receive request from target but the message not received due to data remaining in Receive MCS Buffer (Note 3)
5EH	01H	Phase when REQ signal first received other than in the data-out phase, and attempted to receive one status in response to status receive request from target but the status not received due to data remaining in Receive MCS Buffer (Note 3)
60H	02H	Terminated normally (Note 5)
	0AH	Terminated normally and negated the last ACK signal (Note 6)
65H	01H	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.
67H	01H	**Command Pause** accepted during data transfer (Note 2)
70H	00H	Bus free state detected before transfer (Note 3)

Notes:

Note 1. The ATN signal is asserted to continue transfer.
Note 2. This interrupt is reported after the exit processing (for exit processing, see section 5.7).
Note 3. Reported only when automatic receive mode set
Note 5. Reported only when manual Reset ACK mode set
Note 6. Reported only when automatic Reset ACK mode set (except for padding transfer being executed)
Note 7. Step code at the error is the same as one listed in the table above.
Note 8. When parity error is detected in the other internal register access, interrupt code/step code "22H/00H' is also reported following this interrupt/step code.

(5) RECEIVE DATA to MPU/DMA (with Padding)

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before transfer
	01H	Reset condition detected during transfer (Note 2)
21H	01H	Parity error detected in input/output data over DMA interface during DMA transfer (Note 1)
	09H	Parity error detected in input/output data over DMA interface during DMA transfer, and the last ACK signal negated after the transfer completed (Notes 1 & 6)
22H	01H	Parity error detected when accessing SCSI data register during program transfer (Notes 1& 8)
	09H	Parity error detected in input/output data over MPU interface during program transfer, and the last ACK signal negated after the transfer completed (Notes 1, 6, 8)
24H	01H	Parity error detected in input/output data over SCSI interface (Note 1)
	09H	Parity error detected in input/output data over SCSI interface, and the last ACK signal negated after the transfer completed (Notes 1 & 6)
25H	01H	Parity error detected in input/output data over DMA and SCSI interfaces during DMA transfer (Note 1)
	09H	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 (Notes 1 & 6)
26H	01H	Parity error detected in input/output data over MPU and SCSI interfaces during program transfer (Note 1)
	09H	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 (Notes 1 & 6)
29H	01H	REQ signal greater than specified offset value received in synchronous transfer (Note 1)
	09H	REQ signal greater than specified offset value received in synchronous transfer, and the last ACK signal negated after the transfer completed (Notes 1 & 6)
2AH	01H	SPC could not follow because period of received REQ signal short (Note 1) This interrupt may be reported even in asynchronous transfer.
	09H	SPC could not follow because period of received REQ signal short, and the last ACK signal negated after the transfer completed (Notes 1 & 6) This interrupt may be reported even in asynchronous transfer.
2BH	01H	REQ signal greater than specified offset value received in synchronous transfer and SPC could not follow because period of received REQ signal short (Note 1)
	09H	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 (Notes 1 & 6)

(Continued)

CODE	STEP	Interrupt Source and SPC Operation
2CH	01H	More than specified REQ/ACK timeout time elapsed (only in asynchronous mode) (Note 2)
31H	01H	Bus free state detected during data transfer (Note 2)
32H	01H	Target changed phase during data transfer (Note 2)
54H	01H	Phase when REQ signal first received not data-in phase
55H	01H	Phase when REQ signal first received other than in the data-in phase and one message received in response to message receive request from target (Notes 3 & 7)
56H	01H	Phase when REQ signal first received other than in the data-in phase and one status received in response to status receive request from target (Notes 3 & 7)
5DH	01H	Phase when REQ signal first received other than in the data-in phase, and attempted to receive one message in response to message receive request from target but the message not received due to data remaining in Receive MCS Buffer (Note 3)
5EH	01H	Phase when REQ signal first received other than in the data-in phase, and attempted to receive one status in response to status receive request from target but the status not received due to data remaining in Receive MCS Buffer (Note 3)
60H	02H	Terminated normally (Note 5)
	0AH	Terminated normally and negated the last ACK signal (Note 6)
65H	01H	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.
67H	01H	**Command Pause** accepted during data phase (Note 2)
70H	00H	Bus free state detected before transfer (Note 3)

Notes:

Note 1. The ATN signal is asserted to continue transfer.
Note 2. This interrupt is reported after the exit processing (for exit processing, see section 5.7).
Note 3. Reported only when automatic receive mode set
Note 5. Reported only when manual Reset ACK mode set
Note 6. Reported only when automatic Reset ACK mode set (except for padding transfer being executed)
Note 7. Step code at the error is the same as one listed in the table above.
Note 8. When parity error is detected in the other internal register access, interrupt code/step code "22H/00H" is also reported following this interrupt/step code.

(6) SEND 1-MSG, SEND 1-MSG with ATN

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before transfer
	01H	Reset condition detected during transfer (Note 1)
2CH	01H	More than specified REQ/ACK timeout time elapsed (Note 1)
31H	01H	Bus free state detected during transfer (Note 1)
32H	01H	Target changed phase during transfer (Note 1)
54H	01H	Phase when REQ signal first received not message-out phase This interrupt is reported without negating the ATN signal.
55H	01H	Phase when REQ signal first received other than in the message-out phase and one message received in response to message receive request from target (ATN signal negated) (notes 2 & 5)
56H	01H	Phase when REQ signal first received other than in the message-out phase and one status received in response to status receive request from target (ATN signal negated) (Notes 2 & 5)
5DH	01H	Phase when REQ signal first received other than in the message-out phase, and attempted to receive one message in response to message receive request from target but the message not received due to data remaining in Receive MCS Buffer (ATN signal negated) (Note 2)
5EH	01H	Phase when REQ signal first received other than in the message-out phase, and attempted to receive one status in response to status receive request from target but the status not received due to data remaining in Receive MCS Buffer (ATN signal negated) (Note 2)
60H	02H	Terminated normally (Note 3)
	0AH	Terminated normally and negated the last ACK signal (Note 4)
64H	01H	Message 33 bytes or more long (Message is not sent.)
70H	00H	Bus free state detected before transfer

Notes:

Note 1. Terminated concurrently with detection

Note 2. Reported when automatic receive mode set

Note 3. Reported when manual Reset ACK mode set

Note 4. Reported when automatic Reset ACK mode set

Note 5. Step code at the error is the same as one listed in the table above.

(7) RECEIVE MSG

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before transfer
	01H	Reset condition detected during transfer (Note 1)
24H	01H	Parity error detected in input data from SCSI bus (1) If the received message is a one-byte or two-byte message, this interrupt is reported after receiving all bytes. (2) If the received message is an extended message, this interrupt is reported after receiving two bytes when they are detected at the first or second byte and after receiving all bytes when they are detected at the third and later byte.
2CH	01H	More than specified REQ/ACK timeout time elapsed (Note 1)
31H	01H	Bus free state was detected during transfer (Note 1)
32H	01H	Target changed phase during transfer (Note 1)
54H	01H	Phase when REQ signal first received not message-in phase
56H	01H	Phase when REQ signal first received other than in the message-in phase and one status received in response to message receive request from target (Notes 2 and 3)
5EH	01H	Phase when REQ signal first received other than in the message-in phase, and attempted to receive one status in response to status receive request from target but the status not received due to data remaining in Receive MCS Buffer (Note 2)
60H	02H	Terminated normally
67H	01H	Received message was extended message and the length was 33 bytes or more long. The message is received up to the second byte.
6CH	01H	Message not received due to data remaining in Receive MCS Buffer
70H	00H	Bus free state detected before transfer

Notes:

Note 1. Terminated concurrently with detection

Note 2. Reported when automatic receive mode set

Note 3. Step code at the error is the same as one listed in the table above

(8) SEND CMD

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before transfer
	01H	Reset condition detected during transfer (Note 1)
2CH	01H	More than specified REQ/ACK timeout time elapsed (Note 1)
31H	01H	Bus free state detected during transfer (Note 1)
32H	01H	Target changed phase during transfer (Note 1)
54H	01H	Phase when REQ signal first received not command phase
55H	01H	Phase when REQ signal first received other than in the command phase and one message received in response to message receive request from target (Notes 2 & 5)
56H	01H	Phase when REQ signal first received other than in the command phase and one status received in response to status receive request from target (Notes 2 & 5)
5DH	01H	Phase when REQ signal first received other than in the command phase, and attempted to receive one message in response to message receive request from target but the message not received due to data remaining in Receive MCS Buffer (Note 2)
5EH	01H	Phase when REQ signal first received other than in the command phase, and attempted to receive one status in response to status receive request from target but the status not received due to data remaining in Receive MCS Buffer (Note 2)
60H	02H	Terminated normally (Note 3)
	0AH	Terminated normally and negated the last ACK signal (Note 4)
65H	01H	(1) Length of group 6/7 CDB not set (2) Data to be sent group 3/4 CDB This interrupt is reported without transferring the CDB
70H	00H	Bus free state detected before transfer

Notes:

Note 1. Terminated concurrently with detection

Note 2. Reported when automatic receive mode set

Note 3. Reported when manual Reset ACK mode set

Note 4. Reported when automatic Reset ACK mode set

Note 5. Step code at the error is the same as one listed in the table above

(9) RECEIVE STATUS

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before transfer
	01H	Reset condition detected during transfer (Note 1)
24H	01H	Parity error detected in input data from SCSI bus (terminated after one-byte transfer) (Note 3)
	09H	Parity error detected in input data from SCSI bus, and negated the ACK signal after the transfer completed (Note 4)
2CH	01H	More than specified REQ/ACK timeout time elapsed (Note 1)
54H	01H	Phase when REQ signal first received not status phase
55H	01H	Phase when REQ signal first received other than in the status phase and one message received in response to message receive request from target (Notes 2 and 5)
5DH	01H	Phase when REQ signal first received other than in the status phase, and attempted to receive one message in response to message receive request from target but the message not received due to data remaining in Receive MCS Buffer (Note 2)
60H	02H	Terminated normally (Note 3)
	0AH	Terminated normally and negated the last ACK signal (Note 4)
6CH	01H	Status not received due to data remaining in Receive MCS Buffer
70H	00H	Bus free state detected before transfer

Notes:

Note 1. Terminated concurrently with detection

Note 2. Reported when automatic receive mode set

Note 3. Reported when manual Reset ACK mode set

Note 4. Reported when automatic Reset ACK mode set

Note 5. Step code at the error is the same as one listed in the table above

B.2. Target Commands

a. Sequential Commands

(1) All commands

CODE	STEP	Interrupt Source and SPC Operation
64H	00H	(1) New command received during execution of command (double receiving) This interrupt is reported after reporting the execution result of the first received command. (2) Command not executed because interrupt source occurred before executing command If the interrupt source that is reported when the SPC is ready occurs, this interrupt is reported after reporting the interrupt source.
65H	00H	(1) Command from initiator received when operating as target (2) Received command undefined (3) Commands issued for starting transfer before establishing nexus and for starting reselection after establishing nexus
22H	00H	Parity error detected in register set by host MPU

(2) RESELECT & 1-MSG, RESELECT & N-Byte-MSG

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected while waiting bus free state
	01H	Reset condition detected during execution of arbitration phase
	02H	Reset condition detected during execution of reselection phase
	03H	Reset condition detected during execution of message phase (Note 1)
2CH	03H	More than specified REQ/ACK timeout time elapsed (Note 1)
60H	04H	Terminated normally
61H	04H	Command normally terminated and attention condition generated by initiator detected (Note 2)
62H	04H	Command normally terminated, attention condition generated by initiator detected, and one message received (Note 3)
64H	00H	(1) Arbitration failed and selection made while waiting for the next bus free state After reporting this interrupt, the result of the operation (automatic selection response mode or user program operation) specified after selection is reported. (2) When command received, selection already made and operation specified after selection executed This interrupt is reported after reporting the result of the specified operation.
	03H	Message 33 bytes or more long (only for **RESELECT & 1-MSG**) Message not sent
65H	03H	MC byte register not set (00H) (only for **RESELECT & N-Byte-MSG**)
66H	04H	Command normally terminated, attention condition generated by initiator detected, and attempted to receive one message but the message not received due to data remaining in Receive MCS Buffer (Note 4)
67H	01H	**Command Pause** accepted in arbitration phase
82H	02H	More than specified selection/reselection timeout time elapsed This interrupt is reported after executing the specified number of retries.

Notes:

Note 1. Terminated concurrently with detection

Note 2. Reported when automatic receive mode not set

Note 3. Reported when automatic receive mode set. Also step code at the error is the same as one listed in the table above

Note 4. Reported when automatic receive mode set.

(3) RESELECT & 1-MSG & TERMINATE, RESELECT & N-Byte-MSG & TERMINATE

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reste conditional detected while waiting bus free state
	01H	Reset condition detected during execution of arbitration phase
	02H	Reset condition detected during execution of reselection phase
	03H	Reset condition detected during execution of message-in phase (Note 1)
	04H	Reset condition detected during execution of status phase (Note 1)
	05H	Reset condition detected during execution of message-in phase (Note 1)
2CH	03H	More than specified REQ/ACK timeout time elapsed during execution of message-in phase (Note 1)
	04H	More than specified REQ/ACK timeout time elapsed during execution of status phase (Note 1)
	05H	More than specified REQ/ACK timeout time elapsed during execution of message-in phase (Note 1)
42H	03H	Attention condition generated by initiator detected when message-in phase normally terminated (Note 2)
	04H	Attention condition generated by initiator detected when status phase normally terminated (Note 2)
	05H	Attention condition generated by initiator detected when message-in phase normally terminated (Note 2)
43H	03H	Attention condition generated by initiator detected and one message received when message-in phase normally terminated (Note 3)
	04H	Attention condition generated by initiator detected and one message received when status phase normally terminated (Note 3)
	05H	Attention condition generated by initiator detected and one message received when message-in phase normally terminated (Note 3)
47H	03H	Attention condition generated by initiator detected when message-in phase normally terminated, and attempted to receive one message but the message not received due to data remaining in Receive MCS Buffer (Note 3)
	04H	Attention condition generated by initiator detected when status phase normally terminated, and attempted to receive one message but the message not received due to data remaining in Receive MCS Buffer (Note 3)
	05H	Attention condition generated by initiator detected when message-in phase normally terminated, and attempted to receive one message but the message not received due to data remaining in Receive MCS Buffer (Note 3)

(Continued)

CODE	STEP	Interrupt Source and SPC Operation
60H	06H	Terminated normally
64H	00H	(1) Arbitration failed and selection made while waiting for the next bus free state After reporting this interrupt, the result of the operation (automatic selection response mode or user program operation) specified after selection is reported. (2) When command received, selection already made and operation specified after selection executed This interrupt is reported after reporting the result of the specified operation.
	03H	Message 33 bytes or more long (only for **RESELECT** & **1-MSG** & **TERMINATE**) Message not transferred.
65H	03H	MC byte register not set (00H) (only for **RESELECT** & **N-Byte-MSG** & **TERMINATE**)
67H	01H	Command Pause accepted in arbitration phase
82H	02H	More than specified selection/reselection timeout time elapsed This interrupt is reported after executing the specified number of retries.

Notes:

Note 1. Terminated concurrently with detection

Note 2. Reported when automatic receive mode not set

Note 3. Reported when automatic receive mode set

(4) RESELECT & 1-MSG & LINK-TERMINATE, RESELECT & N-Byte-MSG & LINK-TERMINATE

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected while waiting bus free state
	01H	Reset condition detected during execution of arbitration phase
	02H	Reset condition detected during execution of reselection phase
	03H	Reset condition detected during execution of message-in phase (Note 1)
	04H	Reset condition detected during execution of status phase (Note 1)
	05H	Reset condition detected during execution of message-in phase (Note 1)
2CH	03H	More than specified REQ/ACK timeout time elapsed during execution of message-in phase (Note 1)
	04H	More than specified REQ/ACK timeout time elapsed during execution of status phase (Note 1)
	05H	More than specified REQ/ACK timeout time elapsed during execution of message-in phase (Note 1)
42H	03H	Attention condition generated by initiator detected when message-in phase normally terminated (Note 2)
	04H	Attention condition generated by initiator detected when status phase normally terminated (Note 2)
43H	03H	Attention condition generated by initiator detected and one message received when message-in phase normally terminated (Note 3)
	04H	Attention condition generated by initiator detected and one message received when status phase normally terminated (Note 3)
47H	03H	Attention condition generated by initiator detected when message-in phase normally terminated, and attempted to receive one message but the message not received due to data remaining in Receive MCS Buffer (Note 3)
	04H	Attention condition generated by initiator detected when status phase normally terminated, and attempted to receive one message but the message not received due to data remaining in Receive MCS Buffer (Note 3)

(Continued)

CODE	STEP	Interrupt Source and SPC Operation
60H	06H	Terminated normally
61H	06H	Command phase normally terminated (message-in phase normally terminated) and attention condition generated by initiator detected (Note 2)
62H	06H	Command phase normally terminated (message-in phase normally terminated), attention condition generated by initiator detected, and one message received (Note 3)
64H	00H	(1) Arbitration failed and selection made while waiting for the next bus free state After reporting this interrupt, the result of the operation (automatic selection response mode or user program operation) specified after selection is reported. (2) When command received, selection already made and operation specified after selection executed This interrupt is reported after reporting the result of the specified operation.
	03H	Message 33 bytes or more long (only for **RESELECT** & **1-MSG** & **LINK-TERMINATE**) Message is not transferred.
65H	03H	MC byte register not set (00H) (only for **RESELECT** & **N-Byte-MSG** & **LINK-TERMINATE**)
66H	06H	Command normally terminated (message-in phase normally terminated), attention condition generated by initiator detected, and attempted to receive one message but the message not received due to data remaining in Receive MCS Buffer (Note 4)
67H	01H	**Command Pause** accepted in arbitration phase
82H	02H	More than specified selection/reselection timeout time elapsed This interrupt is reported after executing the specified number of retries.

Notes:

Note 1. Terminated concurrently with detection

Note 2. Reported when automatic receive mode not set

Note 3. Reported when automatic receive mode set Also, step code at the error is the same as one listed in the table above)

Note 4. Reported when automatic receive mode set

(5) TERMINATE

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before transfer
	01H	Reset condition detected during execution of status phase (Note 1)
	02H	Reset condition detected during execution of message-in phase (Note 1)
2CH	01H	More than specified REQ/ACK timeout time elapsed during execution of status phase (Note 1)
	02H	More than specified REQ/ACK timeout time elapsed during execution of message-in phase (Note 1)
42H	01H	Attention condition generated by initiator detected when status phase normally terminated (Note 2)
	02H	Attention condition generated by initiator detected when message-in phase normally terminated (Note 2)
43H	01H	Attention condition generated by initiator detected and one message received when status phase normally terminated (Note 3)
	02H	Attention condition generated by initiator detected and one message received when message-in phase normally terminated (Note 3)
47H	01H	Attention condition generated by initiator detected when status phase normally terminated, and attempted to receive one message but the message not received due to data remaining in Receive MCS Buffer (Note 3)
	02H	Attention condition generated by initiator detected when message-in phase normally terminated, and attempted to receive one message but the message not received due to data remaining in Receive MCS Buffer (Note 3)
60H	03H	Terminated normally

Notes:
1. Terminated concurrently with detection
2. Reported when automatic receive mode not set
3. Reported when automatic receive mode set

(6) LINK-TERMINATE

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before transfer
	01H	Reset condition detected during execution of status phase (Note 1)
	02H	Reset condition detected during execution of message-in phase (Note 2)
2CH	01H	More than specified REQ/ACK timeout time elapsed during execution of status phase (Note 1)
	02H	More than specified REQ/ACK timeout time elapsed during execution of message-in phase (Note 1)
42H	01H	Attention condition generated by initiator detected when status phase normally terminated (Note 2)
43H	01H	Attention condition generated by initiator detected and one message received when status phase normally terminated (Note 3)
47H	01H	Attention condition generated by initiator detected when status phase normally terminated, and attempted to receive one message but the message not received due to data remaining in Receive MCS Buffer (Note 3)
60H	03H	Terminated normally
61H	03H	Command normally terminated (message-in phase normally terminated) and attention condition generated by initiator detected (Note 2)
62H	03H	Command normally terminated (message-in phase normally terminated), attention condition generated by initiator detected, and one message received (Note 3)
66H	06H	Command normally terminated (message-in phase normally terminated), attention condition generated by initiator detected, and attempted to receive one message but the message not received due to data remaining in Receive MCS Buffer (Note 4)

Notes:

Note 1. Terminated concurrently with detection

Note 2. Reported when automatic receive mode not set

Note 3. Reported when automatic receive mode set Also, step code at the error is the same as one listed in the table above

Note 4. Reported when automatic receive mode set

(7) DISCONNECT SEQUENCE, DISCONNECT SEQUENCE 2

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before transfer
	01H	Reset condition detected during transfer (Note 1)
	02H	Reset condition detected after transfer
2CH	01H	More than specified REQ/ACK timeout time elapsed (Note 1)
42H	01H	Attention condition generated by initiator detected when message-in phase normally terminated (Note 2)
43H	01H	Attention condition generated by initiator detected and one message received when message-in phase normally terminated (Note 3)
47H	01H	Attention condition generated by initiator detected when message-in phase normally terminated, and attempted to receive one message but the message not received due to data remaining in Receive MCS Buffer (Note 3)
60H	02H	Terminated normally

Notes:
1. Terminated concurrently with detection
2. Reported when automatic receive mode not set
3. Reported when automatic receive mode set

(8) SEND N-Byte-MSG

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before transfer
	01H	Reset condition detected during transfer (Note 1)
2CH	01H	More than specified REQ/ACK timeout time elapsed (Note 1)
60H	02H	Terminated normally
61H	02H	Command terminated normally (Message-in phase terminated normally), and Attention condition generated by initiator detected (Note 2)
62H	02H	Command terminated normally (Message-in phase terminated normally), attention condition generated by initiator detected and one message received (Note 3)
65H	01H	MC byte register not set (00H)
66H	02H	Command terminated normally (Message-in phase terminated normally), attention condition generated by initiator detected, and attempted to receive one message but the message not received due to data remaining in Receive MCS Buffer (Note 3)

Notes:

Note 1. Terminated concurrently with detection

Note 2. Reported when automatic receive mode not set

Note 3. Reported when automatic receive mode set Also, step code at the error is the same as one listed in the table above

(9) RECEIVE N-Byte-CMD

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before transfer
	01H	Reset condition detected during transfer (Note 1)
24H	01H	Parity error detected in input data from SCSI bus (Note 1)
2CH	01H	More than specified REQ/ACK timeout time elapsed (Note 1)
60H	02H	Terminated normally
61H	02H	Command normally terminated and attention condition generated by initiator detected (Note 2)
62H	02H	Command normally terminated, attention condition generated by initiator detected and one message received (Note 3)
65H	01H	MC byte register not set (00H)
6CH	01H	Message not received due to data remaining in Receive MCS Buffer

Notes:

1. Terminated concurrently with detection
2. Reported when automatic receive mode not set
3. Reported when automatic receive mode set Also, step code at the error is the same as one listed in the table above

(10) RECEIVE N-Byte-MSG

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before transfer
	01H	Reset condition detected during transfer (Note 1)
24H	01H	Parity error detected in input data from SCSI bus (Note 1)
2CH	01H	More than specified REQ/ACK timeout time elapsed (Note 1)
60H	02H	Terminated normally
61H	02H	Command normally terminated and attention condition generated by initiator detected (Note 2)
65H	01H	MC byte register not set (00H)
6CH	01H	Message not received due to data remaining in Receive MCS Buffer

Notes:

1. Terminated concurrently with detection
2. Reported regardless of the automatic receive mode setting

b. Discrete Commands

(1) All commands

CODE	STEP	Interrupt Source and SPC Operation
64H	00H	(1) New command received during execution of command (double receiving) This interrupt is reported after reporting the execution result of the first received command. (2) When command received, the interrupt source that occurs when SPC is ready was already detected and precedes the interrupt report source. This interrupt is reported after reporting the interrupt source reported when the SPC is ready.
65H	00H	(1) Command from initiator received when operating as target (2) Received command undefined (3) Commands issued for starting transfer before establishing nexus and for starting reselection after establishing nexus
22H	00H	Parity error detected in register set by host MPU

(2) RESELECT

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected while waiting bus free state
	01H	Reset condition detected during execution of arbitration phase
	02H	Reset condition detected during execution of reselection phase
60H	03H	Terminated normally
64H	00H	(1) Arbitration failed and selection made while waiting for the next bus free state After reporting this interrupt, the result of the operation (automatic selection response mode or user program operation) specified after selection is reported. (2) When **RESELECT** command received, selection already made and operation specified after selection executed This interrupt is reported after reporting the result of the specified operation.
67H	01H	**Command Pause** accepted in arbitration phase
82H	02H	More than specified selection/reselection timeout time elapsed This interrupt is reported after executing the specified number of retries.

(3) SET REQ, RESET REQ, DISCONNECT

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before executing command
60H	01H	Terminated normally

(4) SEND DATA from MPU/DMA, RECEIVE DATA to MPU/DMA

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before transfer
	01H	Reset condition detected during transfer (Note 1)
21H	01H	Parity error detected in input/output data over DMA interface during DMA transfer (Note 2)
22H	01H	Parity error detected in input/output data over MPU interface during program transfer (Notes 2 & 5)
24H	01H	Parity error detected in input/output data from and to SCSI bus (Note 2)
25H	01H	Parity error detected in input/output data over DMA and SCSI interfaces during DMA transfer (Note 2)
26H	01H	Parity error detected in input/output data from MPU and SCSI DUS during MPU transfer (Note 2)
29H	01H	ACK signal greater than output REQ signal received during synchronous transfer (Note 1)
2AH	01H	SPC could not follow because period of received ACK signal short (Note 1) This interrupt may be reported even in asynchronous transfer.
2BH	01H	During synchronous transfer, ACK signal greater than output REQ signal received and SPC could not follow because period of received ACK signal short (Note 1)
2CH	01H	More than specified REQ/ACK timeout time elapsed (Note 1)
42H	01H	Attention condition generated by initiator to terminate transfer at transfer block boundary (Note 3)
43H	01H	Attention condition generated by initiator to stop transfer at transfer block boundary and receive one message (Note 4)
47H	01H	Attention condition generated by initiator to stop transfer at transfer block boundary, and attempted to receive one message but the message not received due to data remaining in Receive MCS Buffer (Note 4)
60H	02H	Terminated normally
61H	02H	Command normally terminated and attention condition generated by initiator detected (Note 3)
62H	02H	Command normally terminated, attention condition generated by initiator detected, and one message received (Note 4)
65H	01H	Set value of data block register or data byte register was 0
66H	02H	Command normally terminated, attention condition generated by initiator detected, and attempted to receive one message but the message not received due to data remaining in Receive MCS Buffer (Note 4)
67H	01H	**Command Pause** accepted (Note 1)

Notes:
Note 1. This interrupt is reported after the exit processing. (For the exit processing, see section 5.7.)
Note 2. In byte stop mode, terminated concurrently with detection
In block stop mode, terminated with transfer of block where parity error detected
Note 3. Reported when automatic receive mode not set
Note 4. Reported when automatic receive mode set Also, step code at the error is the same as one listed in the table above
Note 5. When parity error is detected in the other internal register access, interrupt code/Step code '22H' + '00H' is also reported following this interrupt code/step code

(5) SEND 1-MSG

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before transfer
	01H	Reset condition detected during transfer (Note 1)
2CH	01H	More than specified REQ/ACK timeout time elapsed (Note 1)
60H	02H	Terminated normally
61H	02H	Command normally terminated and attention condition generated by initiator detected (Note 2)
62H	02H	Command normally terminated, attention condition generated by initiator detected, and one message received (Note 3)
64H	01H	Message 33 bytes or more long (Message is not transferred.)
66H	02H	Command normally terminated, attention condition generated by initiator detected, and attempted to receive one message but the message not received due to data remaining in Receive MCS Buffer (Note 3)

Notes:
Note 1. Terminated concurrently with detection
Note 2. Reported when automatic receive mode not set
Note 3. Reported when automatic receive mode set Also, step code at the error is the same as one listed in the table above

(6) RECEIVE MSG

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before transfer
	01H	Reset condition detected during transfer (Note 1)
24H	01H	Parity error detected in input data from SCSI bus (Note 1)
2CH	01H	More than specified REQ/ACK timeout time elapsed (Note 1)
60H	02H	Terminated normally
61H	02H	Command normally terminated and attention condition generated by initiator detected (Note 2)
6CH	01H	Message not received due to data remaining in Receive MCS Buffer
67H	01H	Received message 33 bytes or more long The message is received up to second byte data and terminated.

Notes:

1. Terminated concurrently with detection
2. Reported regardless of automatic receive mode setting

(7) SEND STATUS

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before transfer
	01H	Reset condition detected during transfer (Note 1)
2CH	01H	More than specified REQ/ACK timeout time elapsed (Note 1)
60H	02H	Terminated normally
61H	02H	Command normally terminated and attention condition generated by initiator detected. (Note 2)
62H	02H	Command normally terminated, attention condition generated by initiator detected, and one message received (Note 3)
66H	02H	Command normally terminated, attention condition generated by initiator detected, and attempted to receive one message but the message not received due to data remaining in Receive MCS Buffer (Note 3)

Notes:
1. Terminated concurrently with detection
2. Reported when automatic receive mode not set
3. Reported when automatic receive mode set Also, step code at the error is the same as one listed in the table above

(8) RECEIVE CMD

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before transfer
	01H	Reset condition detected during transfer (Note 1)
24H	01H	Parity error detected in input data from SCSI bus (Note 1)
2CH	01H	More than specified REQ/ACK timeout time elapsed (Note 1)
60H	02H	Terminated normally
61H	02H	Command normally terminated and attention condition generated by initiator detected (Note 2)
62H	02H	Command normally terminated, attention condition generated by initiator detected, and one message received (Note 3)
65H	01H	(1) Group 6/7 command not set (2) Group 3/4 command received The first byte was received and terminated.
6CH	01H	Message not received due to data remaining in Receive MCS Buffer

Notes:
Note 1. Terminated concurrently with detection

Note 2. Reported when automatic receive mode not set

Note 3. Reported when automatic receive mode set Also, step code at the error is the same as one listed in the table above.

B.3. Common Commands

(1) All commands

CODE	STEP	Interrupt Source and SPC Operation
64H	00H	(1) New command received during execution of command (double receiving) This interrupt is reported after reporting the execution result of the command received first received. (2) The command could not be executed because the interrupt source occurred before executing the command. If the interrupt source that is reported when the SPC is ready occurs, this interrupt is reported after reporting the interrupt source.
65H	00H	Received command undefined
22H	00H	Parity error detected in register set by host MPU

(2) TRANSFER RESET, SET UP REG

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before executing command
60H	01H	Terminated normally
70H	00H	Bus free state detected before executing command

(3) SCSI RESET

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before executing command
60H	01H	Terminated normally
70H	00H	Bus free state detected before executing command

(4) INIT DIAG START, TARG DIAG START

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before executing command
65H	01H	Issued DIAG START command after nexus established
68H	01H	Self-diagnosis performed with good result
69H	01H	Self-diagnosis performed with no-good result

(5) DIAG END

CODE	STEP	Interrupt Source and SPC Operation
01H	01H	Reset condition detected after executing command
60H	01H	Terminated normally
65H	00H	(1) DIAG END command issued even though DIAG START was not issued (2) DIAG END command issued after issuing DIAG START and reporting 'DIAG No-GOOD interrupt.

(6) COMMAND PAUSE

CODE	STEP	Interrupt Source and SPC Operation
67H	01H	Command Pause accepted (If Command Pause accepted in phases other than arbitration and data phases, interrupt is not reported.)

(7) SET RST, RESET RST

CODE	STEP	Interrupt Source and SPC Operation
60H	01H	Terminated normally

B.4. Automatic Selection/Reselection Response

a. Automatic Reselection Response

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before nexus
	01H	After reselection phase, phase switched to message-in phase but reset condition detected during transfer (Note 1)
24H	01H	After reselection phase, SPC switched to message-in phase but parity error detected in input data from SCSI bus (1) If received message one-byte or two-byte message, interrupt reported after receiving all bytes (2) If received message extended message, interrupt reported after receiving two bytes when detected at first or second byte, and after receiving all bytes when detected at third and later byte
2CH	01H	After reselection phase, SPC switched to message-in phase but more than specified REQ/ACK timeout time elapsed during transfer (Note 1)
31H	01H	After reselection phase, SPC switched to message-in phase but bus free state detected during transfer (Note 1)
32H	01H	After reselection phase, SPC switched to message-in phase but target changed phase during transfer (Note 1)
64H	00H	(1) Arbitration failed and reselection made while waiting for the next bus free state After reporting this interrupt, the result of the operation (automatic selection response mode or user program operation) specified after reselection is reported. (2) When command received, reselection already made and operation specified after reselection executed This interrupt is reported after reporting the result of the specified operation.
67H	01H	After reselection phase, SPC switched to message-in phase but received message was extended message and 33 bytes or more long The message is received up to the second byte.
70H	00H	Nexus established but bus free state detected before transfer
90H	00H	SPC reselected as initiator by target (Note 2)
94H	00H	After SPC reselected as initiator by target, it switched to message-in phase and attempted to receive one message but the message not received due to data remaining in Receive MCS Buffer
B0H	01H	SPC reselected as initiator by target, after reselection phase, SPC switched to a phase other than message-in phase
B1H	02H	SPC reselected as initiator by target, after reselection phase, SPC switched to message-in phase, and one message received

Notes:

Note 1. Terminated concurrently with detection

Note 2. This interrupt/step code is reported even if the automatic reselection response mode is not set.

b. Automatic Selection Response

CODE	STEP	Interrupt Source and SPC Operation
01H	00H	Reset condition detected before nexus
	81H	After selection phase, SPC switched to message-out phase but reset condition detected during transfer (Note 1)
	82H	After selection phase, identify message received and attempt made to receive one message, but reset condition detected during transfer (Note 1)
	92H	After selection phase, identify message received and attempt made to receive one command, but reset condition detected during transfer (Note 1)
	93H	After selection phase, identify message and one command received, ATN signal from initiator detected, and SPC switched to message-in phase, but reset condition detected during transfer (Note 1)
	B1H	After selection phase, SPC switched to command phase but reset condition detected during transfer (Note 1)
	B2H	After selection phase, one command received, ATN signal from initiator detected, and SPC switched to message-in phase, but reset condition detected during transfer (Notes 1, 2, & 4)
24H	81H	After selection phase, SPC switched to message-out phase but parity error detected in input data from SCSI bus (Note 1)
	82H	After selection phase, identify message received and attempt made to receive one message, but parity error detected in input data from SCSI bus (Note 1)
	92H	After selection phase, identify message received and attempt made to receive one command, but parity error detected in input data from SCSI bus (Note 1)
	93H	After selection phase, identify message and one command received, ATN signal from initiator detected, and SPC switched to message-in phase, but parity error detected in input data from SCSI bus (Note 1)
	B1H	After selection phase, SPC switched to command phase but a parity error detected in input data from SCSI bus (Notes 1 and 2)
	B2H	After selection phase, one command received, ATN signal from initiator detected, and SPC switched to message-in phase, but parity error detected in input data from SCSI bus (Notes 1, 2, & 4)
2CH	81H	After selection phase, SPC switched to message-out phase but more than specified REQ/ACK timeout time elapsed (Note 1)
	82H	After selection phase, identify message received and attempt made to receive one message, but more than specified REQ/ACK timeout time elapsed (Note 1)
	92H	After selection phase, identify message received and attempt made to receive one command, but more than specified REQ/ACK timeout time elapsed (Note 1)
	93H	After selection phase, identify message and one command received, ATN signal from initiator detected, and SPC switched to message-in phase, but more than specified REQ/ACK timeout time elapsed (Note 1)

(Continued)

CODE	STEP	Interrupt Source and SPC Operation
2CH	B1H	After selection phase, SPC switched to command phase but more than specified REQ/ACK timeout time elapsed (Notes 1 and 2)
	B2H	After selection phase, one command received, ATN signal from initiator detected, and SPC switched to message-in phase, but more than specified REQ/ACK time-out time elapsed (Notes 1, 2, & 4)
64H	00H	(1) Arbitration failed and selection made while waiting for the next bus free state After reporting this interrupt, the result of the operation (automatic selection response mode or user program operation) specified after selection is reported. (2) When command received, selection already made and operation specified after selection executed This interrupt is reported after reporting the result of the specified operation.
67H	81H	After selection phase, SPC switched to message-out phase but received message 33 bytes or more long The message is received up to two bytes and terminated.
	82H	After selection phase, identify message received, ATN signal from initiator detected, and SPC switched to message-in phase, but received message 33 bytes or more long The message is received up to two bytes and terminated.
	93H	After selection phase, identify message and one command received, ATN signal from initiator detected, and SPC switched to message-in phase, but received message 33 bytes or more long The message is received up to two bytes and terminated.
	B2H	After selection phase, one command received, ATN signal from initiator detected, and SPC switched to message-in phase, but received message 33 bytes or more long (Notes 2 and 4) The message is received up to two bytes and terminated.
80H	00H	SPC selected as target by initiator (Note 6)
81H	00H	SPC selected as target by initiator and ATN generated by initiator detected (Note 6)
84H	00H	SPC selected as target by initiator and attempted to receive one message transiting to command phase but the message not received due to data remaining in Receive MCS Buffer (Note 2)
85H	00H	SPC selected as target by initiator, ATN generated by initiator detected, and attempted to receive one message transitting to message-out phase but the message not received due to data remaining in Receive MCS Buffer

(Continued)

CODE	STEP	Interrupt Source and SPC Operation
A0H	81H	SPC selected as target by initiator but ATN signal not asserted in selection phase (Note 3)
A1H	A2H	SPC selected as target by initiator and after selection phase one message other than identify message received
	A3H	SPC selected as target by initiator and after selection phase one message other than identify message received and attention condition generated by initiator detected
A2H	82H	SPC selected as target by initiator and after selection phase illegal identify message received and attention condition generated by initiator detected
	92H	SPC selected as target by initiator and after selection phase illegal identify message received
A3H	83H	SPC selected as target by initiator, after selection phase identify message received, attention condition generated by initiator detected and one message received
	84H	SPC selected as target by initiator, after selection phase identify message received, attention condition generated by initiator detected and one message received, but attention condition generated by initiator further detected
A4H	B1H	SPC selected as target by initiator and after selection phase attempt made to receive CDB, but the CDB undefined (Note 2) Terminates receiving first byte of CDB
A5H	92H	SPC selected as target by initiator and after selection phase identify message received and attempt made to receive CDB, but the CDB undefined Terminates receiving first byte of CDB
A6H	B2H	SPC selected as target by initiator and after selection phase one CDB received (Note 2)
A7H	93H	SPC selected as target by initiator and after selection phase identify message and one CDB received
A8H	B3H	SPC selected as target by initiator and after selection phase one CDB received, ATN signal from initiator detected, and one message received (Notes 2 & 4)
	B4H	SPC selected as target by initiator and after selection phase one CDB received, ATN condition from initiator detected, and one message received, but ATN condition detected again (Notes 2 & 4)
A9H	94H	SPC selected as target by initiator and after selection phase identify message and one CDB received, ATN condition from initiator detected, and one message received
	95H	SPC selected as target by initiator and after selection phase identify message and one CDB received, ATN condition from initiator detected, and one message received, but ATN condition from initiator detected again
AAH	B2H	SPC selected as target by initiator and after selection phase one CDB received and ATN condition from initiator detected (Notes 2 & 5)

Notes:

Note 1. Terminated concurrently with detection

Note 2. Reported only when message or command phase is set in the phase setting after selection in the SEL/RESEL operation mode setting register.

Note 3. Reported when only message phase is set in the phase setting after selection in the SEL/RESEL operation mode setting register.

Note 4. Reported when automatic receive mode is set

Note 5. Reported when automatic receive mode is not set

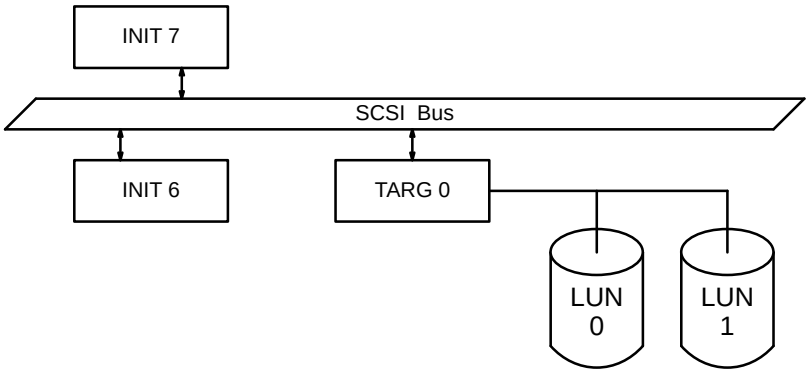
Note 6. Reported when automatic selection response mode is not set in the SEL/RESEL operation mode setting register

APPENDIX C. COMMAND QUEUING

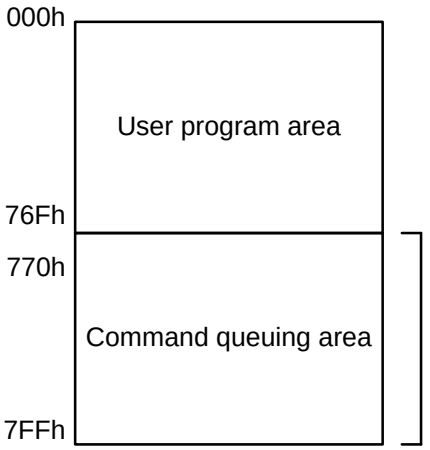
The execution of command queuing requires the following items.

(1) Using user program
(2) Assigning part of user program area to command queuing area
(3) Managing queuing strings by host MPU

Example: System composed of two initiators and two LUNs



To write three command descriptor blocks (CDBs) to one queue, assign part of the user program area to the command queuing area and assign the area for three CDBs to each queue (assigning 12 bytes as one CDB).



	LUN0	LUN1
INIT 7	Addresses 770h to 793h	Addresses 794h to 7B7h
INIT 6	Addresses 7B8h to 7DBh	Addresses 7DCh to 7FF

In order to store the received CDBs, set the **RECEIVE CMD** command as the user program command to addresses assigned for each queue.

• Operation sequence

Set the automatic selection response bit in the SEL/RESEL operation mode register to '1'.

If the initiator performs command queuing through a sequence shown in Figure C-1 below, the **MSG Receive with Identify-MSG** interrupt is reported.

After checking the initiator ID with the nexus status register (addr : 03h), analyze the **identify-MSG** with the host MPU to analyze the LUN for queuing. Define one of four queues.

Then, use the user program **RECEIVE CMD** command to write the CDBs to the command queuing area. At this time, the host MPU specifies the operand for the **RECEIVE CMD** command (the location of the empty area).

The host MPU should manage the starting address of the command queuing area to which Queue-TAG MSG, Queue-TAG Value and CDB are written, as one block, as listed in Table below.

	LUN0	LUN1
INIT 7	Block 0	Block 0
	Block 1	Block 1
	Block 2	Block 2
INIT 6	Block 0	Block 0
	Block 1	Block 1
	Block 2	Block 2

One block

- Queue-TAG MSG
- Queue-TAG Value
- SPC queuing address

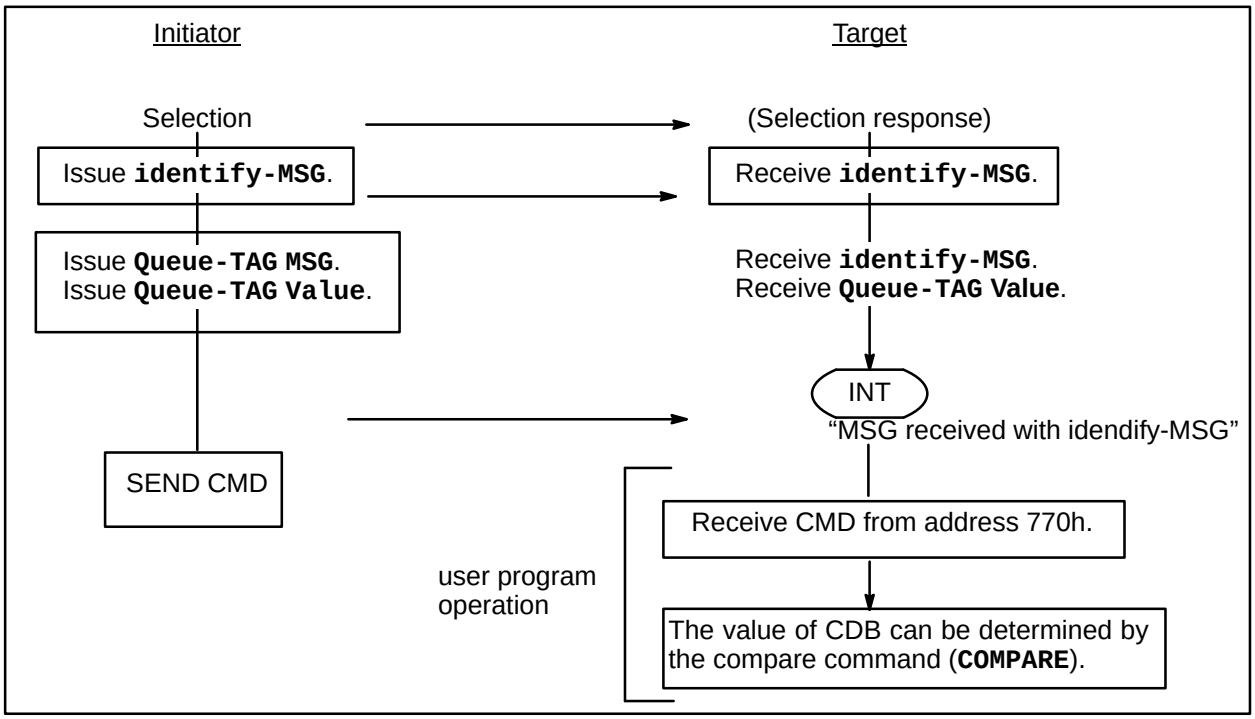


Figure C-1. Sequence for Command Queuing

APPENDIX D. PACKAGE DIMENSIONS

The MB86605 is housed in a 144-pin plastic shrink QFP package. The package dimension is shown in the figure below. (Package code : FPT-144P-M05)