Integrated Circuits Division

FAST TRACK TO SCSI

Product Guide

Fujitsu Microelectronics, Inc.

This document is published by the Publications Department, Fujitsu Microelectronics, Inc.,
3545 North First Street, San Jose, California, 95134–1804; U.S.A.

Application Notes are provided by the Applications Engineering Depatment of Fujitsu Microelectronics, Inc.

Printed in the U.S.A.

Edition 1.0

NOTICE TO THE READER

In response to your request for literature on our SCSI product line, please find enclosed a preliminary copy of our SCSI User Manual. We apologize for not being able to supply you with a final printed version of the document and hope this interim copy is temporarily acceptable. The technical contents are correct and future versions will simply include additions such as background information, Table of Content, and an Index. The final version will be completed by the end of September.

Thank you for your understanding.

Chapter 1

PRODUCT OVERVIEW

The following section is a detailed description of the SCSI Protocol Controller (SPC) registers.

Please note the SPCs individual bit functions within the registers use positive logic i.e., to turn a function on or assert a condition, write a 1 to the respective bit location within the register. To turn a condition off or to deassert, write a 0 to the bit location.

Fujitsu SPCs share a common architecture consisting of 16 directly-addressable registers. All products are 100% software compatible with one another, making migration within the product line a relatively simple process. Accessed via a unique 4-bit address (A3 - A0), the register set is laid out in a logical and straight forward manner to simplify the job of firmware design. Many critical functions such as arbitration, phase sensing handshake, and data transfer count are all handled in hardware, thus reducing software overhead and development time while increasing performance. All products feature an 8-byte first-in first-out (FIFO) to ease timing requirements between the system bus and the SCSI bus.

Fujitsu SPCs can be driven using hardware interrupts or they may operate in a polled mode with interrupts disabled. An implementation requiring a combination of polled and hardware interrupts can also be accommodated.

The three operating modes of the Fujitsu SPCs include a Manual Transfer Mode, where total control of the SCSI bus signals is possible; i.e., direct control of the REQ/ACK handshake. The designer will find this operating condition useful for testing and debugging software and hardware.

The other two operating modes are designed for automated data transfers. The Program Transfer Mode is designed for non-direct memory access (DMA) transfers with complete hardware handshake. This mode is useful for short data transfers. The DMA Mode is used for large block transfers. On-chip DMA interface logic simplifies hardware connection to commercially available DMA controllers.

In addition to the aforementioned transfer modes Fujitsu SPCs also feature a Diagnostic Mode for simulating events on the SCSI bus. This mode can be used to check the software and the SPC hardware interface without actually connecting it to the SCSI bus.

The Fujitsu SPC family consists of general-purpose controllers. The four products can be grouped into two product categories; asynchronous and synchronous (data transfer method). Within each of these groups we offer a version with on-chip single ended drivers and a version optimized for differential drive (to be used with off-chip drivers).

The optimization for differential drive consists of the separation of SCSI Out and SCSI In ports (see Figure 1–1). If these buses were combined, 3-state logic would be required between the driver/receiver and the SPC to support the arbitration feature of SCSI. The following explanation is offered to help clarify this point.

> According to the ANSI X.131-1986 document defining SCSI, when SCSI devices arbitrate for control of the SCSI bus, each must assert their individual ID bit (bits 0 through 7 are available with bit 7 awarded the highest priority). While asserting one bit, the SPC must concurrently read the other seven lines to determine if arbitration was won or lost. Simultaneous assertion of one bit while reading others necessitates the use of 3-state logic.

The complexity and added connections required for differential drivers makes the addition of 3-state logic significant. Fujitsu SPCs eliminate the need by allowing the SPC to read incoming IDs on the SCSI In bus while asserting the ID bit on the SCSI Out bus.

In a single-ended environment the number of connections is relatively low so using 3-state logic does not add a significant amount of logic to the board. Those SPCs that contain on chip single-ended drivers feature a single SCSI I/O port and on-chip 3-state circuitry.

As stated earlier, the Fujitsu SPCs are based on a common architecture so the functional and operational descriptions found in Chapters 3 and 4 respectively apply to all products (except where noted). The following descriptions and tables are provided as a means of highlighting the distinguishing features of each SPC.

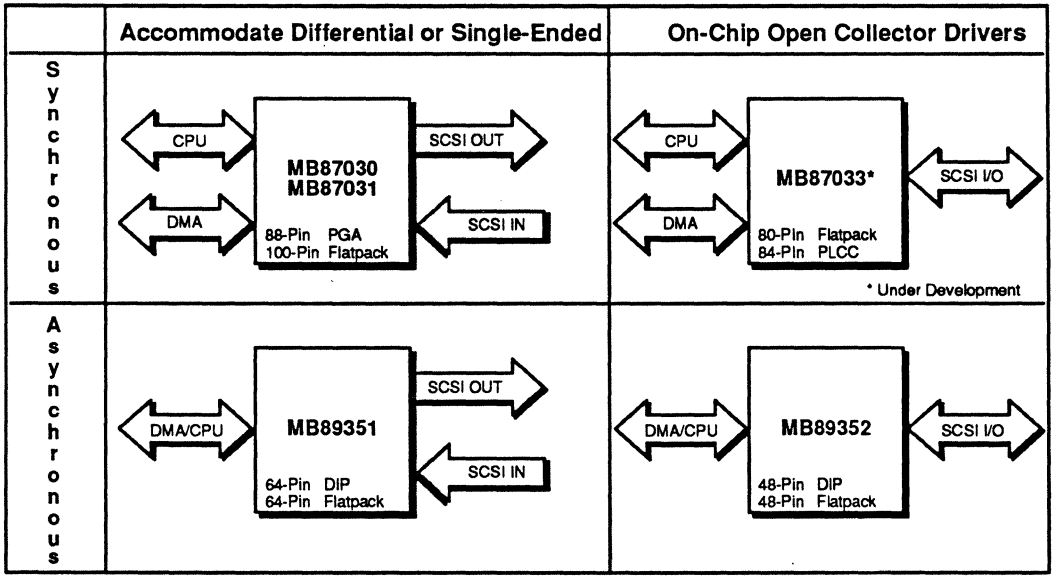


Figure 1–1. Bus Structures in Relation to Chip Features (asynchronous vs. synchronous and on chip driver vs. off-chip drivers)

This matrix illustrates the various bus structures and how they relate to the chip features.

MB87030CR and MB87031PF

These designs are best suited for high performance applications requiring differential drivers and synchronous transfer capability. Separate control and data buses allow uninterrupted access to the SPC. This is particularly critical during synchronous data transfer operations. A combined bus structure would require the transfer to stop while the MPU is accessing the chip.

MB87033B

The MB87033B is also intended for high performance systems requiring synchronous data transfer. However, the MB87033B includes on-chip single ended drivers allowing direct connection to a single-ended SCSI bus. Separate control and data buses permit the central processing unit (CPU) to monitor the SPC during high-speed synchronous-burst data transfers.

MB89351

The MB89351 is based on the MB87030 except the synchronous logic has been removed. Intended for lower cost applications, the MB89351 combines the data and control buses as a means of reducing pin count, external parts count, and cost.

The MB898351 features separate SCSI In and Out buses similar to the MB87030/31. This configuration simplifies the interface to differential drivers by eliminating the need for 3-state logic.

MB89352

This device is functionally the same as the MB89351 except the single-ended drivers are on-chip. Because the device is used only in single-ended applications, the SCSI In and SCSI Out buses are combined.

Table 1–1 highlights the salient features of each device.

Table 1–1. Device Features

	MB87033	MB87030 MB87031	MB89352	MB89351
Single Ended Driver/Rcvr	On Chip	No	On Chip	No
Differential Driver/Rcvr	No	External	No	External
Synchronous Transfer	Yes	Yes	No	No
Transfer Byte Counter	28-bit	24-bit	24-bit	24-bit
Arbitration Fail Interrupt	Yes	No	No	No
Atn Condition Detect Interrupt	Yes	No	No	No
FIFO Full/Empty Interrupt	No	No	Yes	Yes
Interrupt Signal Line Count	2	1	1	1
MPU Bus Parity Generator	Yes	No	Yes	Yes
DMA Bus	Separate From CPU BUS	Separate From CPU BUS	Common To CPU BUS	Common To CPU BUS
Process	CMOS	CMOS	CMOS	CMOS
Package	QFP-80P LCC-84P	PGA-88C QFP-100P	DIP-48P QFP-48P	SDIP-64P QFP-64P

Chapter 2

ARCHITECTURE

Addressing

A unique address is assigned to each internal register, and a particular register is identified by address bits A0 to A3. Table 2–1 shows internal register addressing.

Internal Register Assignments

Table 2–2 shows the bit assignment to each internal register. When accessing an internal register (in read/write), remember the following:

(1) The internal register block includes read-only/write-only registers.

(2) Some registers serve two functions depending on whether the register is being written to or read.

(3) A write to a read-only register is ignored.

(4) If a write-only register is read out, the data and parity bit are undefined.

(5) At bit positions indicating "—" for a write in Tables 2–1 and 2–2 indicate either 1 or 0, and may be written.

Table 2–1. Internal Register Addressing

$\overline{CS}$	A3	A2	A1	A0	OP	Register Name	Abbr.
0	0	0	0	0	R W	Bus Device ID	BDID
0	0	0	0	1	R W	SPC Control	SCTL
0	0	0	1	0	R W	Command	SCMD
0	0	0	1	1	R W	Transfer Mode	TMOD
0	0	1	0	0	R W	Interrupt Sense Reset Interrupt	INTS
0	0	1	0	1	R W	Phase Sense SPC Diagnostic Control	PSNS SDGC
0	0	1	1	0	R W	SPC Status —	SSTS
0	0	1	1	1	R W	SPC Error Status —	SERR
0	1	0	0	0	R W	Phase Control	PCTL
0	1	0	0	1	R W	Modified Byte Counter —	MBC
0	1	0	1	0	R W	Data Register	DREG
0	1	0	1	1	R W	Temporary Register	TEMP
0	1	1	0	0	R W	Transfer Counter High	TCH
0	1	1	0	1	R W	Transfer Counter Middle	TCM
0	1	1	1	0	R W	Transfer Counter Low	TCL
0	1	1	1	1	R W	External Buffer	EXBF

Table 2–2. Internal Register Bit Assignment

HEX Address	Name (Abbr.)	OP	7	6	5	4	3	2	1	0	P
0	Bus Device ID (BDID)	R	SCSI Bus Device ID								
			#7	#6	#5	#4	#3	#2	#1	#0	'0'
		W						SCSI Bus Device ID			
								ID4	ID2	ID1	
1	SPC Control (SCTL)	R	Reset and Disable	Control Reset	Diag Mode	ARBIT Enable	Parity Enable	Select Enable	Reselect Enable	INT En-abel	P
		W									
2	Command (SCMD)	R	Command Code			RST Out	Inter-cept xfer	Transfer Modifier			P
		W						PRG Xfer	'0'	Term. Mode	
3	Transfer Mode (TMOD)	R	Sync. Xfer	Max. Transfer Offset			Min. Transfer Period		'0'	'0'	P
		W		4	2	1	2	1			
4	Interrupt Sense (INTS)	R	Selected	Resel-ected	Discon-nected	Com mand Com plete	Service Required	Time Out	SPC Hard Error	Reset Con dition Reset	P
		W	Reset Interrupt								—
5	Phase Sense (PSNS)	R	REQ	ACK	ATN	SEL	BSY	MSG	C/D	I/O	P
		W	—								—
5	SPC Diagnostic Control (SDGC)	R	—								—
		W	Diag. REQ	Diag. ACK			Diag. BSY	Diag. MSG	Diag. C/D	Diag. I/O	
6	SPC Status (SSTS)	R	Connected		SPC Busy	Xfer in Pro-gress	SCSI RST	TC=0	DREG Status		P
			INIT	TARG					FULL	EMPTY	
		W	—								—
7	SPC Error Status (SERR)	R	Data Error		'0'	'0'	TC P-Error	Phase Error	Short Period	Offset Error	P
			SCS	SPC							
		W									
8	Phase Control (PCTL)	R	Bus Free Omter Write Enable	'0'				Transfer Phase			P
		W						MSG Out	C/D Out	I/D Out	
9	Modified Byte Counter (MBC)	R	'0'					MBC			P
							Bit 3	2	1	0	
		W	—								

Table 2–2. Internal Register Bit Assignment (Continued)

HEX Address	Name (Abbr.)	OP	7	6	5	4	3	2	1	0	P
A	Data Register (DREG)	R	Internal Data Register (8 Byte FIFO)								P
		W	Bit 7	6	5	4	3	2	1	0	
B	Temporary Register (TEMP)	R	Temporary Data Input From SCSI)								P
			Bit 7	6	5	4	3	2	1	0	
		W	Temporary Data (Output From SCSI)								P
			Bit 7	6	5	4	3	2	1	0	
C	Temporary Counter High (TCH)	R	Transfer Counter High (MSB)								P
		W	Bit 23	22	21	20	19	18	17	16	
D	Temporary Counter Mid (TCM)	R	Transfer Counter Middle (2nd Byte)								P
		W	Bit 15	14	13	12	11	10	17	8	
E	Temporary Counter Low (TCL)	R	Transfer Counter Low (LSB)								P
		W	Bit 7	6	5	4	3	2	1	0	
E	Temporary Counter Low (TCL)	R	External Buffer								P
		W	Bit 7	6	5	4	3	2	1	0	

Functional Blocks

Figure 2–1 shows the SPC functional block diagram. SPC mainly consists of the following functional blocks:

(1) Internal register block

(2) MPU interface control block

(3) Bus phase control block

(4) Arbitration/selection sequence control block

(5) Transfer sequence control block

(6) Transfer byte counter block

(7) Data buffer register block

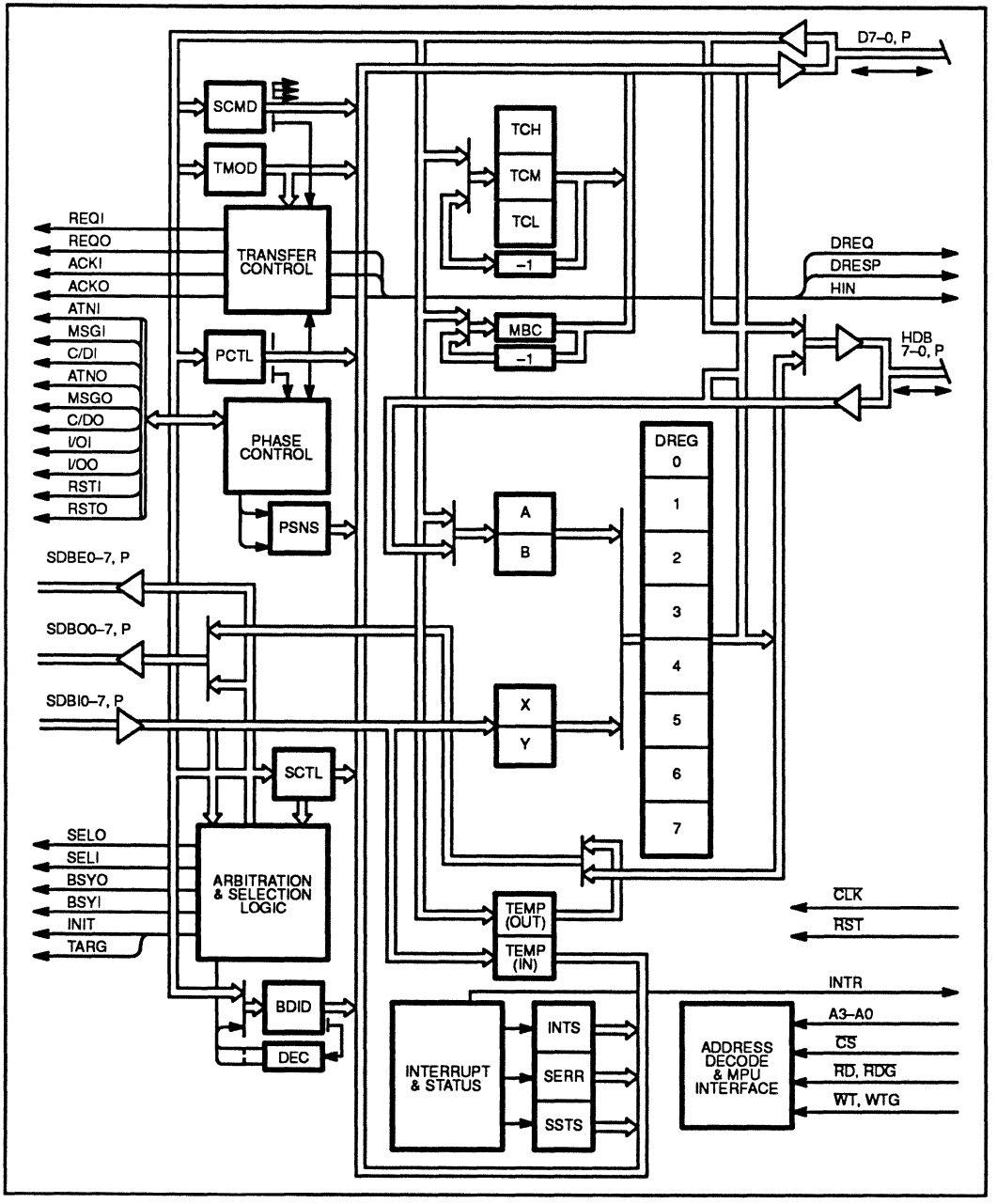


Figure 2–1. SPC Functional Block Diagram

Operation or Block Diagram Description

MPU Interface Control Block

The MPU Interface Control Block selects a specified internal register and controls a read/write operation.

(This is done to report the result status of the SPC internal operation, and the detection of an error, if encountered.)

Bus Phase Control Block

The bus phase control block generates a specified bus phase for SCSI and controls its sequence of execution. Also, this block supervises SCSI status, and responds to the bus phase being executed if necessary.

Arbitration/Selection Sequence Control Block

The sequence control block executes the ARBITRATION phase with the SCSI-specified timing and obtains permission to use the SCSI bus. Then, it carries out the SELECTION/RESELECTION phase and checks the response from the selected/reselected device. Also, this block detects the SELECTION/RESELECTION phase on SCSI, and checks the bus device ID specified in the internal register against that specified on the SCSI data bus. Then, if this block finds that SPC has been selected by another SCSI bus device, it executes the response sequence for the SELECTION/RESELECTION phase.

Transfer Sequence Control Block MB87030/1/3

This sequence control block controls the DATA IN/OUT, COMMAND, STATUS, and MESSAGE IN/OUT phases to be executed in SCSI. The following two modes are available for execution of these transfer phases:

(1) Manual Transfer Mode

In Manual Transfer Mode, the MPU interface is used for transferring data and sending/receiving/checking the REQ/ACK signals on SCSI.

(2) Hardware Transfer Mode

In Hardware Transfer Mode, SPC controls the SCSI transfer sequence according to the transfer mode and transfer byte count specified in the internal registers, and reports the end result of transfer. In the asynchronous mode transfer operation, the REQ and ACK signals are controlled by the interlock protocol. In the synchronous mode transfer operation, a maximum of 8-byte offset is available for the DATA TRANSFER phases. The hardware transfer mode is subdivided into two modes according to the data routing:

a) Program Transfer Mode

Data is routed via MPU interface in the following data paths:
SCSI ↔ Data buffer register ↔ Data bus lines D7 to D0 and DP.

b) DMA Mode

Data is routed using DREQ and DRESP signals for DMA control in the following data paths:
SCSI ↔ Data buffer register ↔ Data bus lines HDB7 to HDB0 and HDBP.

Transfer Sequence Control Block MB89351 and MB89352

This sequence control block controls the DATA IN/OUT, COMMAND, STATUS, and MESSAGE IN/OUT phases to be executed in SCSI. The following two modes are available for execution of these transfer phases:

(1) Manual Transfer Mode

In Manual Transfer Mode, the MPU interface is used for transferring data and sending/receiving/checking the REQ/ACK signals on SCSI.

(2) Hardware Transfer Mode

In Hardware Transfer Mode, SPC controls the SCSI transfer sequence according to the transfer mode and transfer byte count specified in the internal registers, and reports the end result of transfer. In the asynchronous mode transfer operation, the REQ and ACK signals are controlled by the interlock protocol. The hardware transfer mode is subdivided into two parts according to the following data routing:

a) Program Transfer

Data transfer between the MPU and SCSI is executed using DREG. The DREG status is acknowledged by the MPU with one of the following methods.

— The MPU reads the SSTS register at any time and controls data transfer according to the results.

— INTR (interrupt signal) is sent to the MPU to acknowledge the DREG needs data or has data as a result of reading from or writing in SCSI.

b) DMA Transfer

In this mode, the SPC executes data transfer with the MPU via DMAC by sending the DREQ signal.

When requesting memory access, the SPC makes the DREQ signal active.

During an input operation, DREQ is sent when the SPC internal data buffer register contains data received from SCSI. During an output operation, DREQ is sent when all bytes specified by the transfer byte counter have not yet been fetched and the internal data buffer contains empty bytes.

DREQ is not a 1-byte transfer request. DREQ is kept active while the SPC is in the conditions explained above.

Also, the $\overline{\text{DACK}}$ signal may be kept active if the response can be continuously issued. It does not have to be pulsed for each byte.

Transfer Byte Counter Block

The transfer byte counter indicates the number of bytes to transfer to/from SCSI for hardware transfer mode operation. It is 24-bits long. MB87033B transfer byte counter is 28-bits. Except for execution of a special transfer operation (padding), the transfer byte counter is decreased by one each time one byte of data is transferred on SCSI. The transfer byte counter is also used as a timer for supervising the waiting time for a response to be returned from the selected SCSI bus device during execution of the selection/reselection phase. The three 8-bit registers, TCH, TCM, and TCL, make up the 24-bit counter.

Data Buffer Register Block

The data buffer register block is used in execution of a Hardware Transfer Mode operation in SPC. It has a capacity of eight bytes, and operates on the FIFO principle for each byte. In an input operation (from SCSI to SPC), data received from SCSI is loaded into the buffer register. In DMA mode, transfer request signal DREQ is generated to the external buffer memory. Also, in program transfer mode data can be read out from this buffer register. In an output operation (from SPC to SCSI), data supplied from the MPU interface (program transfer mode) or external buffer memory (DMA mode) is loaded into the buffer register and then sent to SCSI. In this case, a maximum of eight bytes of data is prefetched into the SPC. The eight byte FIFO appears as a single register (DREG) to the MPU interface. In an input operation, the byte locations holding valid data are selected in succession for reading out. In an output operation, the empty byte locations are selected in succession for writing in.

Chapter 3

OPERATIONAL DESCRIPTION

BDID

Address
#0

BDID Register – Bus Device Identifier Register

OP	7	6	5	4	3	2	1	0	P
R	Bit Significant Bus Device ID								
	#7	#6	#5	#4	#3	#2	#1	#0	'0'
W	—					SCSI Bus Device ID			—
						ID4	ID2	ID1	

Write Operation (Bits 2 to 0):

The BDID register specifies the SCSI bus device identifier of the SPC (ID) as a three-bit binary number. This setting must be completed before resetting the SCTL register's bit 7 (Reset & Disable). Bits 2 to 0 are cleared during a power-on reset only.

Read Operation:

The BDID register indicates the SCSI physical device ID of the SPC. The SCSI physical device address is decoded from the values of bits 2 to 0 as specified in a write operation to the BDID register described above. One of the bits is 'one', the others are all 'zeros'. The SCSI bus usage priority is assigned in descending order from bit 7 to bit 0. Using a bus device ID indicated in this register, the SPC executes the ARBITRATION phase. Also, if a SELECTION/RESELECTION phase condition occurs in SCSI, the value of the SCSI data bus is checked against the contents of this register to see whether the SPC has been selected/reselected or not.

Examples: Writing hex 03 to this register will result in hex 08 being read.
Writing hex 07 to this register will result in hex 80 being read.

SCTL
Address #1

SCTL Register — SPC Control Register

OP	7	6	5	4	3	2	1	0	P
R	Reset and Disable	Control Reset	Diag Mode	ARBIT Enable	Parity Enable	Select Enable	Reselect Enable	INT Enable	P
W									

Bit 7: Reset and Disable

Bit 7 generates a reset instruction to the internal registers and control circuits in the SPC. When this bit is 1, the SPC is reset and logically disconnected from SCSI (put in the disable state). Execution of the hardware reset ($\overline{RST}$ input = 'L') causes this bit to be 1. To enable the SCSI operation, bit 7 must be made zero to enable SPC operation.

Bit 6: Control Reset

The data transfer control circuit is reset when this bit is 1. Even if the control reset is executed by this bit while the SPC is connected to the SCSI bus, the SPC maintains the connection with SCSI. This bit should be used for resetting the data transfer control circuit while it is connected with SCSI. More specifically, it should be used when an error is indicated by bit 1 (SPC Hardware Error) of the INTS register during execution of the SCSI transfer phase or when a timeout occurs before completion of Transfer command execution (supervised by the MPU program). Bit 1 (SPC Hardware Error) of the INTS register and the SERR register are cleared by this bit.

Note: This reset function initializes the SPC transfer control circuits. So, when the SPC serves as an INITIATOR, the following consideration is required:

— Since the REQ signal may be active during control reset, the first byte should be transferred in manual mode after resetting.

Bit 5: Diagnostic Mode

When bit 5 is 1, the SPC enters a diagnostic mode and is disconnected from SCSI. The diagnostic mode allows pseudo-execution of the SCSI operation using the SDGC register (to be explained later).

Bit 4: Arbitration Enable

Bit 4 indicates whether the ARBITRATION phase is executable in SCSI or not.

1 – ARBITRATION phase executable

0 – ARBITRATION phase nonexecutable

With this bit set to 1, the Select command causes the SPC to execute the ARBITRATION phase. If the SPC wins the arbitration, it executes the SELECTION/RESELECTION phase. With this bit set to 0, the Select command causes the SPC to execute the SELECTION phase without the ARBITRATION phase. As long as bit 4 is 0, the SPC will not respond to a reselection request from other SCSI bus devices. No response is made even if bit 1 (Reselect Enable) of this register is set to 1. Remember that bit 4 must be set correctly before clearing bit 7 (Reset and Disable) of the SCTL register. Note also that this bit should not be changed in other than a diagnostic mode after clearing bit 7 of the SCTL register.

Bit 3: Parity Enable

Bit 3 indicates whether the parity of data received from the SCSI data bus is to be checked or not.

1 – Parity of the data received from the SCSI data bus is checked.

0 – Parity of the data received from the SCSI data bus is not checked.

Regardless of the value of this bit, the parity of the data to be sent to the SCSI data bus is ensured. Also, the parity of the data on the SPC internal data bus is always checked.

Note: While the SPC is connected with SCSI, this bit should not be changed.

A parity check is carried out in the following cases:

(1) When checking an ID value placed on the data bus upon detection of the SELECTION/RESELECTION phase in SCSI detection of a parity error causes no response to the SELECTION/RESELECTION phase even if the SCSI bus device ID has been matched.

(2) If a parity error is detected in a data byte, when receiving data from SCSI in an input transfer sequence, the relevant parity bit value is corrected and the data byte with the corrected parity is sent to the MPU/DMA data bus.

Bit 2: Select Enable

1 – The SPC responds as a TARGET device to the SELECTION phase in SCSI.

0 – The SPC does not respond to the SELECTION phase in SCSI.

Note: If the SPC has already detected the SELECTION phase during an attempt to set this bit to 0, the SPC responds to the SELECTION phase as a TARGET device. In this case, the 0 setting is effective for the subsequent SELECTION phase (with no response).

Bit 1: Reselect Enable

1 – SPC responds as an INITIATOR to the RESELECTION phase in SCSI.

0 – SPC does not respond to the RESELECTION phase in SCSI.

Note: If SPC has already detected the RESELECTION phase during an attempt to set this bit to 0, SPC responds to the RESELECTION phase as an INITIATOR. In

this case, the 0 setting is effective for the subsequent RESELECTION phase (with no response).

Bit 0: Interrupt Enable

Bit 0 serves as a mask for enabling/disabling the hardware interrupt (INTR) output from the SPC.

1 – Interrupt enabled

0 – Interrupt disabled

A hardware interrupt due to a RESET condition detected in SCSI cannot be masked.

Regardless of the value of this bit, an interrupt event is always indicated in the INTS register. This allows poll-mode operation when the hardware interrupt (INTR) is disabled.

SCMD
Address
#2

SCMD Register — SPC Command Register

OP	7	6	5	4	3	2	1	0	P
R W	Command Code			RST Out	Inter-cept Xfer	Transfer Modifier PRG Xfer	'0'	Term. Mode	P

Register Functions

The SCMD register is used for issuing a command to the SPC. Writing into this register causes the SPC to initiate the command specified with bits 7 to 5.

Bit 7 to 5: Command Code

Bit 7	6	5	Command
0	0	0	Bus Release
0	0	1	Select
0	1	0	Reset ATN
0	1	1	Set ATN
1	0	0	Transfer
1	0	1	Transfer Pause
1	1	0	Reset ACK/REQ
1	1	1	Set ACK/REQ

Bit 4: RST Out

If bit 7 (Reset and Disable) of the SCTL register is 0, setting bit 4 to 1 asserts the SCSI RST signal. When bit 4 is set to 1, a command being executed or waiting for execution in the SPC is cleared and all signals to SCSI other than RST are deactivated. To ensure for the SCSI timing requirements, the MPU must maintain this bit at 1 for more than 25 microseconds.

Note: If the RST signal is received from the SCSI bus with this bit set to 0, the operation sequence is as follows:

(1) A command being executed or waiting for execution in the SPC is cleared

(2) All signals to the SCSI bus are deactivated

(3) An interrupt condition (non-maskable) is generated

Whenever bit 7 (Reset and Disable) of the SCTL register is 0, the SPC always accepts the RST signal from the SCSI bus.

Bit 3: Intercept Transfer

Bit 3 specifies the special data transfer mode. It is valid only when SPC serves as an INITIATOR.

This bit should be set to 1 together with the Set ATN. Set ACK/REQ, or Bus Release command (Bus Release command has no effect when the SPC has been connected with SCSI as an INITIATOR). This bit should be reset together with the Reset ACK/REQ command. (When two or more bytes are transferred using the Set ACK/REQ and Reset ACK/REQ commands, this bit must be reset on issuance of the Reset ACK/REQ command for the

last byte.) With bit 3 of SCMD register set to 1, executing manual transfer (MPU-controlled transfer using the Set ACK/REQ and Reset ACK/REQ commands) does not change the contents of the eight-byte data buffer register in the SPC. Therefore, if a TARGET changes bus phase (i.e., it changes to MESSAGE IN during execution of the DATA OUT phase), this intercept transfer mode makes it possible to optionally restart the DATA OUT phase at the end of the interrupting phase. The phase change during transfer execution is reported by a 'service required' interrupt. To execute this intercept transfer mode, bit 3 of SCMD register must be set to 1 prior to the resetting of an interrupt (an interrupt must be reset after bit is set to 1). Even when not using the intercept transfer mode, bit 3 may be specified for resetting a 'service required' interrupt. In this case, bit 3 must be set/reset together with the Bus Release command. For more details, see the description of bit 3 (Service Required) of the INTS register.

Bits 2 to 0: Transfer Modifier

Bits 2 to 0 are used as a field for specifying the execution mode of the information transfer phase. A value must be set in this field when the Transfer command is issued. If any of the following commands are issued during execution of the Transfer command, this field's value must not be changed.

- Set ATN
- Transfer Pause
- Reset ACK/REQ

Bit 2: Program Transfer

1 – Data are transferred between the MPU and the data buffer register in SPC.

0 – Data are transferred in the DMA mode in which the SPC signals a transfer request to the external buffer memory.

Bit 1: Unused

Bit 1 must always be set to 0.

Bit 0: Termination Mode

Bit 0 provides different functions depending on the SPC operating mode. When SPC serves as an INITIATOR, bit 0 specifies the following operations:

1 – Even after the transfer byte counter reaches 0 during execution of the Transfer command, data transfer will continue if the REQ signal arrives from a TARGET within the same phase. If an output operation is in progress, all 0 bits (with a parity bit set to 1) are transmitted as data. During an input operation, the received data is ignored. But parity is checked if it is enabled (Parity Enable). The above data transfer is referred to as padding transfer, which is effective only when the DATA IN or DATA OUT phase is executed. Padding transfer is executed only within SPC, and a transfer request is not signaled to the external buffer memory even if the DMA transfer mode is specified. Padding transfer is maintained until a TARGET changes the bus phase. In the padding transfer mode, if the Transfer command is issued with the initial value of the transfer byte counter set to 0, execution of padding transfer is started with the first byte. To carry out an output operation in this case, the TEMP register must be set to X'00' prior to issuance of the Transfer command.

0 – Transfer command execution terminates when the transfer byte counter reaches 0. The Transfer command must be reissued to receive the next REQ signal from a TARGET.

When the SPC serves as a TARGET, bit 0 specifies the following operations:

1 – If a parity error is detected in the received data during execution of the Transfer command for input, the current transfer sequence is immediately stopped to terminate Transfer command execution.

0 – Even if a parity error is detected in the received data during execution of the Transfer command for input, the current transfer sequence continues until the transfer byte counter reaches 0.

Command Functions

Bus Release command

When the SPC acts as a TARGET, the Bus Release command instructs a transition to the BUS FREE phase. During execution of the information transfer phase, the Transfer Pause command must be issued to halt the data transfer operation prior to this command. Otherwise, the SCSI bus sequence is not ensured. The Bus Release command may also be used to cancel the Select command waiting for the bus to

become free. Note that the Bus Release command is ignored if the SPC has already started the ARBITRATION or SELECTION phase.

Select command

The Select command requests the SELECTION/RESELECTION phase to be started. It shall be issued only when the SPC is not connected with SCSI. When the SPC receives this command, it carries out the following operation upon detection of the BUS FREE phase is SCSI.

(1) When bit 4 (Arbitration Enable) of the SCTL register is set to 1:

After the BUS FREE phase has been detected, the SPC executes the ARBITRATION phase to try to obtain bus usage permission. If the SPC has lost the arbitration, the Select command terminates its execution. If the SPC has won the arbitration, the SPC executes the SELECTION or RESELECTION phase. The SELECTION phase is executed when bit 0 (I/O Out) of the PCTL register is set to 0, and the RESELECTION phase is executed when it is set to 1.

(2) When bit 4 (Arbitration Enable) of the SCTL register is set to 0:

After the BUS FREE phase has been detected, the SPC executes the SELECTION phase.

Before the select command is issued, the following settings must be made in either of the above cases:

(1) PCTL register

Specify the phase to be executed at bit 0 (I/O Out).

0 – SELECTION phase to be executed
1 — RESELECTION phase to be executed

Note that whenever bit 4 (Arbitration Enable) of the SCTL register is set to 0, the SELECTION phase is executed regardless of the value of bit 0 in the PCTL register.

(2) Set ATN command

Issue the Set ATN command if it is required to assert an ATN signal at the SELECTION phase.

(3) TEMP register

In the TEMP register, specify a value to be sent to the SCSI data bus during execution of the SELECTION/RESELECTION phase.

(4) TCH and TCM register

Specify a response (BSY signal) waiting supervisory time for execution of the SELECTION/RESELECTION phase. The supervisory time T_{SL} should be calculated as follows:

Assuming that the value of TCH and TCM is N (MSB: TCH; LSB: TCM): When N does not equal 0, $T_{SL} = (N \times 256 + 15) \times T T_{CLF} \times 2$. When N equals 0, T = infinite. Where T_{CLF} is a cycle time of the clock signal supplied to the $\overline{CLK}$ pin of the SPC.

(5) TCL register

Specify a period of time (T_{WAIT}) from the moment when both BSY and SEL signals become inactive on SCSI (upon detection of the BUS FREE phase) to the moment when the SPC initiates the ARBITRATION/SELECTION phase. Parameters (X'00') to (X'OF') can be specified in the TCL register.

The average value can be derived using the following equation:

$$TWAIT = [(T_{CL}) + 6] \times T_{CLF} \text{ to } [(T_{CL}) + 7] \times T_{CLF}$$

Where,

(T_{CL}) equals the value in the TCL register.
T_{CLF}: equals the cycle time of the clock signal supplied to the $\overline{CLK}$ pin of the SPC.

Table 3–1 lists the recommended values for the TCL register.

Table 3–1. TCL Register Recommended Values for Use of Select Command

T_{CLF} (ns)	T_{CL}	T_{WAIT} (average) (ns)
124 – 180	$(04)_{16}$	1,250 – 1,980
140 – 200	$(03)_{16}$	1,260 – 2,000

In ARBITRATION phase execution by the Select command, the bus device identifier (ID) which is sent to the SCSI data bus is the value specified in the BDID register. The following equation can be used to obtain the period of time (T_{ARB}) required from the moment when the arbitration is started (BSY signal assertion) to the moment when the bus usage priority is examined:

$$T_{ARB} = 32 \times T_{CLF}$$

Where, T_{CLF} indicates a cycle time of the clock signal supplied to the $\overline{CLK}$ pin of the SPC. After the SELECTION/RESELECTION phase execution is started, a time-out interrupt occurs if no response is acknowledged within the supervisory time specified in the TCH and TCM registers. When a time-out interrupt occurs, the SPC holds the current execution state of SELECTION/RESELECTION phase for SCSI. However, until the time-out interrupt condition is reset, the SPC executes "no operation" to the response from the bus device being selected. Either of the following procedures can be used for a time-out interrupt:

- Restart of SELECTION/RESELECTION phase:

  After specifying a new supervisory time in the TCH, TCM and TCL registers, reset the time-out interrupt condition. Then, the SPC will restart the SELECTION/RESELECTION phase in progress. At this time, changing the TEMP register contents can alter the value being sent to the SCSI data bus. New supervisory time T_{SL} is expressed as follows:

  Assuming TCH, TCM and TCL value to be N (MSB: TCH, LSB: TCL);

  $T_{SL} = N \times T_{CLF} \times 2 \ (N \neq 0)$

  Where

  T_{CLF} indicates a cycle time of the clock signal supplied to the $\overline{CLK}$ pin of the SPC.

- Termination of SELECTION/RESELECTION phase:

  When a time-out interrupt occurs, the values of TCH, TCM and TCL registers are 0. Resetting the time-out interrupt condition in this state causes the SPC to deactivate all signals to SCSI and terminate the SELECTION/RESELECTION phase unless the BSY signal is returned. If the BSY signal is returned when the interrupt condition is being reset, then the SPC executes the normal sequence to complete the Select command. To reset a time-out interrupt condition, set bit 2 of the INTS register and TCM registers, and the time-out interrupt will not occur. However, the above time-out interrupt resetting procedure must be carried out to terminate the

SELECTION/RESELECTION phase in progress. If the SPC recognizes a response from the selected/reselected device during the SELECTION/RESELECTION phase execution, the SPC executes an interface sequence to serve as an INITIATOR (at SELECTION phase) or TARGET (at RESELECTION phase). When the Select command is issued, the SPC status is indicated in the SSTS register. Figure 3–1 shows the status transitions.

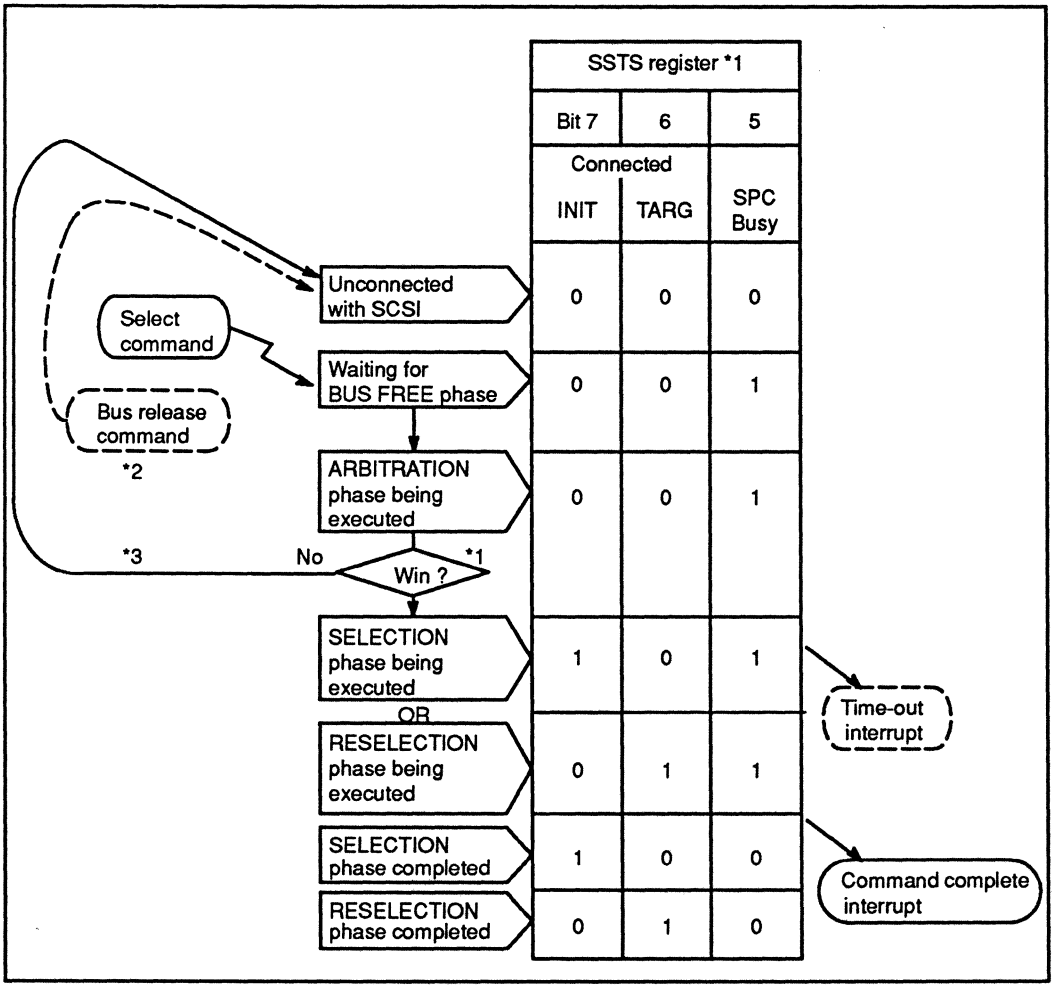


Figure 3–1. Status Transition in Select Command Execution

Notes:

*1: Check the SPC status after the elapsed time (shown below) following issuance of the Select command:
(1) If Arbitration Enable bit is set to 0 the minimum waiting time is 22 x (Clock cycle: T_{CLF})
(2) If Arbitration Enable bit is set to 1 the minimum waiting time is (55 + Set value in TCL) x (Clock cycle: T_{CLF})

*2: The Select command waiting for BUS FREE phase can be canceled using the Bus Release command. However, if the bus becomes free simultaneously with issue of the Bus Release command, the Select command remains valid to execute the ARBITRATION and SELECTION/RESELECTION phases. The MPU program must make sure that the Select command has been canceled after more that four clock cycles pass from the time the Bus Release command (write to SCMD register) was issued.

*3: If the SPC lost the arbitration, the Select command terminates automatically (a command complete interrupt does not occur). In this case, note that the register contents are unpredictable. When issuing the Select command again, be sure to specify the relevant value (see (2)-e, TCL register.

Note: the MB87033 can be configured to issue a command complete interrupt if the SPC loses the arbitration.

Set ATN command

The Set ATN command is valid only when the SPC is acting as an initiator. If this command is issued prior to the Select command, the ATN signal is sent to SCSI during the execution of the SELECTION phase. If the Set ATN command is issued while the SPC is connected with SCSI as an INITIATOR, the ATN signal is sent to SCSI immediately. When the parity checking for the SCSI data bus is enabled and the SPC detects a parity error in the data received from SCSI (during execution of the input transfer operation in hardware transfer mode), the ATN signal is sent automatically to SCSI regardless of the Set ATN command (see Figure 3–2) The assertion of ATN signal is retained until the condition described in the following subsection, Reset ATN command, is satisfied. However, the ATN signaling condition held in SPC, by the Set ATN command issued prior to the Select command, is cleared, if on of the following conditions is met:

- The Select command is canceled by the Bus Release command
- A selected/reselected interrupt occurs before execution of the selection phase

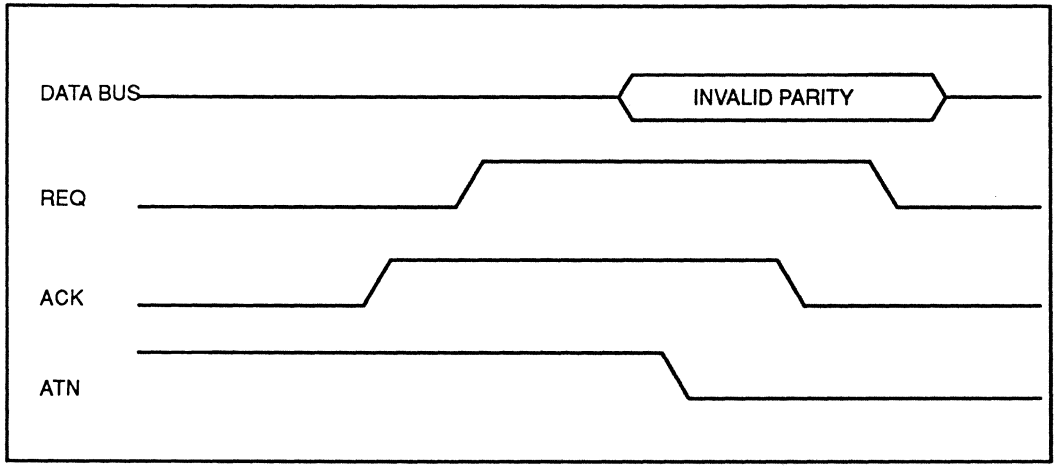


Figure 3–2. ATN Signal Generation in Data Transfer

Reset ATN command

The Reset ATN command is used to reset the ATN signal being sent to SCSI. If the SPC generates an ATN signal (due to a parity error in the received data) during execution of the Hardware Transfer Mode operation, do not issue this command to reset the ATN signal until execution of the current Transfer command is complete. Also, to reset the ATN signal in manual transfer mode, execute the Reset ATN command before the ACK signal is sent to SCSI. In the following cases, the SPC will automatically reset the ATN signal without the Reset ATN command:

- On occurrence of a disconnected interrupt
- On sending the last byte during execution of the MESSAGE OUT phase in hardware transfer mode (see Figure 3–3)

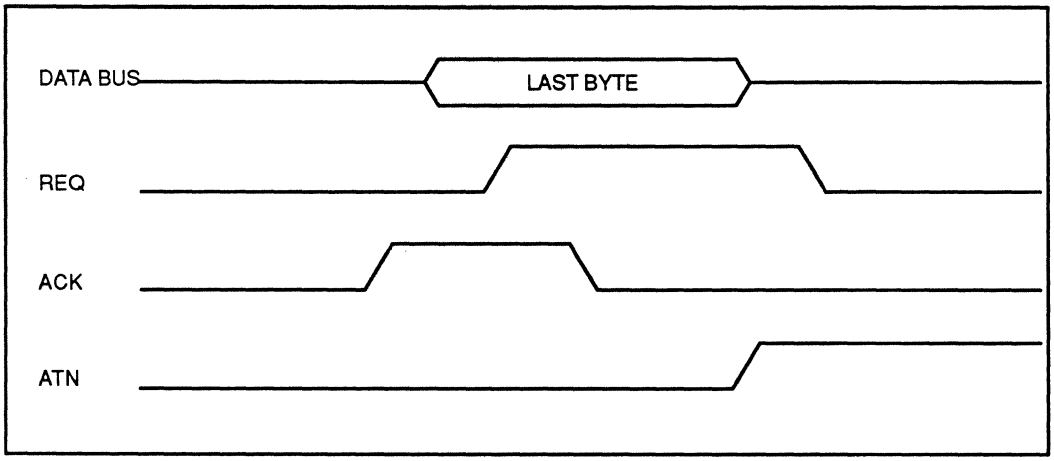


Figure 3–3. ATN Signal Resetting in MESSAGE OUT Phase

Transfer command

The Transfer command executes the following information transfer phases in SCSI:

- DATA IN/OUT phase
- STATUS phase
- COMMAND phase
- MESSAGE IN/OUT phase

The transfer operation initiated by this command is referred to as a hardware transfer operation, the sequence of which is controlled by the SPC. Before issuing the Transfer command, be sure to set up the following:

(1) Transfer byte counter (TCH, TCM, TCL)
Set the byte count of the data to be transferred (MSB: TCH, LSB: TCL)

(2) Bits 2 to 0 of the PCTL register
Set a pattern indicating the phase to be executed

(3) TMOD register
Set detailed transfer mode

When the SPC serves as a TARGET, it executes the information transfer phase specified in the PCTL register and terminates the Transfer command when any of the following conditions is encountered:

- The number of bytes specified in the transfer counter have been transferred.
- Receipt of the Transfer Pause command.
- Detection of a parity error in the received data during an input operation with bit 0 (Termination Mode) of the SCMD register set to 1 (when the parity checking is enabled).

When the SPC serves as an INITIATOR, it starts the transfer operation as follows:

(1) If the REQ signal is received before the Transfer command is issued, the SPC compares the phase requested by the SCSI with that specified in the PCTL register at receipt of this command. Then, if they march, the SPC starts the transfer operation.

(2) If the REQ signal has not been received when the Transfer command is issued, the SPC will wait to execute this command. When the phase specified in the PCTL register matches that requested by SCSI on receipt of the REQ signal, the SPC will start the transfer operation. In the above phase comparison, if a phase mismatch occurs, the Transfer command is nullified, and the SPC generates a service required interrupt. On occurrence of this interrupt, check the PSNS register for the phase requested by the SCSI bus, and issue the Transfer command again or carry out manual transfer.

When the SPC serves as an INITIATOR, Transfer command execution terminates when any of the following conditions is encountered:

- In other than the padding transfer mode, Transfer command execution terminates when the transfer of data of the byte count specified in the transfer byte counter is completed.
- When another information transfer phase is requested by the TARGET.
- In the padding transfer mode, Transfer command execution terminates when another information transfer phase is requested by the TARGET.
- Transfer command execution terminates when disconnected interrupt occurs.

Transfer Pause command

The Transfer Pause command prematurely halts a hardware transfer operation initiated by the Transfer command when the SPC serves as a TARGET. (Note that the Transfer Pause command cannot be used when the SPC serves as an INITIATOR). On receipt of this command, the SPC performs the following:

(1) Stops sending another REQ signal to SCSI, in an input operation.

(2) Stops sending a transfer request (DREQ) signal to the external buffer memory, (in a DMA mode output operation).

Note: For an output operation in program transfer mode, a write to the data buffer register is not allowed after this command has been issued.

Finally, the hardware transfer operation terminates when the internal data buffer register in the SPC becomes empty.

Set ACK/REQ command

The Set ACK/REQ command is used to set ACK or REQ signals for SCSI during execution of manual transfer. When the SPC acts as an INITIATOR, this command causes the ACK signal to be sent. When SPC acts as a TARGET, it causes the REQ signal to be sent. In manual transfer mode,data is transferred via the TEMP register. In this case, the pattern (type) of the information transfer phase to be executed must be preset in bits 2 to 0 of the PCTL register. During execution of manual transfer, the transfer byte counter remains unchanged. Figures 3–4 and 3–5 show the manual transfer procedures.

Reset ACK/REQ command

The Reset ACK/REQ command is used to reset the ACK or REQ signals to the SCSI bus. When the SPC acts as an INITIATOR, this command resets the ACK signal. When the SPC acts as a TARGET, it resets the REQ signal. Use this command for execution of manual transfer. See Figure 3–4 and 3–5 for the manual transfer procedures. Also, reset the ACK signal for the last byte in the MESSAGE IN phase of the hardware transfer mode. In the MESSAGE IN phase, the end of the Transfer command is reported with the ACK signal for the last byte being asserted. The MPU program checks the validity of the received message first, then issues this command. In this case, the ATN signal, if necessary, may be sent out using the Set ATN command prior to this command.

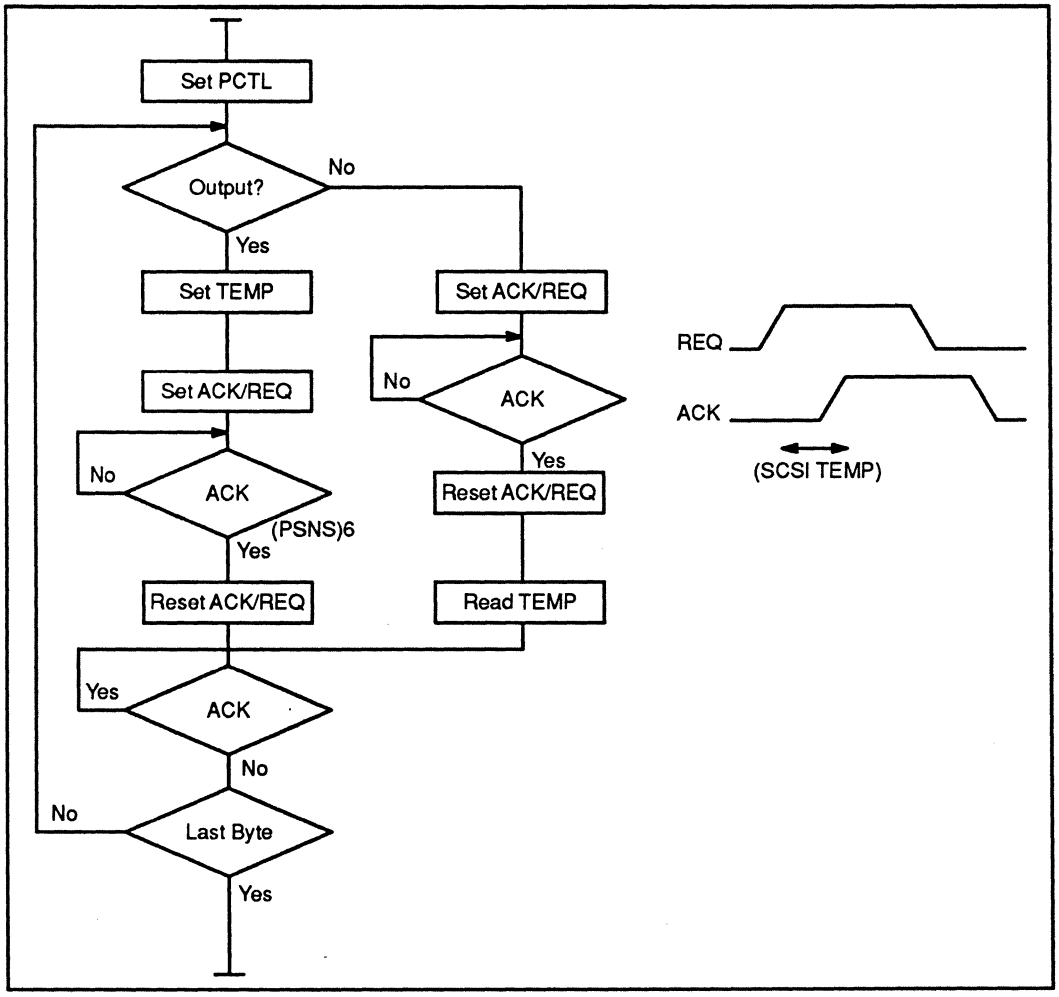


Figure 3–4. Manual Transfer Procedure (SPC serving as a TARGET)

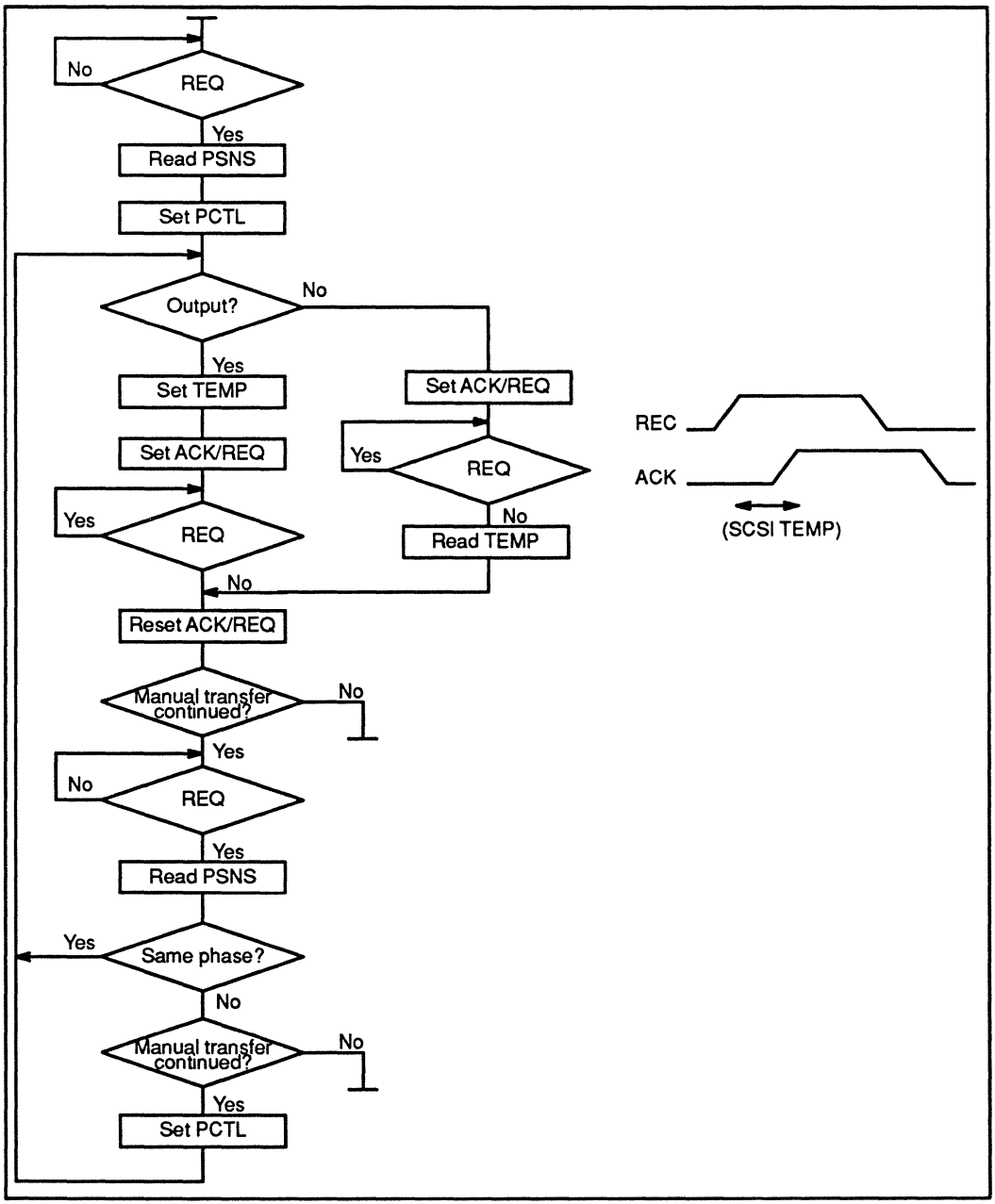


Figure 3–5. Manual Transfer Procedure (SPC serving as an INITIATOR)

Command Termination Report

(1) Immediate commands

The following are immediate type commands which terminate operations immediately after being issued. For these commands, the SPC does not report termination.

- Set ATN
- Reset ATN
- Set ACK/REQ
- Reset ACK/REQ

(2) Interrupt commands

For the following commands, an interrupt occurs at the end of execution. The interrupt cause is indicated in the INTS register.

- Select
- Transfer

(3) Non-interrupt commands

The following commands terminate with different timings depending on the SPC operation being executed. Check the termination status according to the status information in the SSTS register.

Termination status (SSTS register)				
	Bit 7 INIT	6 TARG	5 SPC Busy	4 Xfer in Prg.
Transfer Pause	0	1	0	0
Bus Release*1	0	0	0	0
Note: If a selected/reselected interrupt condition is detected immediately after termination of the Bus Release command, an interrupt occurs and either the INIT or TARG bit is set.				

Command Issuance Timing

Issuance of a command requires a write to the SCMD register. The SPC synchronizes a write to the SCMD register with a clock supplied from the $\overline{CLK}$ pin, and then starts executing the command specified at bits 7 to 5. Figure 3–6 shows the command execution timing. When issuing commands successively, leave an interval between them for more than the sync-loss period (four clock cycles) in the SPC.

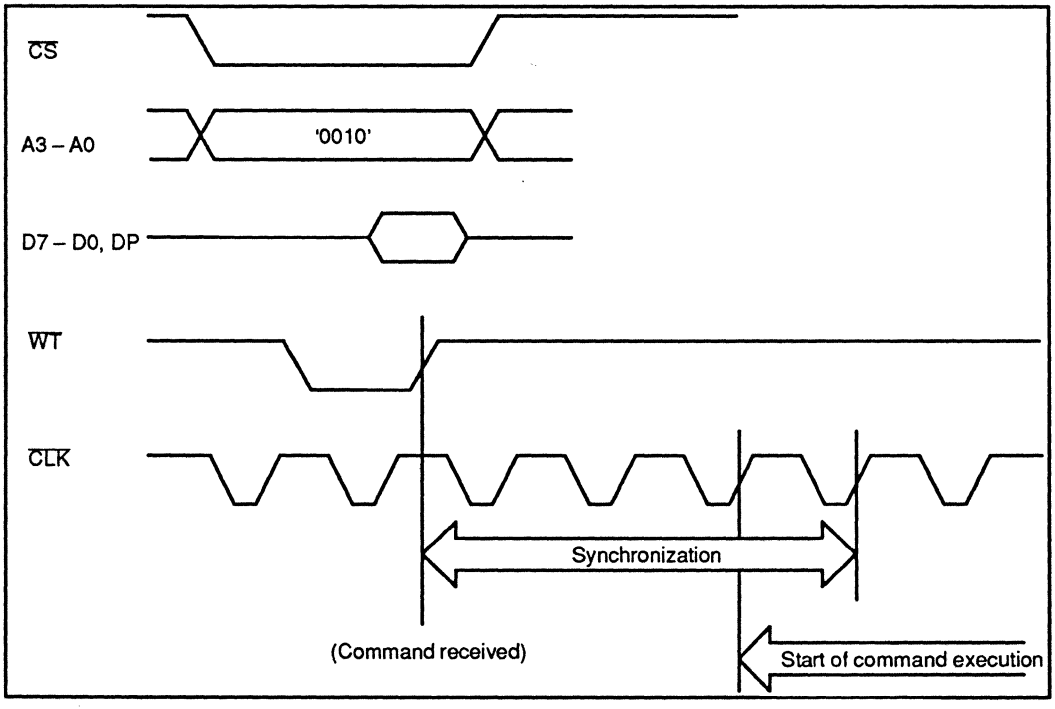


Figure 3–6. Command Execution Timing

TMOD
Address
#3

TMOD Register (MB87030/31/33 only)

TRANSFER MODE REGISTER

OP	7	6	5	4	3	2	1	0	P
R / W	Sync. Xfer	MAX. Transfer Offset			MIN. Transfer Period		'0'	'0'	P
		4	2	1	2	1			

Register Functions

Bit 7: Synchronous Transfer

1 – Indicates that the DATA IN/OUT phase is executed in synchronous transfer mode. The COMMAND, STATUS, and MESSAGE IN/OUT phases are executed in asynchronous transfer mode regardless of this bit value.

0 – Indicates that the DATA IN/OUT phase is executed in asynchronous transfer mode.

Bit 6 to 4: Maximum Transfer Offset

Bits 6 to 4 indicate a maximum value of the REQ/ACK offset to be used in synchronous transfer mode.

Bit 6	Bit 7	Bit 4	Maximum offset value
0	0	0	8
0 – 1	0 – 1	1 – 1	1 – 7

When the SPC serves as a TARGET, it sends the REQ signal in advance within the specified maximum offset value. When the SPC serves as an INITIATOR, it can receive the REQ signal and input data within the specified maximum offset value. If the maximum offset value is exceeded in reception of the REQ signal, an error condition is detected at bit 0 (Transfer Offset Error) of the SERR register.

Bit 3 and 2: Transfer Period

Bits 3 and 2 indicate a parameter for determining the minimum repeat cycle of the REQ and ACK signals in synchronous transfer mode. Specify a period between the trailing edge of the REQ (ACK) signal and the leading edge of the next REQ (ACK) signal in multiples of a clock signal cycle (T_{CLF}) supplied at the $\overline{CLK}$ pin of the SPC. Figure 3–7 shows an example of a transfer period setting. In synchronous transfer mode, the REQ (ACK) signal pulse width (Typical) equals a cycle of the clock signal (T_{CLF}) supplied to the $\overline{CLK}$ pin.

Transfer Mode Setting Timing

The TMOD register value determines the way the DATA IN/OUT phase is executed. When SPC serves as a TARGET, specify the DATA IN/OUT phase (C/D, MSG = 00) in the PCTL register. And, before issuing the Transfer command, be sure to complete the TMOD register setting. When the SPC acts as an INITIATOR, be sure to specify a correct value before the target initiates the DATA IN/OUT phase. A TARGET's phase changing

time and the period from it, until transmission of the first REQ signal, are unpredictable; therefore, set the transfer mode with the following timing:

(1) After completion of SELECTION phase

(a) Before issuing the Transfer command for execution of the COMMAND phase

(b) Before resetting the ACK signal for the last byte in the MESSAGE IN phase (before issuing the Reset ACK command) if a SYNCHRONOUS DATA TRANSFER REQUEST message is transmitted at the end of the command phase.

(2) After reselected interrupt

Before issuing the Reset ACK command in the MESSAGE IN phase (IDENTIFY MESSAGE).

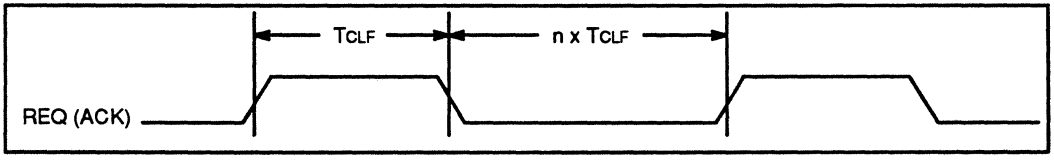


Figure 3–7. Transfer Period Setting

INTS
Address
#4

INTS Register – Interrupt Status Register

OP	7	6	5	4	3	2	1	0	P
R	Sele-cted	Resele-cted	Dis-Connected	Command Complete	Service Required	Time Out	SPC Hardware Error	Reset Condi-tion	P
W	Reset Interrupt								—

Register Functions

The INTS register is used to indicate the cause of an interrupt and reset it. An interrupt issued by the SPC (INTR pin) can be masked by bit 0 (INT Enable) of the SCTL register (except an interrupt whose cause is indicated at bit 0 [RESET condition]). To clear an interrupt, set 1 at the corresponding bit position in this register. Note that only the interrupt condition specified by 1 is reset. The bit positions having 0s remain unchanged (corresponding interrupt conditions are maintained). In this register, two or more interrupts may be reset at the same time.

Interrupt Processing

(1) Selected interrupt (Bit 7)

Bit 7 indicates that the SELECTION phase in SCSI has resulted in the SPC being selected from another bus device (INITIATOR). When the SELECTION phase is detected, the SPC checks the contents of the SCSI data bus. If the following conditions are satisfied, the SPC executes a response sequence on SCSI, then generates a selected interrupt.

a) The ID specified in the BDID register is selected.

b) Not more than two bits are set on the SCSI data bus (excluding the parity bit).

c) When parity checking is enabled, the parity bit value is correct.

In the SELECTION phase, the TEMP register holds the value of SCSI data bus. The SPC serves as a TARGET fromthe occurance of this interrupt until the Bus Release command is issued or the RESET condition is detected in SCSI. During this period the SPC asserts the BSY signal to SCSI. Before issuing the Bus Release command, be sure to reset the cause of this interrupt.

(2) Reselected interrupt (Bit 6)

Bit 6 indicates that the RESELECTION phase in SCSI has resulted in the SPC being reselected from another bus device (TARGET). When the RESELECTION phase is detected, the SPC checks the contents of the SCSI data bus. If the following conditions are satisfied, the SPC executes a response sequence on SCSI, then generates a reselected interrupt.

a) The ID specified in the BDID register is selected.

b) Not more than two bits are set on the SCSI data bus (excluding the parity bit).

c) When parity checking is enabled, a parity bit value is correct.

In the RESELECTION phase, the TEMP register holds the value of the SCSI data bus. The SPC serves as an INITIATOR from when this interrupt occurs until when the disconnected interrupt occurs or the RESET condition is detected in SCSI. Before starting the transfer operation in SCSI, be sure to reset the cause of this interrupt. If the disconnected interrupt is indicated together with the reselected interrupt, reset both of these interrupts simultaneously.

(3) Disconnected interrupt (Bit 5)

Bit 5 indicates that the BUS FREE phase has been detected in SCSI when bit 7 (Bus Free Interrupt Enable) of the PCTL register is set to 1. Also, when the SPC serves as an INITIATOR, bit 5 indicates transition to the BUS FREE phase in SCSI. After this interrupt condition has occurred, the next SELECTION/RESELECTION phase may be executed in SCSI. However, the SPC does not respond to SCSI until this interrupt condition is reset. If the disconnected interrupt condition is detected during hardware transfer (Transfer command) execution with the SPC serving as an INITIATOR, the SCSI operation stops, but the SPC internal transfer sequence continues until one of the following events is encountered:

- The internal data buffer register becomes empty in an input operation.
- The data prefetch sequence to the internal data buffer register is completed in an output operation.

When a disconnected interrupt occurs, check the SSTS register to confirm if the transfer operation has been completed.

(4) Command complete interrupt (Bit 4)

Bit 4 indicates that the Select command/Transfer command operation has been completed.

- Completion of Select command

This interrupt indicates that the SPC has acknowledged a response (BSY signal) from the selected bus device in SELECTION/RESELECTION phase execution. It indicates that the SELECTION /RESELECTION phase has completed in SCSI. The SPC serves as an INITIATOR after the SELECTION phase has been executed, and it serves as a TARGET after the RESELECTION phase has been executed.

- Completion of Transfer command (when the SPC serves as a TARGET)

This interrupt indicates that the number of bytes transferred equals the byte count specified in the transfer byte counter. Or, in an input operation with bit 0 (Termination Mode) of the SCMD register set to 1, this interrupt indicates transfer stop due to parity error being detected in the data received from SCSI. In either case, this interrupt occurs after checking that the ACK signal for the REQ signal of the last byte is inactive on SCSI during asynchronous mode transfer. This interrupt also occurs after checking that the number of ACK signals received

matches the number of REQ signals transmitted during synchronous mode transfer.

- Completion of Transfer command (when the SPC serves as an INITIATOR)

  When padded transfer mode is not specified, this interrupt indicates the byte count specified in the transfer byte counter has been completed. When padding mode transfer is performed, this interrupt indicates that the current transfer operation has been terminated due to another transfer phase requested in SCSI. In this case, the service required interrupt occurs at the same time. In the MESSAGE IN phase, this interrupt occurs while the ACK signal to the last byte is held active. Before resetting this interrupt, be sure to issue the Reset ACK/REQ command.

(5) Service required interrupt (Bit 3)

This interrupt indicates a request for MPU program intervention and only occurs if the SPC serving as an INITIATOR is put in either of the following conditions:

- SPC has received the Transfer command, but cannot start the transfer operation because the transfer phase specified by bits 2 to 0 of the PCTL register does not match that requested in SCSI.
- The SPC has stopped the current hardware transfer operation (Transfer command) because of a request for another transfer phase in SCSI. In this case, the transfer operation in SCSI stops immediately, but the SPC internal transfer sequence continues until one of the following events is encountered:
  - — Input operation
    The internal data buffer register becomes empty.
  - — Output operation
    The data prefetch sequence to the internal data buffer register is completed.

Therefore, when this interrupt occurs, read out the SSTS register and check if the SPC internal transfer operation is completed. If the service required interrupt occurs during an output operation, the data prefetched in the SPC internal data buffer register may remain (not sent to SCSI). To determine how to handle the remaining data (up to eight bytes), see the interrupt processing procedure described below. When the service required interrupt occurs, the MPU program examines a transfer phase request from SCSI and proceeds to execute the transfer operation using one of the following procedures: (Figures 3–8 and 3–9 show examples of interrupt processing procedure.)

- Hardware transfer

  The MPU program specifies the transfer phase pattern requested by SCSI as bit 2 to 0 of the PCTL register, and reissues the Transfer command.

- Manual transfer

  If this interrupt occurs during an output operation, the remaining data in the SPC internal data buffer register may have to be preserved. In this case, with bit 3 (Intercept Transfer) of the SCMD register set to 1, perform manual transfer and interrupt resetting. When the interrupted original transfer phase (output) is requested again after manual transfer by the above processing procedure, the suspended transfer operation can be restarted using the remaining data held in the data buffer register.

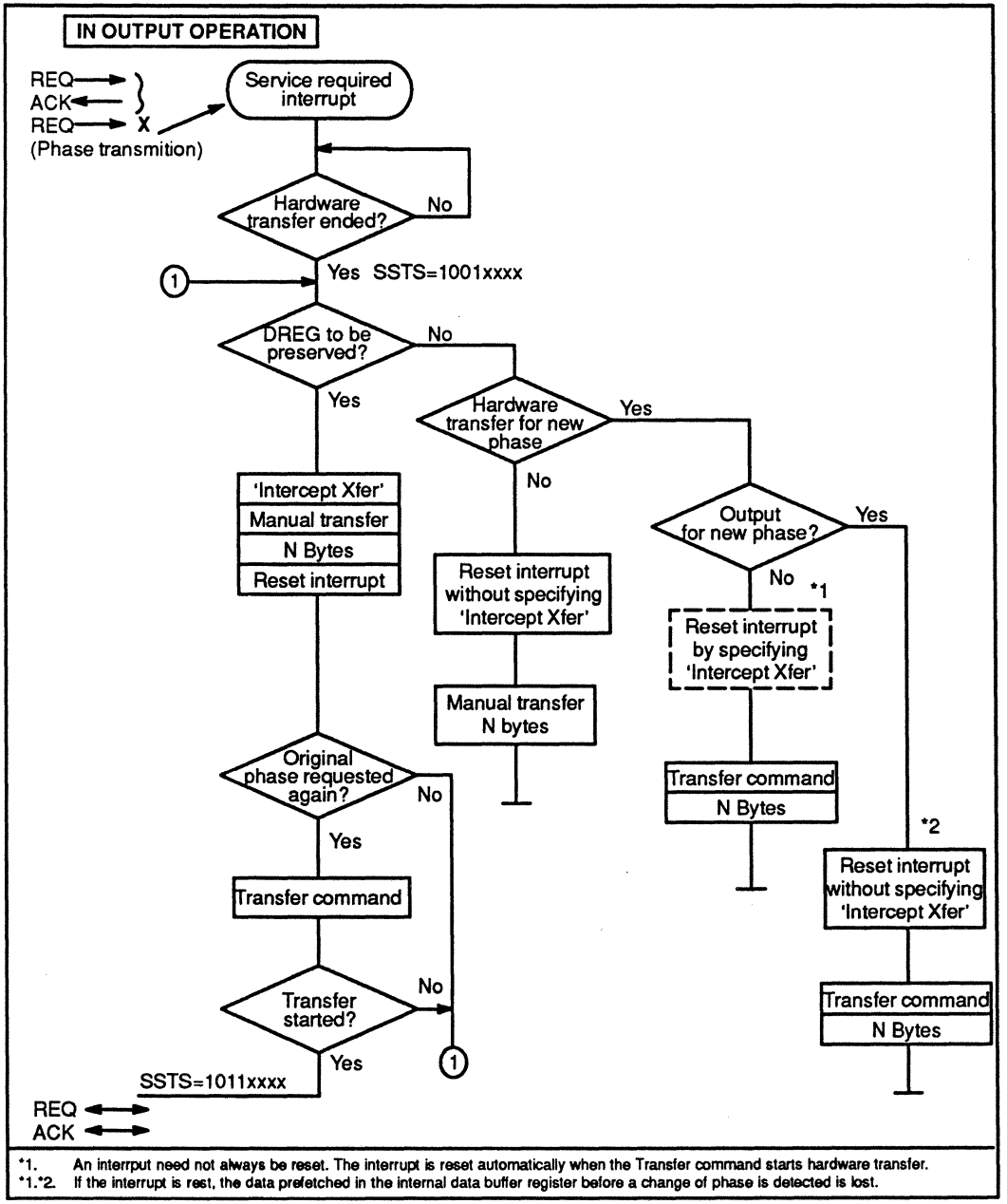


Figure 3–8. Service Required Interrupt Processing Procedure (Output)

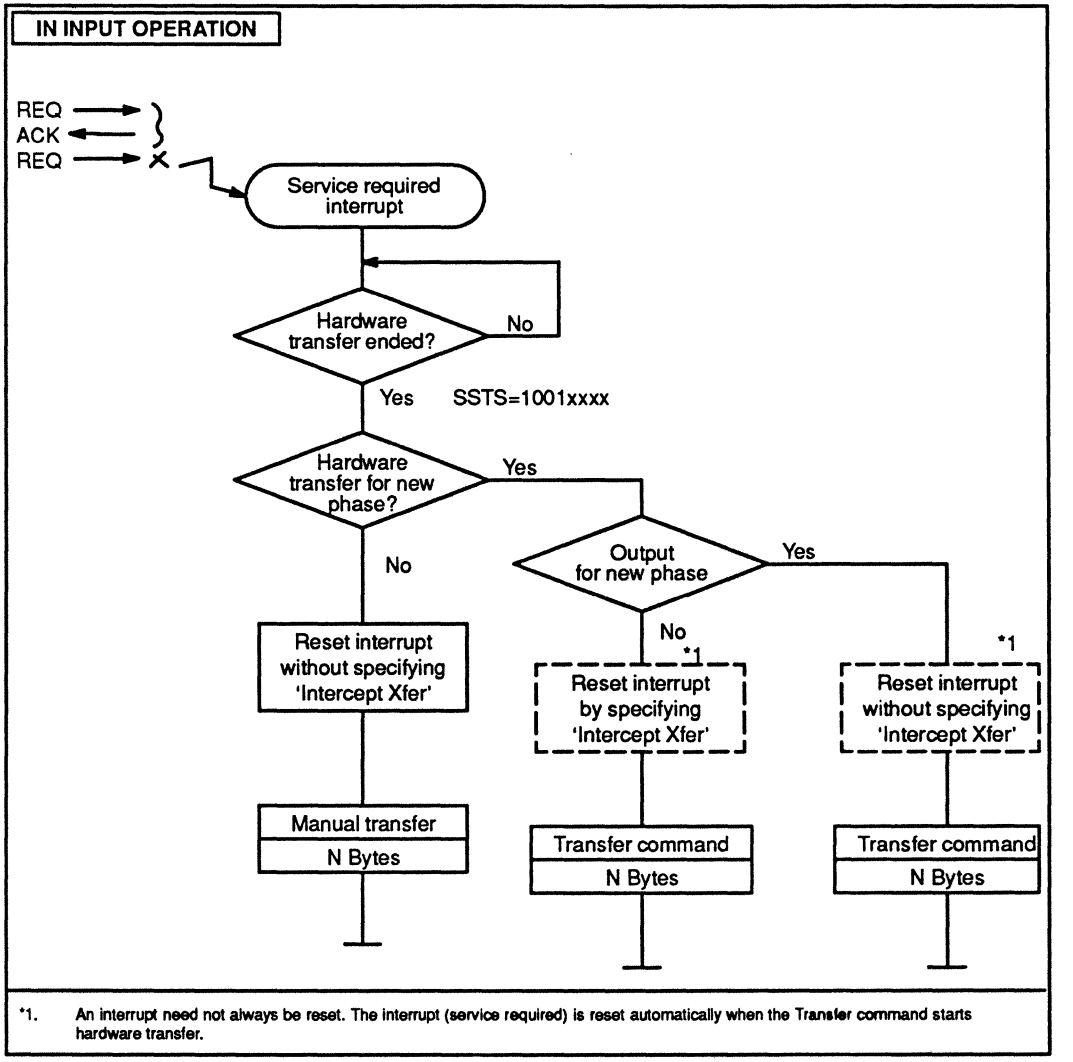


Figure 3–9. Service Required Interrupt Processing Procedure (Input)

- Time-out interrupt (Bit 2)

  This interrupt indicates that the selected bus device has not responded within the predetermined supervisory time after the SPC initiates the

SELECTION/RESELECTION phase. (See 3–2 Select command for details of the supervisory time setting procedure and the interrupt processing procedure.)

(7) SPC hardware error interrupt (Bit 1)

This interrupt indicates that the SPC has detected one of the following error indications in the SERR register:

- TC parity error
- Phase error
- Short transfer period
- Transfer offset error

See the SERR register for details of these errors. When this interrupt occurs, the SPC does not stop the operation being executed (for any error cause). Since the normal operational sequence and end report cannot always be guaranteed in this case, the MPU program must be used for SPC control and bus phase control of SCSI.

Figures 3–10 to 3–12 show examples of error control processing.

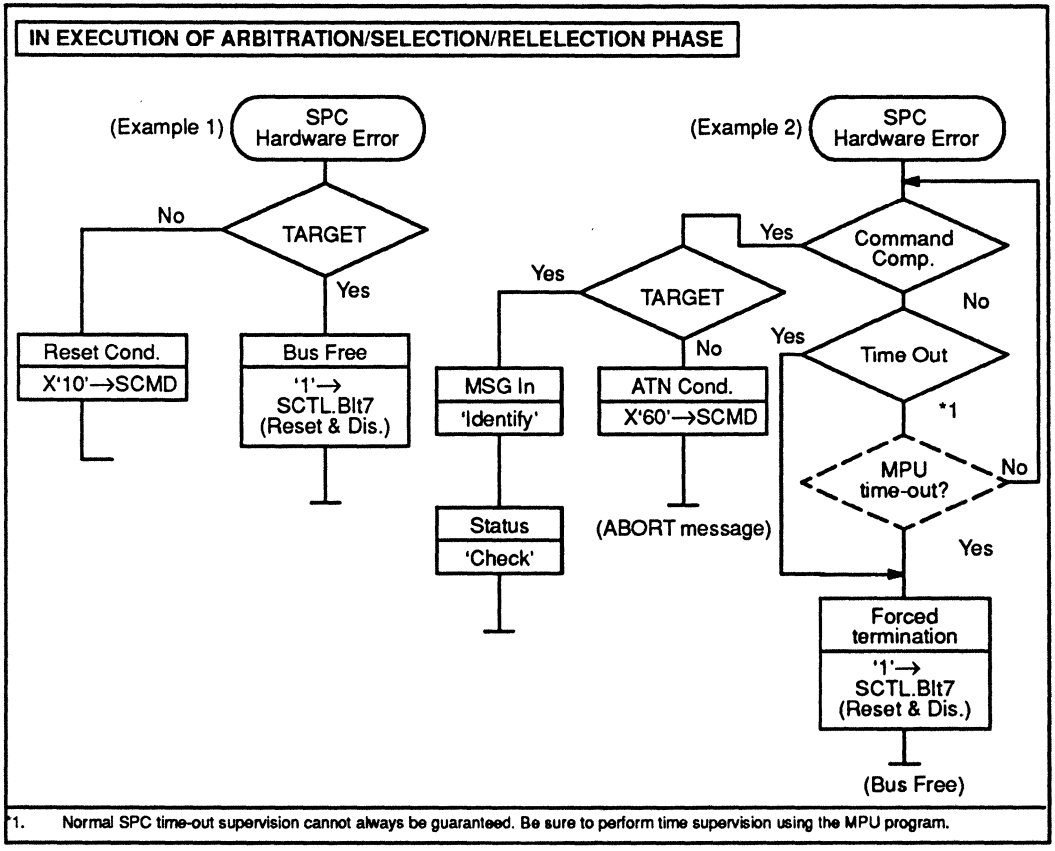


Figure 3–10. Example Of Control Processing For SPC-Detected Error (1)

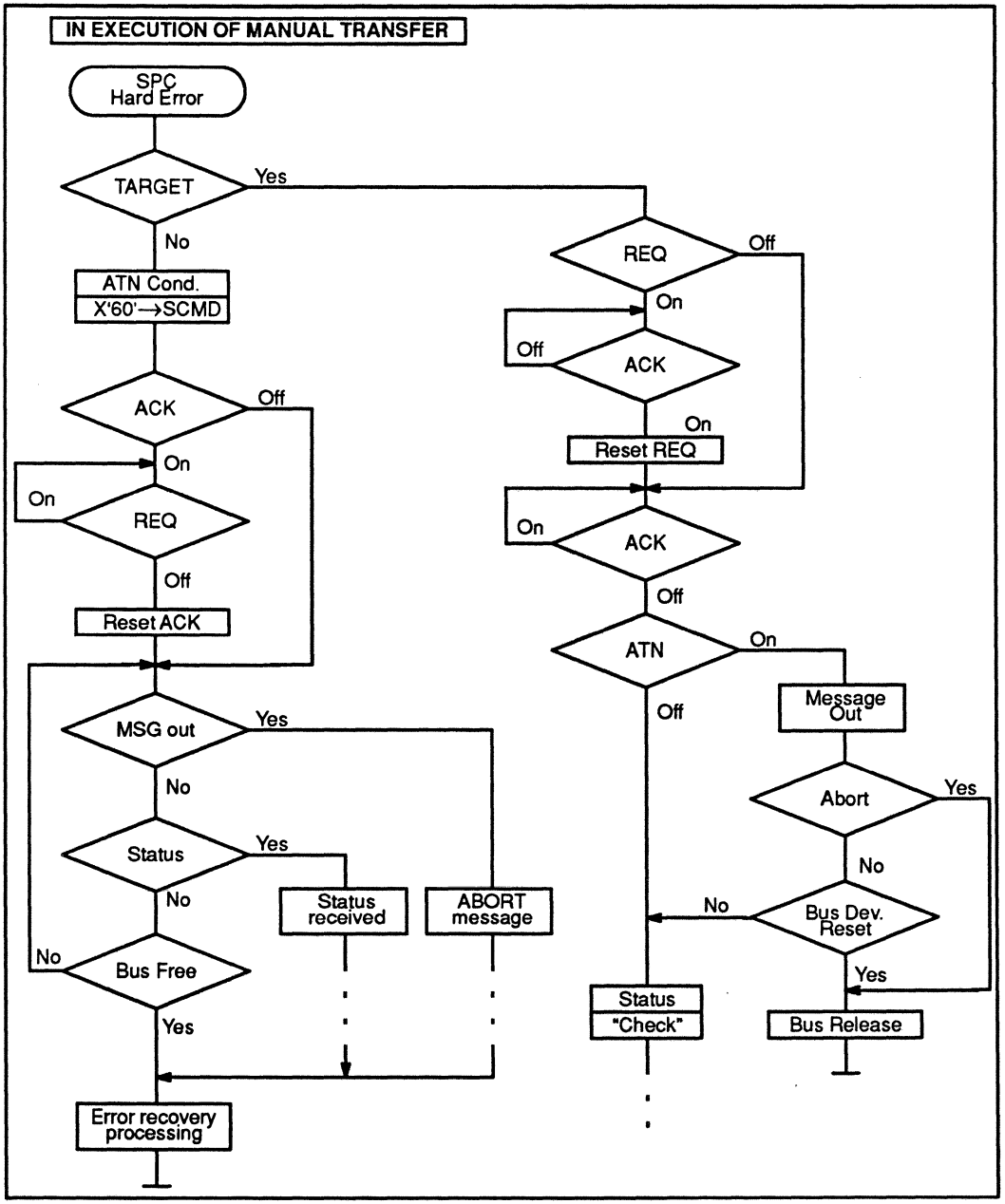


Figure 3–11. Example Of Control Processing For SPC-Detected Error (2)

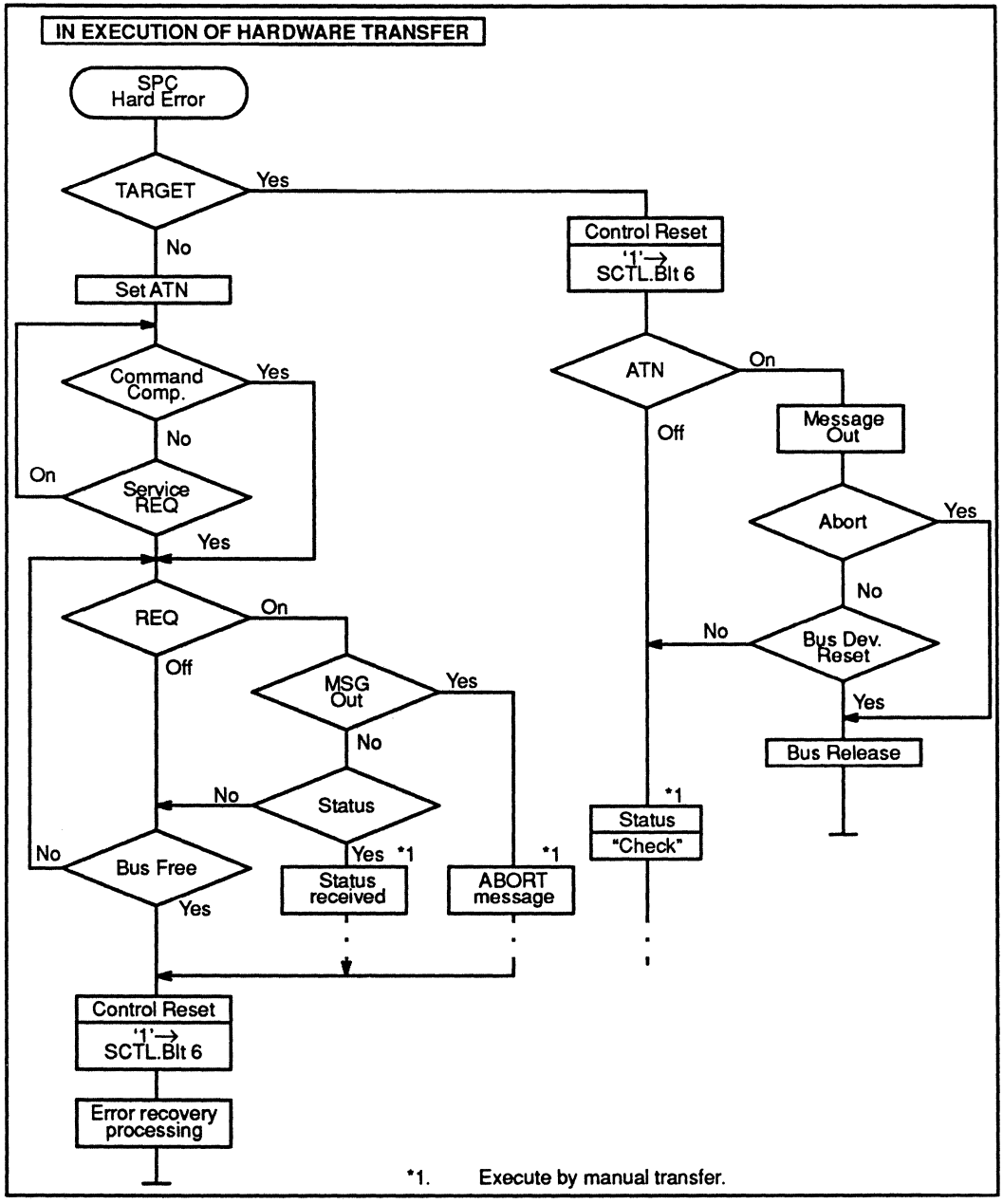


Figure 3–12. Example Of Control Processing For SPC-Detected Error (3)

(8) Reset condition interrupt (Bit 0)

This interrupt indicates that the RESET condition has been detected in SCSI. Note that this interrupt cannot be masked. The reset condition persists for an unpredictable period of time. After making sure that bit 3 (Reset In) of the SSTS register becomes 0, the MPU must reset this interrupt condition. The reset condition interrupt may occur regardless of whether the SPC is connected with SCSI or not. When the SPC is connected with SCSI, occurrence of this interrupt causes the SPC to immediately deactivate a signal being sent to SCSI. Then the SPC proceeds to the BUS FREE phase. When the SPC is executing a command, occurrence of this interrupt causes the SPC to terminate its operation and reset its internal state. However, the following internal registers hold the control information unchanged:

- BDID register
- SCMD register
- PCTL register
- SCTL register
- TMOD register
- Transfer byte counter

Until this interrupt is reset, the SPC internal reset state is maintained even if the RESET condition is released in SCSI. Therefore, the SPC does not respond even when a new bus phase (e.g., SELECTION) is executed in SCSI.

PSNS/SDGC
Address #5

PSNS Register – Phase Sense Register

OP	7	6	5	4	3	2	1	0	P
R	REQ	ACK	ATN	SEL	BSY	MSG	C/D	I/O	P

When bit 5 (Diagnostic Mode) of the SCTL register is set to 0, the PSNS register indicates the control signal status on SCSI (input signals to the SPC).

Bit	Signal	SPC input pins
7	REQ	REQI
6	ACK	ACKI
5	ATN	ATNI
4	SEL	SELI
3	BSY	BSYI
2	MSG	MSGI
1	C/D	C/DI
0	I./O	I/OI

A read from this register is allowed at any time regardless of the SPC condition. Bit 5 (ATN) of this register indicates the ATN signal status on SCSI. When the SPC serves as a TARGET, receipt of the ATN signal is indicated in this register but does not have any effect on other operations. The MPU program examines the ATN signal status at the following times in sequence and responds to the ATTENTION condition from SCSI:

(1) When a selected interrupt occurs

(2) When the RESELECTION phase has been completed

(3) When the Transfer command has been completed

(4) During execution of the Transfer command

When the ATN signal is detected, the Transfer Pause command can halt the transfer operation for transition to the MESSAGE OUT phase.

(5) During execution of manual transfer

When bit 5 (Diagnostic Mode) of the SCTL register is set to 1, the PSNS register indicates the status of control signals sent from the SPC to SCSI. In the pseudo SCSI operation using the SDGC register, the PSNS register can be used to check the SCSI control signal status.

PSNS/SDGC

Address #5

SDGC Register – SCSI Diagnostic Control Register

OP	7	6	5	4	3	2	1	0	P
W	Diag REQ	Diag ACK	*xfer enable	—	Diag BSY	Diag MSG	Diag C/D	Diag I/O	—
Note: *This bit is valid only for the MB89351/352 parts.									

The SDGC register is used to operate the SPC in the diagnostic mode (with bit 5 Diagnostic Mode of the SCTL register set to 1). To simulate the SCSI operation, the SDGC register bits are used as having alternative SCSI control signal lines. In diagnostic mode, the SDGC register is used to check SPC internal operation. The Diagnostic Mode stops signaling to the physical SCSI bus and nullifies input signals from the SCSI bus except the data bus signals. The SPC internal operation can be performed in the ordinary manner. To check SPC internal operation, the MPU can manipulate the SDGC register bits to generate input signals to and from SCSI.

The following bus phases can be simulated:

- BUS FREE
- ARBITRATION (Always win)
- SELECTION (For INITIATE operation)
- RESELECTION (For TARGET operation)
- Information transfer (Input data manipulation not allowed in an input operation). Bit 5 is used as an enable bit for issuing interrupt signal (INTR) when the SPC FIFO needs servicing.

Bit 5 is set to enable a hardware interupt (INTR) when the SPC FIFO needs to be serviced. During an SCSI input operation, an interrupt will occur when the SPC has data to be read. During an SCSI ouput operation, an interrupt will occur when the SPC needs a byte to be written into the FIFO. Bit 5 can be SET/RESET during NORMAL and DIAGNOSTIC mode of SPC operation. This bit is write only, it will be read back as '0' regardless of its setting.

INITIALIZATION

The SPC is in the reset state when the input to $\overline{RST}$ pin is low (hardware reset) or bit 7 (Reset and Disable) of the SCTL register is set to 1. SPC-SCSI operation is disabled until the SPC reset state is released. The MPU program shall release the reset state after initializing the SPC internal registers. The following SPC internal registers remain unchanged even in the SPC reset state:

- BDID register
- TMOD register
- Transfer byte counter
- SCMD register
- PCTL register
- TEMP register (for sending)

At the moment power is turned on, the contents of these internal registers are unpredictable even though hardware reset ($\overline{RST}$ input = low) is executed. After $\overline{RST}$ input is released, the MPU program shall initialize the SPC as shown in Figure 3–13 .

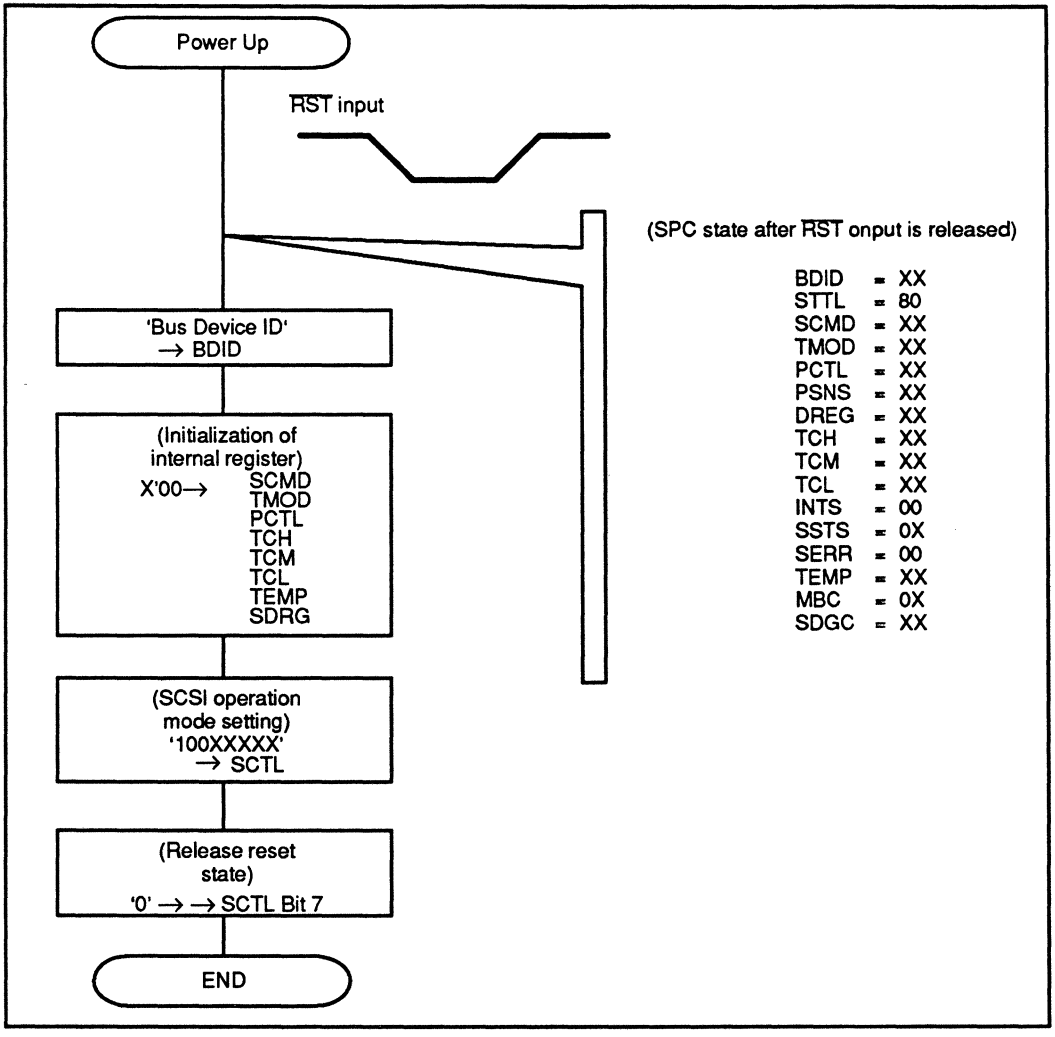


Figure 3–13. SPC Initialization (at power-up)

SSTS Address #6

SSTS Register – SPC Status (Register)

OP	7	6	5	4	3	2	1	0	P
R	CONNECTED INIT	CONNECTED TARG	SPC Busy	Xfer in Progress	SCSI Reset In	TC=0	DREG FULL	Status EMPTY	P

Register Functions

The SSTS register indicates the SPC internal status, which can be read out at any time.

Bit 7 and 6: Connected (INIT, TARG)

These bits indicate the connecting status between the SPC and SCSI.

Bit 7: INIT	Bit 6: TARG	State
0	0	Not connected with SCSI
1	0	SPC serves as an INITIATOR: • During execution of the SELECTION phase and after its completion. • After a reselected interrupt
0	1	SPC serves as a TARGET: • During execution of the RESELECTION phase and at its completion. • After a selected interrupt
1	1	Undefined

Bit 5: SPC Busy

Bit 5 indicates that a command is being executed or is waiting to be executed.

Bit 4: Transfer In Progress

Bit 4 indicates that a hardware transfer operation is being executed or the SCSI is requesting an information transfer phase.

Bit 3: SCSI Reset In

Bit 3 indicates that the SCSI RST signal is active.

Bit 2: TC = 0 (Transfer Counter Zero)

Bit 2 indicates that the transfer byte counter (TCH, TCM, TCL) has reached 0.

Bits 1 and 0: DREG Status (Full, Empty)

These bits indicate the status of the internal data buffer register. When executing a hardware transfer operation in program transfer mode, determine the data buffer register access timing by using these bit values.

Bit 1: FULL	Bit 0: EMPTY	Data buffer register status
0	0	Holds 1 to 7 bytes of data
0	1	Empty
1	0	Holds 8 bytes of data, leaving no free space
1	1	Undefined

SPC Status

SPC status is represented by combinations of status bits 7 to 4 of this register. Table 3–4 lists the combination of these bits and SPC status.

Table 3–4. SPC Operating Status Indications

SSTS register				SPC operating status
Bit 7: INIT	Bit 6: TARG	Bit 5: SPC Busy	Bit 4: Xfer in Progress	
0	0	0	0	SPC is not connected with SCSI. SPC does not hold a command waiting to be executed.
0	0	1	0	SPC is not connected with SCSI. But SPC holds the Select command, which is waiting for BUS FREE phase. Or it is executing ARBITRATION phase.
0	1	0	0	SPC serves as a TARGET. No operation is being executed in SCSI. Manual transfer is not being executed.
0	1	1	0	SPC is executing RESELECTION phase on SCSI.
0	1	1	1	SPC serves as a TARGET. Hardware transfer operation (Transfer command) is being executed.
1	0	0	0	SPC serves as an INITIATOR. No operation is being executed in SCSI. Manual transfer is not being executed.
1	0	0	1	SPC serves as an INITIATOR. Although SPC has received a REQ signal from SCSI, it is not ready to start transfer operation because no Transfer command has been issued or transfer phase does not match.
1	0	1	0	SPC is executing the SELECTION phase on SCSI.
1	0	1	1	SPC serves as an INITIATOR. Hardware transfer operation (Transfer command) is being executed.

SERR Address #7

SERR Register – SPC Error Status Register

OP	7	6	5	4	3	2	1	0	P
R	Data Error		'0'* xfer out**	'0'	TC Parity Error	Phase Error	Short Xfer Period	Xfer Offset Error	P
	SCSI	SPC							
Note: * MB87030/31/33 **MB89351/52									

Note: **MB89351/352
This bit can only be reset (including the INTR signal) by satisfying the data request by writing or reading to the SPC FIFO (DREG).

Register Functions

The SERR register provides details of an error detected in the SPC. An SPC hardware error interrupt occurs if an error is indicated at any of bits 3 to 0.

Bits 7 and 6: Data Error

These bits indicate that a parity error has been detected in the transferred data during transfer phase execution in SCSI. Table 3–5 lists the data error indication bit patterns and the relevant SPC operations. When changing the transfer phase in SCSI, these error indication bits must be reset.

Table 3–5. Data Error Indication Bit Patterns In Transfer Phase

Bit 7	Bit 6	SPC operations
Data Error		
SCSI	SPC	
0	1	No parity error was detected in the transferred data.
0	1	During execution of an output operation in hardware transfer mode, a parity error was detected in the data to be sent to SCSI. The parity is checked regardless of the value of bit 3 (Parity Enable) of the SCTL register. The erroneous data (parity bit value) is corrected and then sent to SCSI.
1	1	A parity error was detected in the data received from SCSI during an input operation. The parity is checked only when bit 3 (Parity Enable) of the SCTL register is set to 1. After an error is detected, the parity bit is corrected. If the SPC serving as an INITIATOR detects this error during hardware transfer execution, it generates an ATN signal to SCSI. (See Figure 3–2) In this case, the MPU program must reset this error condition before the ATN signal is reset. If the SPC serving as a TARGET detects this error during hardware transfer execution, it follows the specification at bit 0 (Termination Mode) of the SCMD register See SCMD register for details.
1	0	Undefined

Bit 5: Xfer Out

When bit 5 of SDGC is one (1) in program transfer mode, interrupt signal, INTR, is output for a data request. This bit is a flag indicating the data request.

In the MPU, this bit is referenced in the interrupt routine to perform data transfer according to the data requests.

Bit 3: TC Parity Error

Bit 3 indicates that a parity error occurred while the transfer byte counter (TCH, TCM, TCL) was being decremented.

Bit 2: Phase Error

When the SPC serves as an INITIATOR, the transfer phase has been changed in SCSI during hardware transfer mode operation (service required interrupt occurs). In this case, bit 2 indicates that:

- The new phase is a synchronous transfer mode DATA IN phase.
- The REQ signal has been received two or more times before the MPU program completed interrupt processing and issued the Transfer command for the new phase. In this case, the SPC cannot receive data and return the ACK signal normally.

Bit 1: Short Transfer Period

Bit 1 indicates that the REQ/ACK signal (input signal to REQI/ACKI pin) has a cycle exceeding the specified input range see Figure 3–14). If this error occurs, the transfer sequence executed by the SPC is not guaranteed.

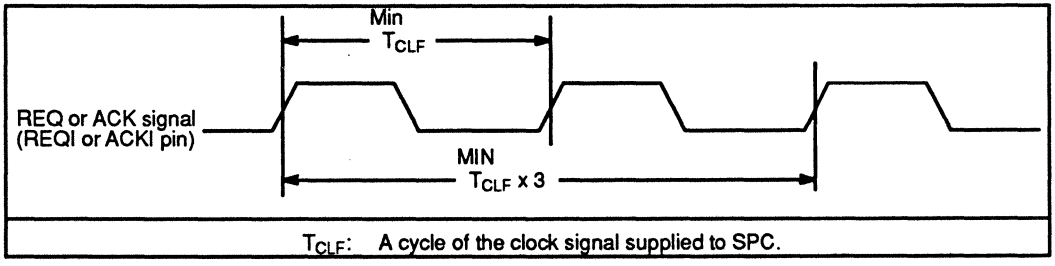


Figure 3–14. Specified Input Cycle Of REQ/ACK Signal

Bit 0: Transfer Offset Error

Bit 0 indicates that one of the following errors has been detected during synchronous transfer mode. (The offset value referred to below denotes the REQ/ACK maximum offset value specified in the TMOD register).

- When SPC serves as a TARGET:
- The number of ACK signals received exceeds that of REQ signals transmitted.
- The number of REQ signals transmitted exceeds the offset value (SPC malfunction).
- When SPC serves as an INITIATOR
- The number of REQ signals received exceeds the offset value.
- The number of ACK signals transmitted exceeds that of REQ signals received (SPC malfunction). If this error occurs, the transfer sequence executed by the SPC is not guarantee.

Error Reset

To reset an error indication in the SERR register, do one of the following:

- Generate SCSI RESET condition (X10 → SCMD)
- Reset and disable (Bit 7 of SCTL register)
- Control reset (Bit 6 of SCTL register)
- Interrupt (SPC hardware error) reset (X02 → INTS)
  Bits 7 and 6 (Data Error) of this register do not cause an interrupt, but can be reset by this means.

PCTL
Address #8

PCTL Register – Phase Control Register

OP	7	6	5	4	3	2	1	0	P
R / W	Bus Free Interrupt Enable	'0'				MSG Out	C/D Out	I/O Out	P

Bit 7: Bus Free Interrupt Enable

With bit 7 set to 1, detection of the BUS FREE phase on the SCSI causes a disconnected interrupt to occur. To prevent an undesired interrupt from occurring, be sure to set bit 7 to 0 in the following cases:

- When issuing the Select command
- When resetting a 'disconnected' interrupt

Bit 2: MSG Out

Bit 1: C/D Out

Bit 0: I/O Out

When the SPC serves as a TARGET, specify the information transfer phase to be executed in SCSI. These bit values are sent to SCSI as MSG, C/D and I/O signals.

When the SPC acts as an INITIATOR, specify the pattern indicating the transfer phase to be executed. Before executing the transfer operation, the specified transfer phase pattern is compared with a bus phase actually requested by the TARGET. If they match, the transfer operation is initiated. Table 3–6 shows how to set a transfer command.

Also, use bit 0 (I/O OUT) to specify SELECT or RESELECT operation when the select command is issued to the SPC. See SELECT command for details.

Table 3–6. Transfer Phase Setting

Bit 2 MSG Out	Bit 1 C/D Out	Bit 0 I/O Out	SCSI transfer phase
0	0	0	Data Out
0	0	1	Data In
0	1	0	Command
0	1	1	Status
1	0	0	Unused
1	0	1	Unused
1	1	0	Message Out
1	1	1	Message In

MBC
Address #9

MBC Register – Modified Byte Control Register

OP	7	6	5	4	3	2	1	0	P
R	'0'				MBC				
					Bit 3	2	1	0	P

The MBC register controls the data count during transfer between the SPC internal data buffer register and the MPU (Program Transfer mode) or external buffer memory (DMA mode). When data are written into the TCL register, its four low-order bits are set as an initial value for the MBC register. In an output operation, data are prefetched into the SPC internal data buffer register. Each time one byte is prefetched, the MBC register is decreased by one.

Data prefetch stops when the transfer byte counter is decreased below 15 and the MBC register reaches 0. In an input operation, data received from SCSI is stored in the internal data buffer register. Each time data is sent to the MPU or external buffer memory, the MBC register is decreased. The difference between the transfer byte counter and the MBC register corresponds to the byte count of data remaining in the internal data buffer register.
While the SPC is executing a DMA mode transfer operation, this register must not be read.

DREG
Address
#OA

DREG Data Buffer Register

OP	7	6	5	4	3	2	1	0	P
R/W	Internal Data Register (8 Bytes FIFO)								
	Bit 7	6	5	4	3	2	1	0	P

The SPCs internal data buffer register consists of eight bytes and operates on the FIFO principle. When executing the Transfer command in Program Transfer Mode, MPU transfers data using this register. Figure 3–15 shows the DREG access procedure to be taken in Program Transfer Mode. Therefore, never repeat accessing more than the number of times required.

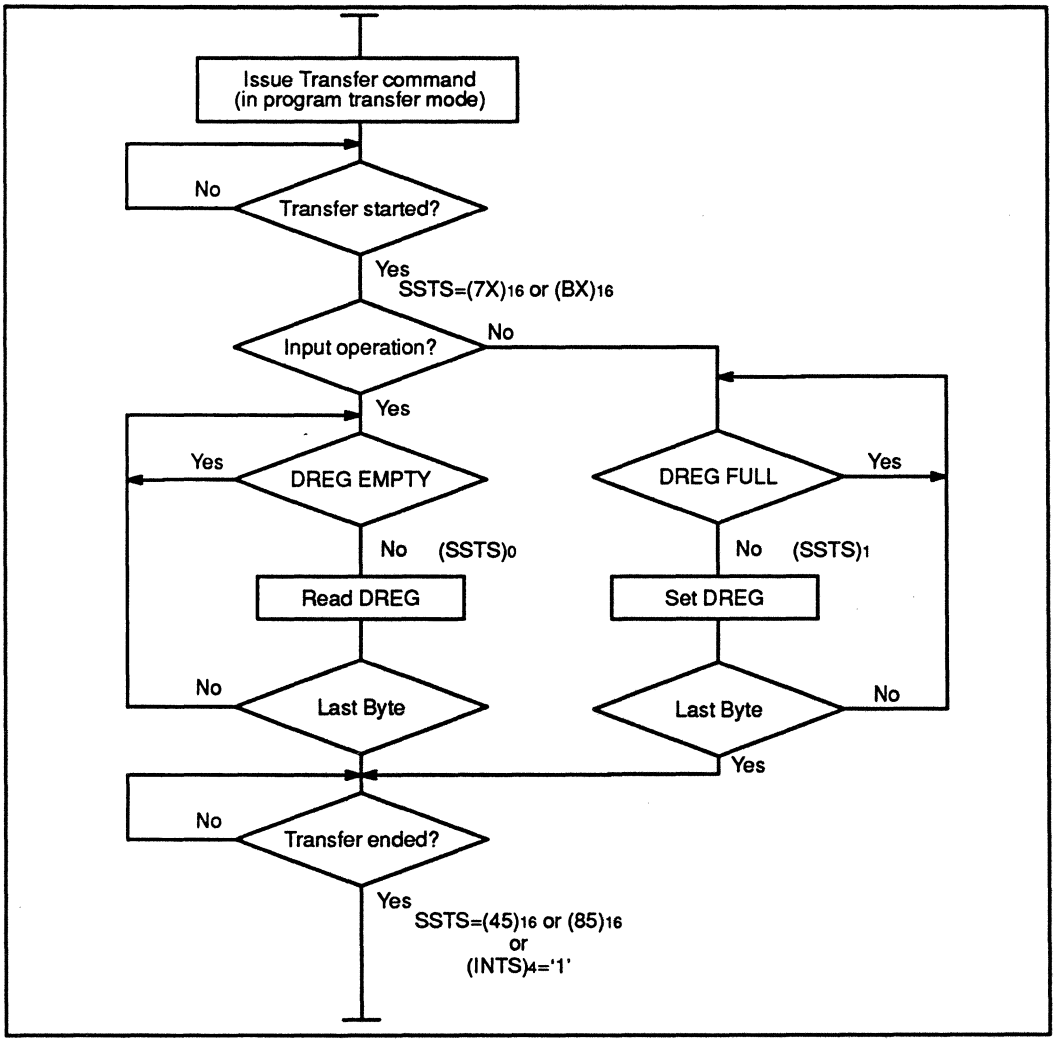


Figure 3–15. DREG Access Procedure In Program Transfer Mode

TEMP
Address
#OB

TEMP REGISTER – Temporary Register

OP	7	6	5	4	3	2	1	0	P
R	Temporary Data (Input: From SCSI)								
	Bit 7	6	5	4	3	2	1	0	P
W	Temporary Data (Output: To SCSI)								
	Bit 7	6	5	4	3	2	1	0	P

The TEMP register is used for controlling the SCSI data bus except when hardware transfer is executed. It consists of two bytes, each of which is dedicated exclusively to receiving/sending data.

- Data receiving element (read only)

  a) When a SELECTION/RESELECTION phase is detected in SCSI, the contents on the SCSI data bus are saved in the TEMP register. If a selected/reselected interrupt occurs, a bus device can be identified by the TEMP register

contents. Read this register before resetting the selection or reselection interrupt status bit.

b) For a manual transfer input operation, the contents on the SCSI data bus are saved in the timing sequence shown in Figure 3–16.

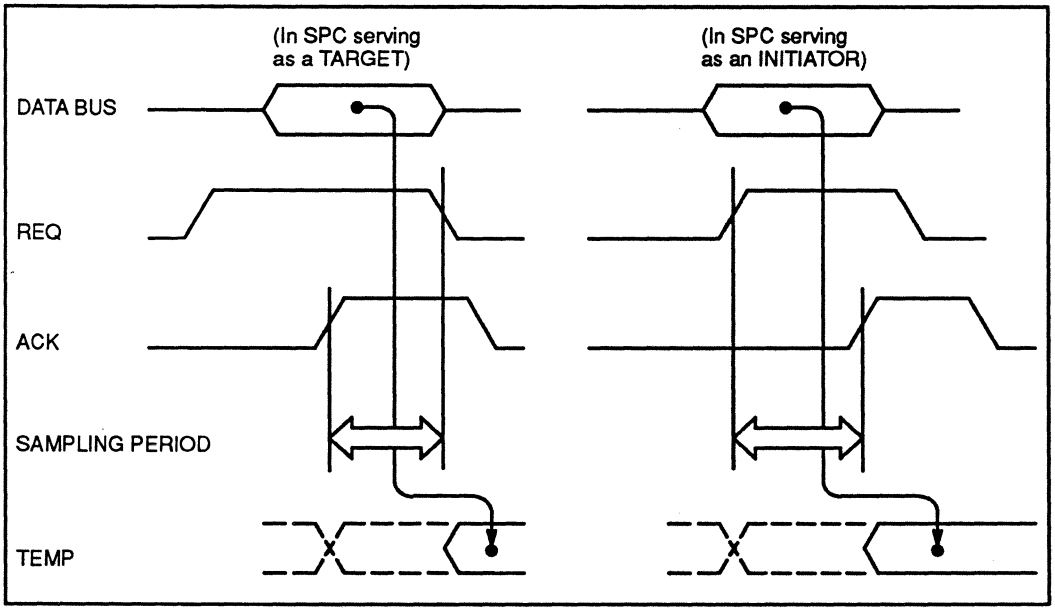


Figure 3–16. Data Bus Save In Manual Transfer Input Operation

(2) Data sending element (write only)

a) Before issuing the Select command, set the contents to be sent to the SCSI data bus in the SELECTION/RESELECTION phase.

b) For a manual transfer output operation, set the data to be sent out.

TCH, TCM, TCL

Address #OC, OD, OE

(TCH, TCM, TCL) – Transfer Byte Counter, High, Middle, and Low

OP	7	6	5	4	3	2	1	0	P
R/W	TCH: Transfer Counter (MSB)								
	Bit 23	22	21	20	19	18	17	16	P
R/W	TCM: Transfer Counter (2nd Byte)								
	Bit 15	14	13	12	11	10	9	8	P
R/W	TCL: Transfer Counter (LSB)								
	Bit 7	6	5	4	3	2	1	0	P

The transfer byte counter consists of three bytes and functions as a down counter. In execution of a hardware transfer operation, it is decreased by one each time one byte of data is transferred over SCSI. It indicates the remaining byte count of data to be transferred. In the Select command execution, this counter operates as a response waiting time supervisory timer and sequence control counter. See Select command and Transfer command for transfer byte counter initialization. While the transfer byte counter is operating, do not carry out a read/write.

Note: Do not read/write these registers while a transfer operation is occuring. If necessary, use the "TC = 0" bit in the SSTS register to determine when the transfer count has reached zero.

EXBF
Address #OF

EXBF – External Buffer Register MB87030, 31, 33 only

OP	7	6	5	4	3	2	1	0	P
R/W	External Buffer								P
	Bit 7	6	5	4	3	2	1	0	

This register's address is reserved for access from the MPU data bus (D7 to D0, DP) to the DMA data bus (HDB7 to HDBO, HDBP). It does not not exist as an internal register in SPC but provides a pathway between buses. As shown in Figure 3–17, the MPU program can execute a write/read to/from the external buffer memory by using this virtual register.

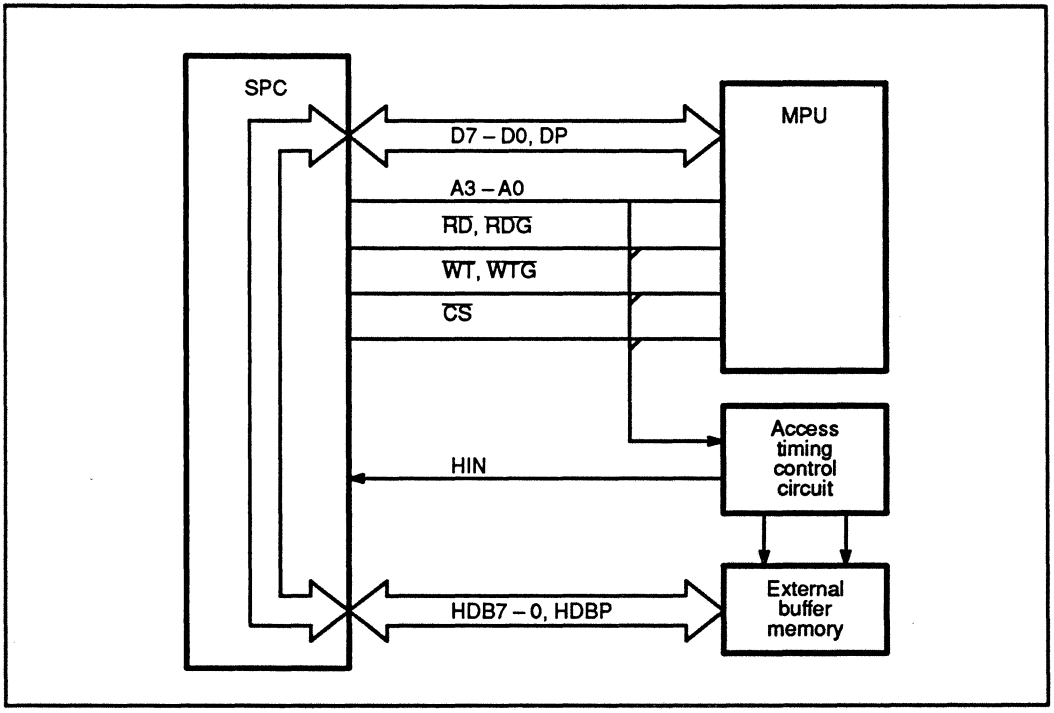


Figure 3–17. Access From MPU to External Buffer Memory

Devices without a separate DMA data bus do not have this register (MB89351/2).

This register is normally used by the MPU to load/unload an external memory buffer before/after (write to SCSI/read from SCSI) a DMA transfer.

Chapter 4

EXAMPLES OF EXTERNAL CIRCUIT CONNECTIONS

Figure 4–1 shows the external circuit blocks that can be connected with this LSI circuit. The external circuit configuration depends on the application environment, intended purpose, and required performance. Application examples for this circuit are given below.

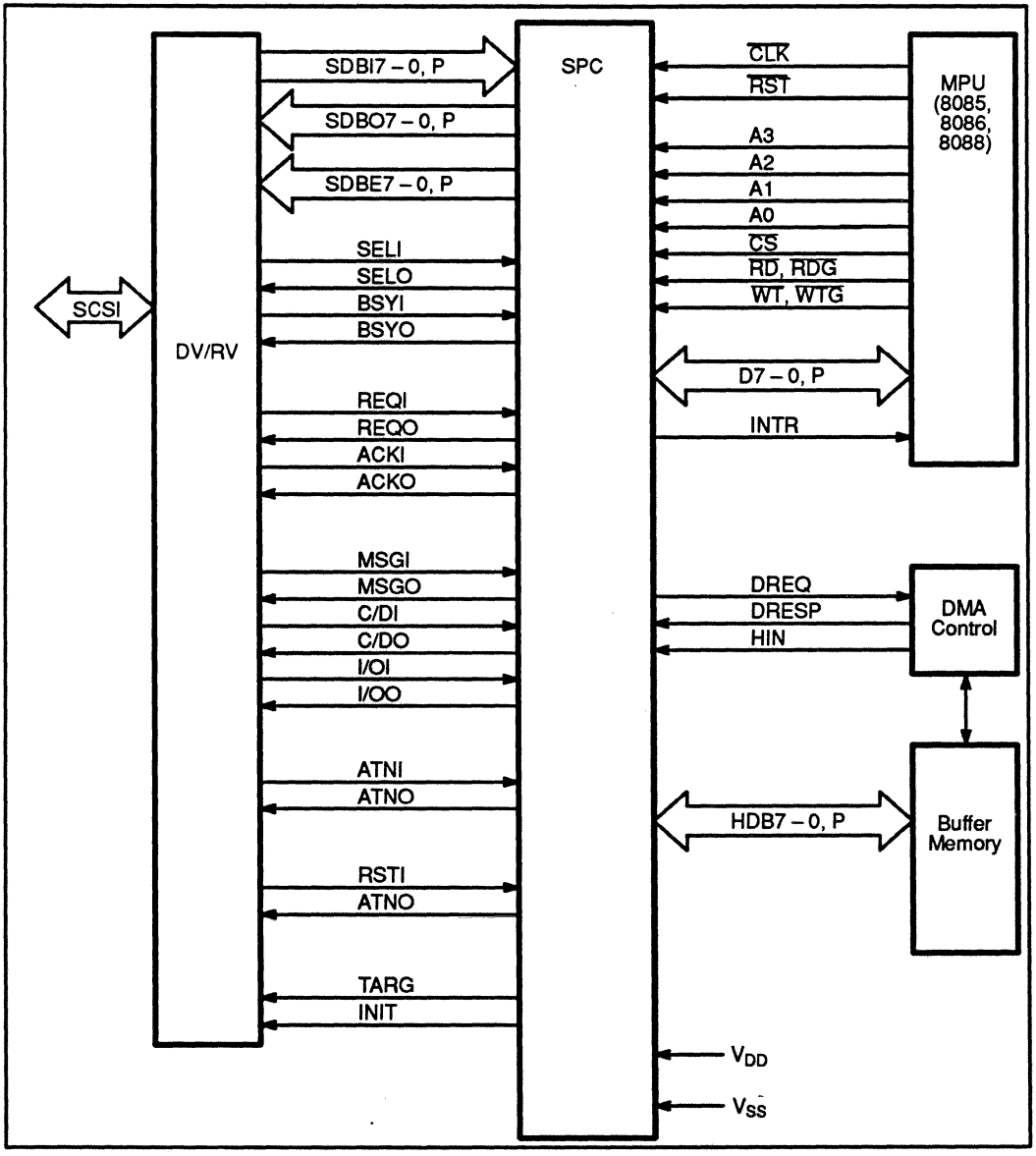


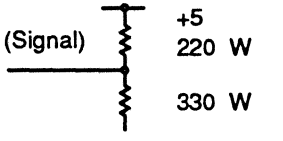
Figure 4–1. Examples of External Circuit Connections

SCSI Driver/Receiver Circuits

Single-Ended Type

Table 4–1 lists the major components of this SCSI driver/receiver circuit. Figure 4–2 shows a example of a connection of the SPC and the SCSI single-ended type driver/receiver circuits.

Table 4–1. Major Components of Single-Ended Type SCSI Driver/Receiver Circuit (example)

Component	Part No.	Manufacturer	Characteristics	Q'ty
REQ/ACK signal driver	MB412 *	Fujitsu	3-state buffer circuit) 2 circuits/DIP 14	1
REQ/ACK signal receiver	MB413 * Resistor Resistor Capacitor	Fujitsu — — —	4 circuits/DIP 16 390W ±2% 1/4 W 200W ±2% 1/4 W 0.1mF/50 V Ceramic	1 1 1 1
Other signal driver	MB463 *	Fujitsu	(Open-collector buffer circuit) 4 circuits/DIP 14	4
Other signal receiver	74LS240	Fujitsu TI	(Schmitt trigger inverter) 8 circuits/DIP 20	2
Terminator (Required only when the driver/receiver is located at either end of SCSI)	—	—		18 elements

Notes:
The MB412 is compatible with the SN7519.
The MB413 is compatible with the Am26LS32.
The MB463 is compatible with the SN7438.

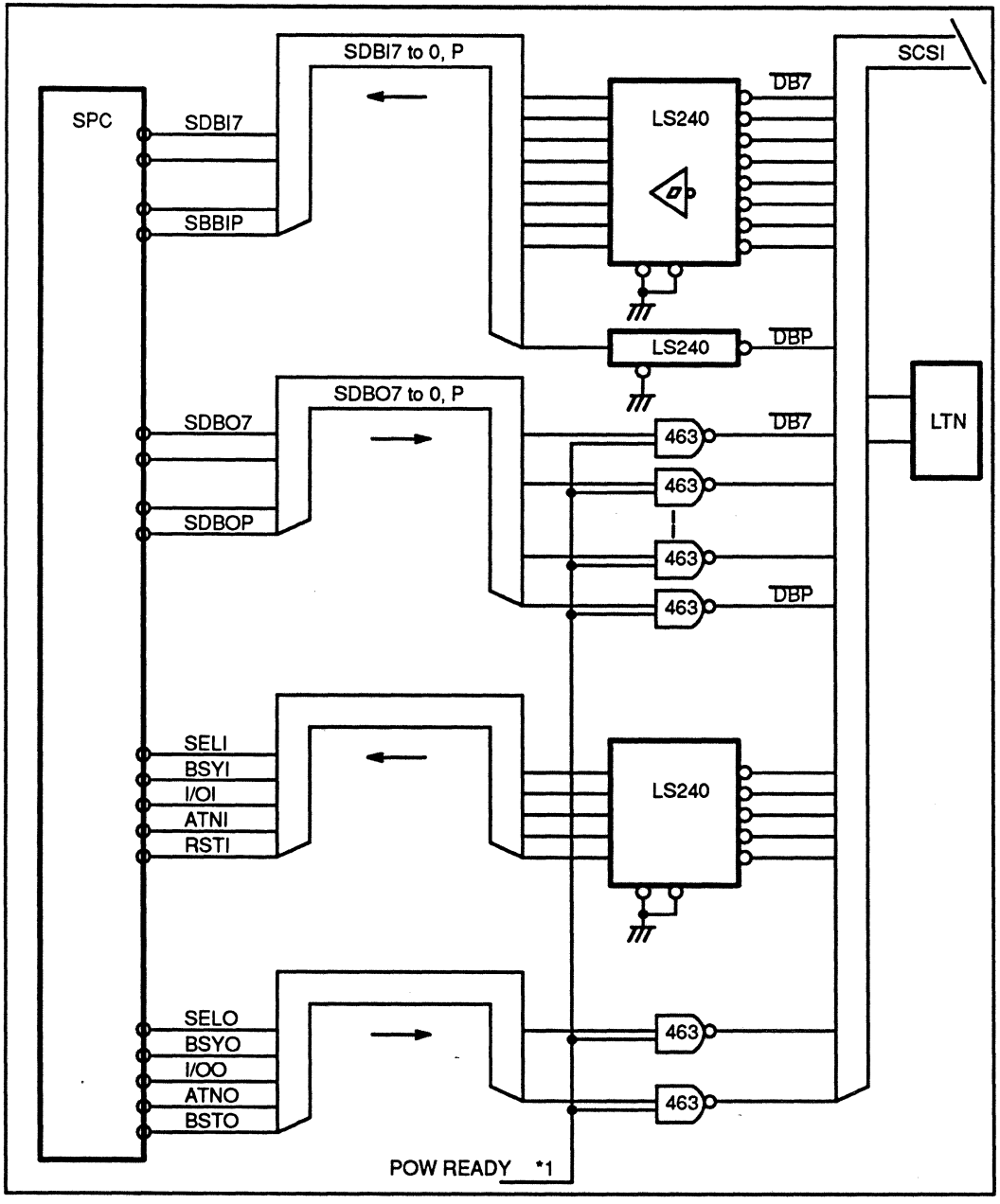


Figure 4–2. Example of Single-Ended Type SCSI Driver/Receiver Circuit

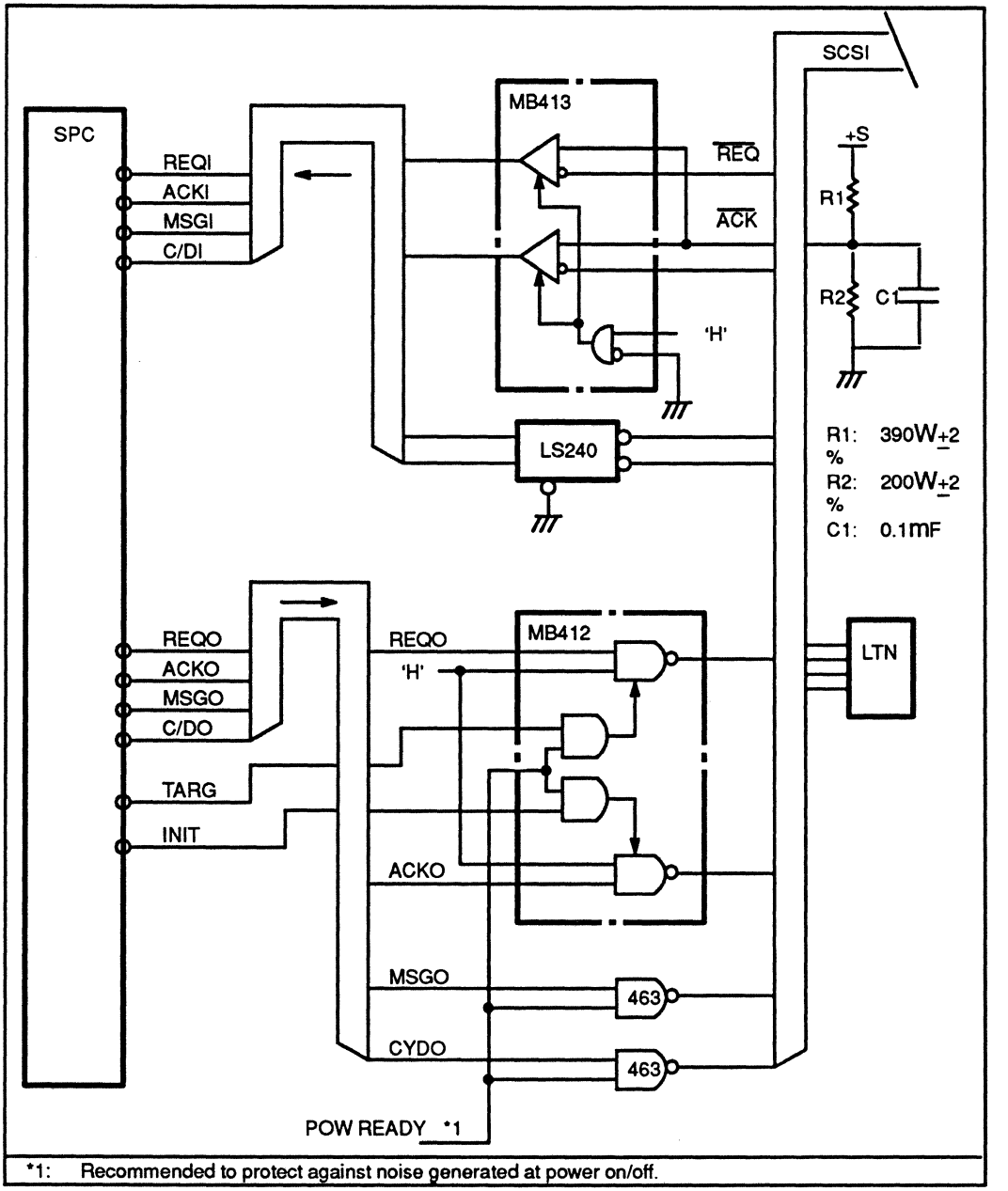
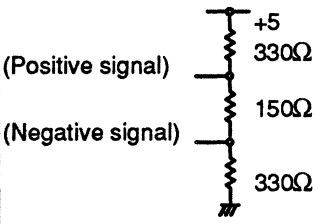


Figure 4–2. Example of Single-Ended Type SCSI Driver/Receiver Circuit (continued)

Differential Type

Table 4–2 lists the major components of this SCSI driver/receiver. Figure 4–3 shows a example of the connection of the SPC and the SCSI differential type driver/receiver circuit.

Table 4–2. Major Components of Differential Type SCSI Driver/Receiver Circuit (example)

Component	Part No.	Manufacturer	Characteristics	Q'ty
Driver/receiver (Common to all signals)	SN75176	TI	(Differential transceiver 1DV +1RV/Dip 8	18
Terminator (Required only when the driver/receiver is located at either end of SCSI)	—	—		18 element

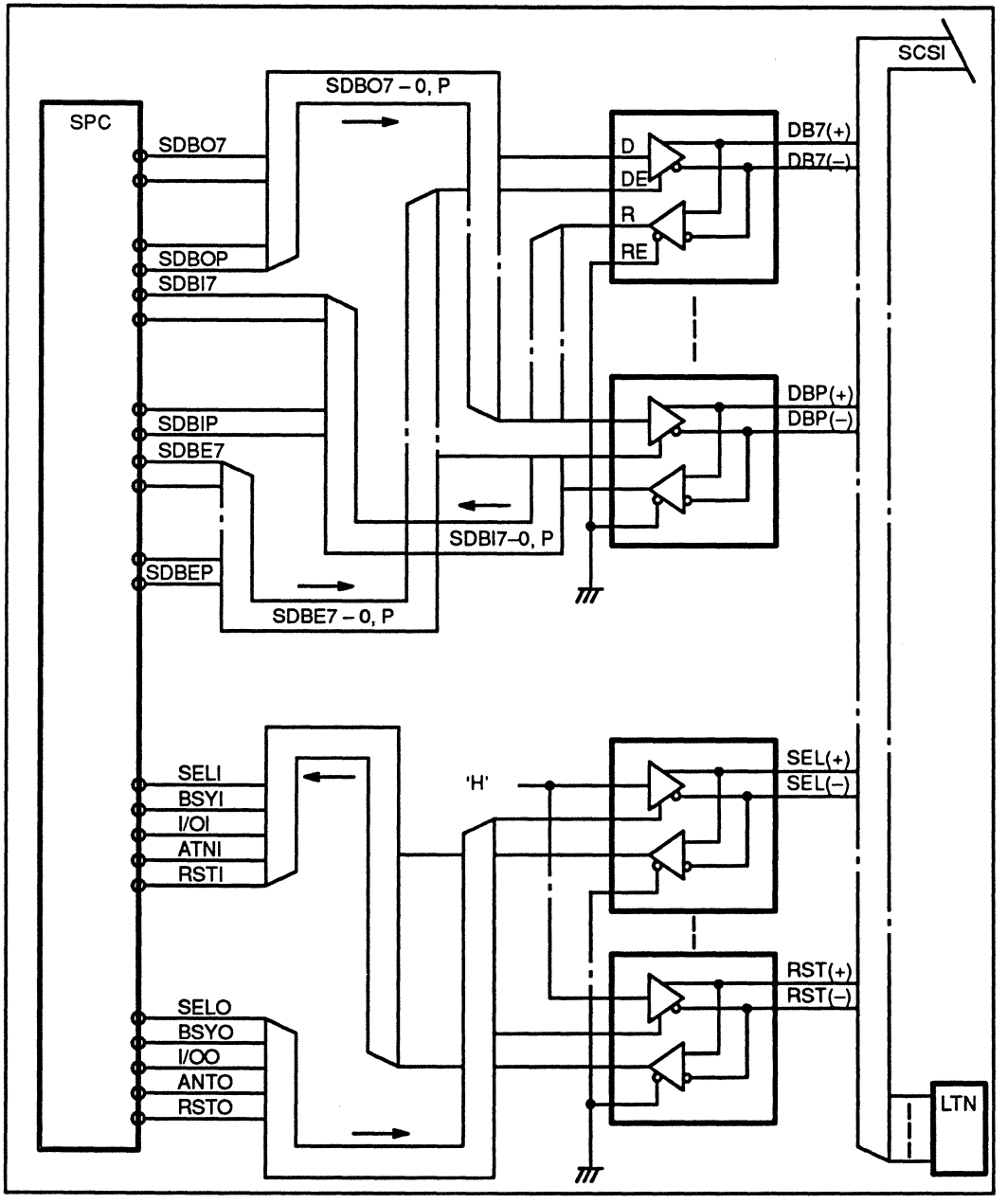


Figure 4–3. Example of Differential Type SCSI Driver/Receiver Circuit

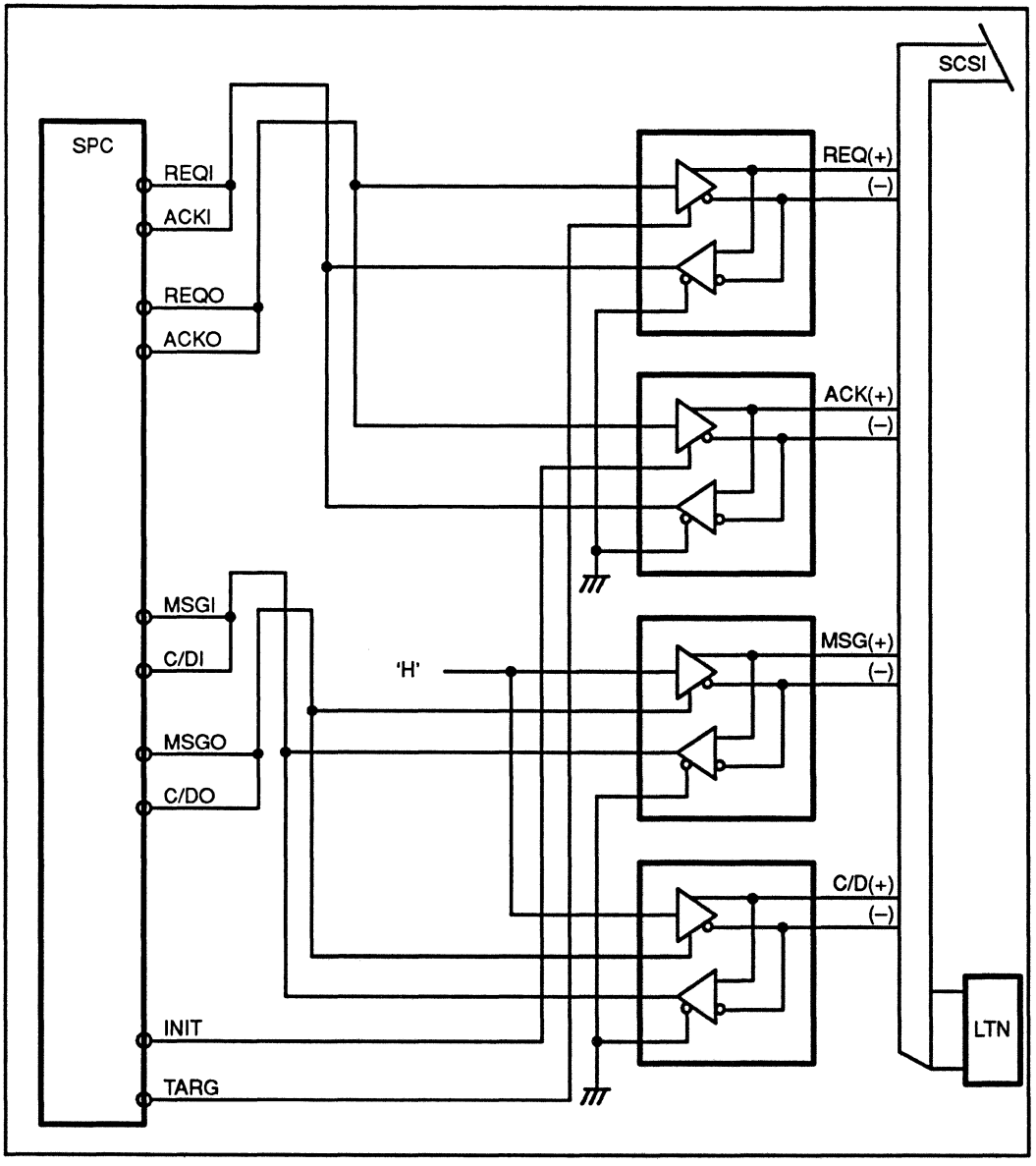
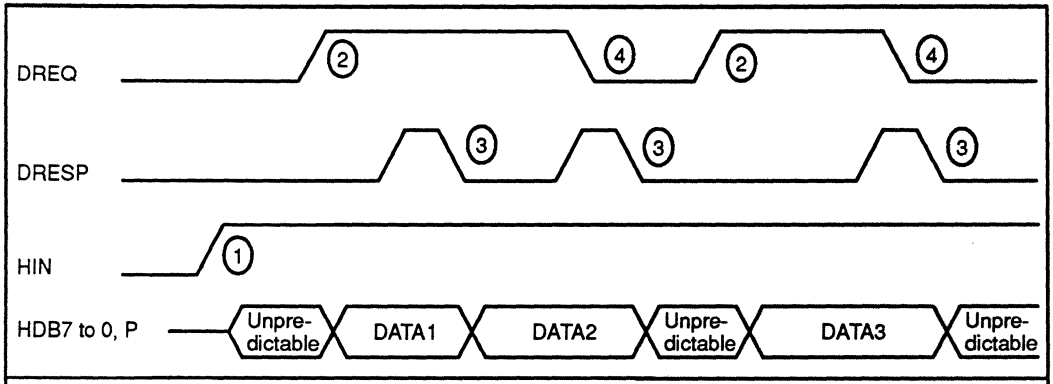


Figure 4–3. Example of Differential Type SCSI Driver/Receiver Circuit (continued)

External Data Buffer

To provide a buffer area for high-speed data transfer or a temporary storage for data to be transferred, the external data buffer memory can be connected to the SPC as shown in Figure 4–1. In this case, a buffer control circuit is required to control the timing of data transfer between the SPC and external buffer memory. To execute the transfer phase on SCSI, specify DMA Transfer Mode for the SPC. The SPC accesses the external buffer memory with the timing sequences shown in Figures 4–4 and 4–5. Data is transferred via the DMA data bus lines HDB7 to HDB0, HDBP. The transfer direction must be specified externally using the HIN signal. When requesting access to the external buffer memory, the SPC makes the DREQ signal active. In an input operation, the DREQ signal is sent out when the SPC internal data buffer register holds data received from SCSI. In an output operation, the DREQ signal is sent out when data corresponding to the byte count specified in the transfer byte counter is not all prefetched, and when the internal data buffer register has free byte locations available. The external buffer control circuit must return the DRESP signal in response to the DREQ signal on completion of transferring each byte. DRESP is a pulse signal whose trailing edge is used to indicate the end of transfer. The DREQ signal is held active as long as the above conditions exist in the SPC (this signal is not a transfer request signal for each byte). The access interface signals (DREQ, DRESP, HIN, HDB7 to HDB0, HDBP) to the external memory are asynchronous with an SPC clock signal supplied to the $\overline{CLK}$ pin.



Notes:

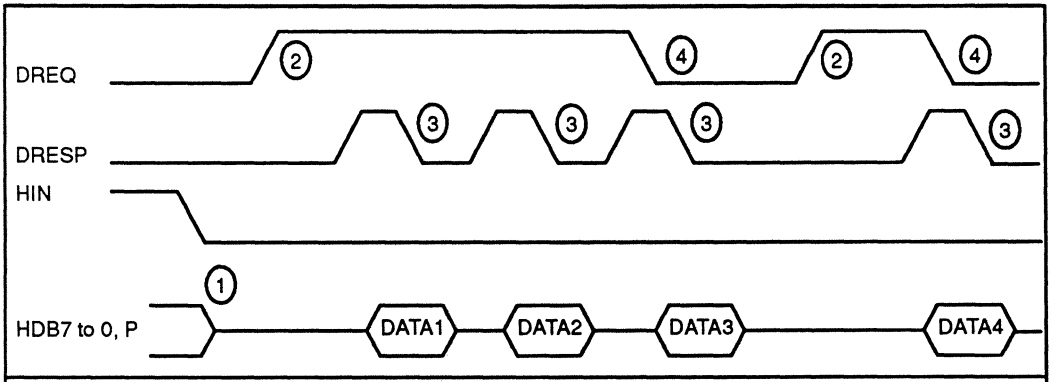
(1) HIN signal puts the DMA data bus (HDB7 to O, P) in an output mode. However, the value on the DMA data bus is unpredictable until data is loaded into the SPC internal data buffer register from SCSI.

(2) After receiving data from SCSI, the SPC issues the DREQ signal (transfer request) to the external buffer memory. At this time, valid data is placed on the DMA data bus.

(3) Data on the DMA data bus is held until the trailing edge of the DRESP signal.

(4) After the DRESP signal is received, valid data is unloaded from the SPC internal data buffer register. Then, the DREQ signal becomes inactive to stop transfer with the external buffer memory until the next data becomes available.

Figure 4–4. Transfer with External Data Buffer (Input operation)



Notes:

(1) As long as the HIN signal is low, the DMA data bus (HDB7 to O, P) is in an input mode.

(2) A transfer request (DREQ) signal is issued when the SPC internal data buffer register has free byte locations available for prefetching data.

(3) The external buffer control circuit puts data onto the DMA data bus and also sends the DRESP signal in response. On the trailing edge of the DRESP signal, the SPC loads data from the DMA bus into the internal data buffer register.

(4) When no free location is available in the SPC internal data buffer register, the DREQ signal becomes inactive. Then, transfer from the external buffer memory is stopped until the next data prefetch becomes available.

Figure 4–5. Transfer to External Data Buffer (output operation)

MB87030
MB87031
SCSI Protocol Controller (SPC)

GENERAL DESCRIPTION

The MB87030 and MB87031 SCSI Protocol Controller (SPC) are CMOS LSI circuits specifically designed to control a Small Computer Systems Interface (SCSI). In terms of features, functional operation, and electrical specifications, the two devices are identical. However, the MB87030 is housed in an 88-pin ceramic pin grid array package, whereas, the MB87031 is designed for surface mounting and is housed in a 100-pin plastic flat package.

To achieve optimum performance and interface flexibility, the SPC contains an 8-byte First In First Out (FIFO) data buffer register and a 24-bit transfer byte counter. Independent data busses for the CPU and the DMA controller plus separate input/output pins for all control signals greatly enhances performance. Data transfers can be executed in either the asynchronous or synchronous mode with a maximum offset of 8-bytes.

SCSI Compatibility

- Supports all of SCSI Specification (ANSI X3.131/1986)
- Serves as either INITIATOR or TARGET
- Both synchronous and asynchronous operation
- Software compatible with MB87033

Data Buses

- Independent busses for CPU and DMA transfer

Transfer Modes

- Asynchronous data transfers with programmable offset of up to 8 bytes
- Synchronous data transfers with programmable offset of up to 8 bytes

Data Transfer Speed

- Up to a maximum of 4 megabytes per-second (sustained).

Operating Modes

- DMA transfer
- Manual transfer
- Program transfer
- Diagnostic

Interface Connections

- Single-ended or differential options
- TTL compatible I/O

Clock Requirements

- 8 MHz clock with 33% to 66% duty cycle

Technology/Power Requirements

- Silicon-gate CMOS
- Single +5 V power supply

Available Packaging

- 88-pin ceramic repeated quad-in-line
- 100-pin plastic flat package

ABSOLUTE MAXIMUM RATINGS[1]

Rating	Symbol	Values		Unit
		Min.	Max.	
Supply Voltage	V_{DD}	V_{SS}[2] – .05	7.0	V
Input Voltage	V_I	V_{SS}[2] – .05	V_{DD} + .05	V
Output Voltage[2]	V_O	V_{SS}[2] – .05	V_{DD} + .05	V
Storage Temperature (Ceramic)	T_{STG}	–65	+150	°C
Temperature Under Bias (Ceramic)	T_{BIAS}	–40	+125	°C
Output Current[3]	I_{CS}	–40	+70	mA

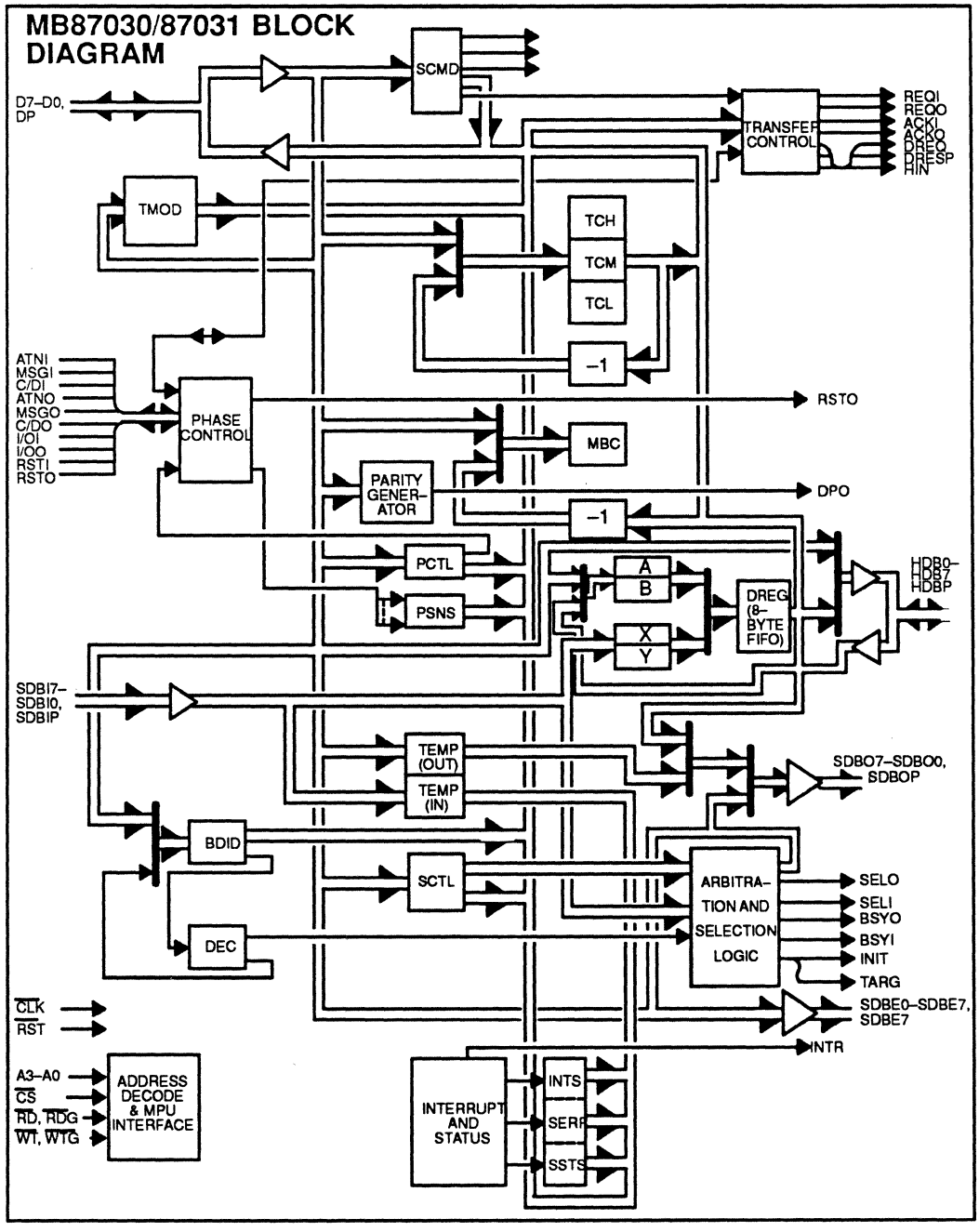
Notes: 1. Permanent device damage may occur if the above absolute maximum ratings are exceeded. Functional operation should be restricted to the conditions as detailed in the operational sections of this data sheet. Expsosure to absolute maximum rating conditions for extended periods may affect device reliability.

2. V_{SS} = oV.

3. Not more than one output may be shorted at a time for a maximum duration of one second.

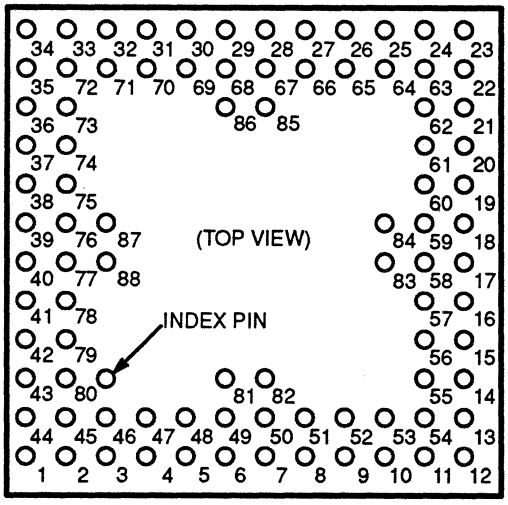
RECOMMENDED OPERATING CONDITIONS

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
Supply Voltage	V_{DD}	4.75	5.0	5.25	V
Input High Voltage	V_{IH}	2.2			V
Input Low Voltage	V_{IL}			0.8	V
Operating Temperature	T_A	0		70	°C


MB87030/87031 BLOCK DIAGRAM
SCMD
D7–D0, DP
TRANSFER CONTROL
REQI
REQO
ACKI
ACKO
DREQ
DRESP
HIN
TMOD
TCH
TCM
TCL
–1
ATNI
MSGI
C/DI
ATNO
MSGO
C/DO
I/OI
I/OO
RSTI
RSTO
PHASE CONTROL
RSTO
MBC
PARITY GENER-ATOR
DPO
–1
PCTL
PSNS
A
B
X
Y
DREG (8-BYTE FIFO)
HDB0–HDB7 HDBP
SDBI7–SDBI0, SDBIP
TEMP (OUT)
TEMP (IN)
SDBO7–SDBO0, SDBOP
BDID
SCTL
DEC
ARBITRA-TION AND SELECTION LOGIC
SELO
SELI
BSYO
BSYI
INIT
TARG
SDBE0–SDBE7, SDBE7
CLK
RST
INTR
A3–A0
CS
RD, RDG
WT, WTG
ADDRESS DECODE & MPU INTERFACE
INTERRUPT AND STATUS
INTS
SERR
SSTS

PIN ASSIGNMENTS

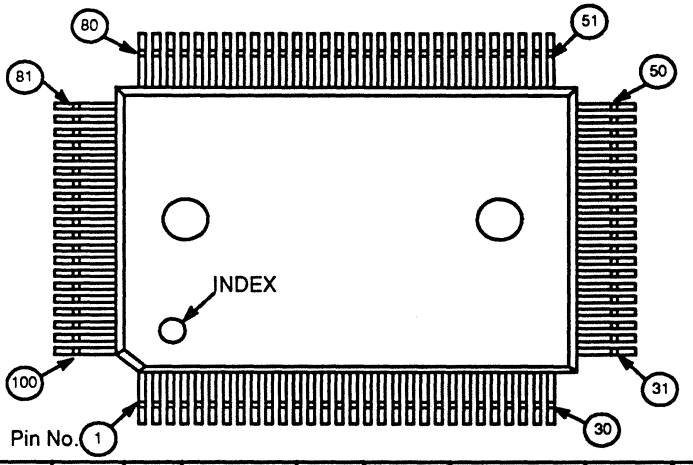
PGA–88C–A01



Pin No.	I/O	Designator	Pin No.	I/O	Designator	Pin No.	I/O	Designator	Pin No.	I/O	Designator
1	I	HIM	23	O	SDBOP	45	1	A1	67	O	SDBE5
2	I/O	HDBO0	24	O	SDBE7	46	I	A2	68	O	SDBE4
3	I/O	HDBO1	25	I	SDBI7	47	I	A3	69	I	SDBI4
4	I/O	HDBO2	26	O	SDBE6	48	I/O	D4	70	O	SDBO3
5	I/O	HDBO3	27	O	SDBO5	49	I/O	D5	71	I	SDBI2
6	I/O	HDBO4	28	I	SDBI5	50	I/O	D6	72	O	SDBO1
7	I/O	HDBO5	29	O	SDBO4	51	I/O	D7	73	O	SDBE0
8	I/O	HDBO6	30	O	SDBE3	52	I/O	DP	74	I	SDBI0
9	I/O	HDBO7	31	I	SDBI3	53	O	INTR	75	I	$\overline{RST}$
10	I/O	HDBOP	32	O	SDBO2	54	I	I/OI	76	O	DREQ
11	O	INIT	33	O	SDBE2	55	I	C/DI	77	I	$\overline{WT}$
12	O	TARG	34	I	SDBI1	56	I	SELI	78	I	$\overline{WTG}$
13	O	I/OO	35	O	SDBE1	57	I	MSG1	79	I/O	D2
14	O	C/DO	36	O	SDBO0	58	I	REQI	80	I/O	D3
15	O	SELO	37	I	$\overline{CS}$	59	I	RSTI	81	Power Supply	
16	O	MSGO	38	I	$\overline{CLK}$	60	I	ACKI	82	Power Supply	
17	O	REQO	39	I	$\overline{RD}$	61	I	BSYI	83	Power Supply	
18	O	RSTO	40	I	$\overline{RGD}$	62	I	ANTI	84	Power Supply	
19	O	ACKO	41	I	DRESP	63	I	SDBIP	85	Power Supply	
20	O	BSYO	42	I/O	D0	64	O	SDBO7	86	Power Supply	
21	O	ATNO	43	I/O	D1	65	O	SDBO6	87	Power Supply	
22	O	SDBEP	44	I	A0	66	I	SDBI6	88	Power Supply	

PIN ASSIGNMENTS (Continued)

FPT–100P–M01



Pin No.	I/O	Designator	Pin No.	I/O	Designator	Pin No.	I/O	Designator	Pin No.	I/O	Designator
1	I	DRESP	26	I/O	D2	51	O	TARG	76	O	SDBE5
2	O	DREQ	27	I/O	D3	52	O	INIT	77	O	SDBO5
3	—	V_{DD}	28	—	V_{DD}	53	—	V_{DD}	78	—	V_{DD}
4	—	V_{SS}	29	—	V_{SS}	54	—	V_{SS}	79	—	V_{SS}
5	I	HIN	30	I/O	D4	55	I	ACKI	80	—	N/C
6	I/O	HDB0	31	I/O	D5	56	O	ACKO	81	I	SDBI4
7	I/O	HDB1	32	I/O	D6	57	—	N/C	82	O	SDBE4
8	I/O	HDB2	33	I/O	D7	58	—	N/C	83	O	SDBO4
9	I/O	HDB3	34	I/O	DP	59	I	BSYI	84	I	SDBI3
10	I/O	HDB4	35	I	A0	60	O	BSYO	85	O	SDBE3
11	I/O	HDB5	36	I	A1	61	I	ATNI	86	O	SDBO3
12	I/O	HDB6	37	I	A2	62	O	ATNO	87	I	SDBI2
13	I/O	HDB7	38	I	A3	63	I	RSTI	88	O	SDBE2
14	I/O	HDBP	39	I	$\overline{RST}$	64	O	RSTO	89	O	SDBO2
15	—	V_{SS}	40	—	V_{SS}	65	—	V_{SS}	90	—	V_{SS}
16	I	$\overline{CLK}$	41	I	REQI	66	I	SDBIP	91	I	SDBE1
17	I	$\overline{CS}$	42	O	REQO	67	O	SDBEP	92	O	SDBE1
18	I	$\overline{WT}$	43	I	I/OI	68	O	SDBOP	93	O	SDBO1
19	I	$\overline{WTG}$	44	O	I/OO	69	I	SDBI7	94	I	SDBI0
20	I	$\overline{RD}$	45	I	C/DI	70	O	SDBE7	95	O	SDBE0
21	I	$\overline{RDG}$	46	O	C/DO	71	O	SDBO7	96	O	SDBO0
22	O	INTR	47	I	SELI	72	I	SDBI6	97	—	N/C
23	—	N/C	48	O	SELO	73	O	SDBE6	98	—	N/C
24	I/O	D0	49	I	MSG1	74	O	SDBO6	99	—	N/C
25	I/O	D1	50	O	MSG0	75	I	SDBI5	100	—	N/C

PIN DESCRIPTIONS

<table>
<tr><th colspan="2">Pin No.</th><th rowspan="2">Designator</th><th rowspan="2">Function</th></tr>
<tr><th>MB87030</th><th>MB87031</th></tr>
<tr><td>1</td><td>5</td><td>HN</td><td>Indicates direction of transmission along data bus lines HDB0–HDB7 and HDBP in the DMA transfer mode. to be executed, Direction of transmission must be properly coordinated with internal operation of the SPC.
When HIN is low, the data bus lines are placed in the high-impedence state (input mode). When HIN is high, all bus lines are switched to the output mode.</td></tr>
<tr><td>2–9
10</td><td>6–13
13</td><td>HDB0–HDB7
HDBP</td><td>3-state bidirectional data bus for transferring data to or from the external buffer memory in the DMA mode. As shown below, the direction of data transmission dlepends on the HIN input signal.
<table>
<tr><th>HIN</th><th>HDBn</th><th>Operation</th></tr>
<tr><td>L</td><td>Input Mode</td><td>Output</td></tr>
<tr><td>H</td><td>Output Mode</td><td>Input</td></tr>
</table></td></tr>
<tr><td>11
12</td><td>51
51</td><td>INIT
TARG</td><td>These two signals indicate operating state of SPC; they are also available as control signals for the SCSI driver/receiver circuits.
<table>
<tr><th>Initiator</th><th>Target</th><th>Status</th></tr>
<tr><td>L</td><td>L</td><td>SPC is not connected to SCSI.</td></tr>
<tr><td>L</td><td>H</td><td>SPC is executing reselection phase or is operating as a target.</td></tr>
<tr><td>H</td><td>L</td><td>SPC is executing selection phase or is operating as an initiator.</td></tr>
</table></td></tr>
<tr><td>13
14
15
16
17
18
19
20
21</td><td>44
46
48
50
42
64
56
60
62</td><td>I/O0
C/D0
SEL0
MSG0
REQ0
RST0
ACK0
BSY0
ATN0</td><td>Used to output SCSI control signals. REQ0, MSG0, C/D0, and I/O0 are active high only when the SPC serves as a target. ACK0 and ATN0 are active high only when the SPC services as an initiator.</td></tr>
<tr><td>22
24
26
67
68
30

33
35
73</td><td>67
70
73
76
82
85

88
92
95</td><td>SDBEP
SDBE7
SDBE6
SDBE5
SDBE4
SDBE3

SDBE2
SDBE1
SDBE0</td><td>Drive enable signals (corresponding to respective bit positions) when a 3-state buffer is used for the SCSI data bus. SDBE–7SDBE0 and SDBEP correspond to SDBO7–SDBO0 and SDBOP, respectively. Relationships with respect to the SCSI bus are shown below.
<table>
<tr><th rowspan="2">SCSI BUS STATUS</th><th colspan="2">SDBDOn</th><th colspan="2">SDBEn</th></tr>
<tr><th>ID*1</th><th>ID*1</th><th>ID*1</th><th>ID*1</th></tr>
<tr><td>Bus Free</td><td>L</td><td>L</td><td>L</td><td>L</td></tr>
<tr><td>Arbitration</td><td>H</td><td>L</td><td>H</td><td>L</td></tr>
<tr><td>Selection/Reselection</td><td>D*2</td><td>D*2</td><td>H</td><td>H</td></tr>
<tr><td>Information Transfer</td><td></td><td></td><td></td><td></td></tr>
<tr><td>SPC → SCSI</td><td>D*2</td><td>D*2</td><td>H</td><td>H</td></tr>
<tr><td>SCSI → SPC</td><td>L</td><td>L</td><td>L</td><td>L</td></tr>
</table>
Notes: 1. *ID indicates bit positions corresponding to the SCSI bus device ID; $\overline{ID}$ indicates the other bit position.
2. *D indicates transfer of valid information.</td></tr>
</table>

PIN DESCRIPTIONS (Continued)

Pin No.		Designator	Function
MB87031	MB87031		
25 66 28 69 31 71 34 74 63	69 72 75 81 84 87 91 94 66	SDBI7 SDBI6 SDBI5 SDBI4 SDBI3 SDBI2 SDBI1 SDBI0 SDBIP	Inputs for the SCSI data bus. Most significant bit (MSB) is SDBI7; least significant bit (LSB) is SDBI0. SDBIP is an odd parity bit; parity checking for the SCSI data bus is programmable.
64 65 27 29 70 32 72 36 23	71 74 77 83 85 89 93 96 68	SDBO7 SDBO6 SDBO5 SDBO4 SDBO3 SDBO2 SDBO1 SDBO0 SDBOP	Outputs for the SCSI data bus. (MSB is SDBO7; SDBO0. SDBOP is an odd parity bit. If the bus driver is an open collector device, these signals should be applied directly to the driver circuit. If the new bus driver is a 3-state device, these signals are used as data and SDBO7–SDBO0 and SDBOP are used as drive-enable signals.
37	17	$\overline{CS}$	Selection enable signal for accessing an internal register in SPC. When $\overline{CS}$ is active, input/output signals $\overline{RD}$, $\overline{RDG}$, $\overline{WT}$, $\overline{WTG}$, DP, A0–A3, and D0–D7 are active.
38	16	$\overline{CLK}$	Input clock for controlling internal operation and data transfer speed of SPC.
39 40	20 21	$\overline{RD}$ $\overline{RDG}$	Input strobes used for reading out contents of internal register; strobes are effective only when $\overline{CS}$ is active low. When $\overline{RDG}$ is active low, the contents of an internal register selected buy address inputs A0–A3 are placed on data bus lines D0–D7 and DP. For a data transfer cycle in the program transfer mode, the trailing edge of $\overline{RD}$ is used as a timing signal to indicate the end of data read.
41	1	DRESP	During a data transfer cycle in the DMA mode, DRESP is a response signal to the data transfer request signal DREQ. The DRESP pin must be refreshed with an applied pulse after each byte of data is transferred to output operations, the falling edge of DRESP is used for sampling data on HDB0-HDB7 and HDBP bus lines; in input operations, the SPC holds data to be transferred onto HDB0-HDB7 and HIDBP until the falling edge of DRESP occurs.

PIN DESCRIPTIONS (Continued)

Pin No.		Designator	Function
MB87031	MB87031		
51 50 49 48 80 79 43 42 52	33 32 31 30 27 26 25 24 34	D7 D6 D5 D4 D3 D2 D1 D0 DP	Used for writing or reading data into or from an internal register in SPC; these bus liens are 3-state and bidirectional. The (MSB) is D7; the LSB is D0. DP is an odd parity bit. When the $\overline{CS}$ and $\overline{RDG}$ inputs are active Low, contents of the internal register are output to the data bus (read operation). In operations other than read, these bus lines are kept in a hgh-impedence state.
44–47	35–38	80–83	Address input signals for selecting an internal register in the SPC. The MSB is A3; the LSB is A0. When CS is active low, read/write is enabled and an internal register is selected by these address inputs via data bus lines D0-D7 and DP.
53	22	INTR	Requests an interrupt to indicate completion of an SPC internal operation or the occurrence of an error. Interrupt masking is allowed except for an interrupt caused by the RSTI input (reset condition of SCSI). When an interrupt is permitted, the INTR signal remains active until the interrupt is cleared.
56 61 58 60 57 55 54 62 59	27 59 41 55 49 45 43 61 63	SELI BSYI REQI ACKI MSGI C/DI I/OI ATNI RSTI	Used for receiving SCSI control signals; outputs of the SCSI receiver can be directly connected. (Waveform distortion or any other disturbance should not occur in the REQI and ACKI signals which are used as timing control signals for sequencing data transfers.)

PIN DESCRIPTIONS (Continued)

Pin No.		Designator	Function
MB87031	MB87031		
76	2	DREQ	When executing a data transfer cycle in the DMA mode, DREQ is used to indicate a request for data transfer between the SPC and external buffer memory. In the DMA mode, routing of data is as shown below. Output Operations: From external buffer memory to HDB0-HDB7/HDBBP to SPC internal data buffer register (8 Bytes) to SDBO0-SDBO7/SDBOP to SCSI. Input Operations: From SCSI to SDBI9-SDBI7/SDBIP to SPC internal data buffer register (8 bytes) to HDB0-HDB7/DHDBP to external buffer memory. In an output operation, DREQ becomes active to request a data transfer from the external buffer memory when the SPC internal data buffer register has free space available. In an input operation, DREQ becomes active to request a data transfer to the external buffer memory when the SPC internal buffer memory contains valid data.
77	18	WT	Input strobe used for writing data into an SPC internal register; this signal is asserted only when CS is active Low. On the trailing edge of WT, data placed on data bus lines D0-D7/DP is loaded into the internal register selected by address inputs A0-A3, except when all address lines are High (A0-A3 = H). For a data transfer cycle in the program transfer mode, the trailing edge of WT is used as a timing signal to indicate a data-ready state.
78	19	WTG	When WTG is active low, data appearing on data bus lines D0-D7/DP is output to HDB0-HDB7/HDBP if the following input conditions are satisfied. CS = L A0-A3 = H HIN = H
81, 84 85, 88	4, 15, 29 40, 54, 65, 79,90	V_{SS}	Power supply ground.
82. 83 86, 87	3, 28 53, 78	V_{DD}	+5V Power Supply
—	23, 57, 58, 80, 97, 98, 99, 100	—	Not used.

ADDRESSING OF INTERNAL REGISTERS

Both the MB87030 and the MB87031 contain16 byte-wide registers that are externally accessible. These registers are used to control internal operations of the SPC and also to indicate processing/result status. A unique address, identified by address bits A3-A0, is assigned to each of the sixteen registers. These addresses are defined in Table 1. (Note: The phase sense (PSNS) and SPC diagnostic (SDGC) registers have the same hexadecimal address; however, depending upon whether a read or write command is executed, the registers provide two separate functions.)

Table 1. Internal Register Addressing

Register	Mnemonic	Operation	Chip Select ($\overline{CS}$)	Address Bits A3	A2	A1	A0
Bus Device ID	BDID	R	0	0	0	0	0
		W					
SPC Control	SCTL	R	0	0	0	0	1
		W					
Command	SCMD	R	0	0	0	1	0
		W					
Transfer Mode	TMOD	R	0	0	0	1	1
		W					
Interrupt Sense	INTS	R	0	0	1	0	0
Reset Interrupt		W					
Phase Sense	PSNS	R	0	0	1	0	1
SPC Diagnostic Control	SDGC	W					
SPC Status	SSTS	R	0	0	1	1	0
—		W					
SPC Error Status	SERR	R	0	0	1	1	1
—		W					
Phase Control	PCTL	R	0	1	0	0	0
		W					
Modified Byte Counter	MBC	R	0	1	0	0	1
—		W					
Data Register	DREG	R	0	1	0	1	0
		W					
Temporary Register	TEMP	R	0	1	0	1	1
		W					

Table 1. Internal Register Addressing (Continued)

Register	Mnemonic	Operation	Chip Select ($\overline{CS}$)	Address Bits A3	A2	A1	A0
Transfer Counter High	TCH	R	0	1	1	0	0
		W					
Transfer Counter Middle	TCM	R	0	1	1	0	1
		W					
Transfer Counter Low	TCL	R	0	1	1	1	0
		W					
External Buffer	EXBF	R	0	1	1	1	1
		W					

BIT ASSIGNMENTS

Table 2 lists the bit assignments for the seventeen internal registers defined in Table 1. During read/write access of an internal register, the following rules are invoked:

- Internal registers include only those registers identified in Table 1.
- A write command to a read–only register is ignored.
- For write operations, all bit positions with a "—" (blank) designator can be written as a "0" or as a "1".
- All bit positions with an assigned "0" are always read as a zero (0).

Table 2. Bit Assignments For Internal Registers

HEX Address	Register and Mnemonic	R/W Operation	7 (MSB)	6	5	4	3	2	1	0 (LSB)	Parity
0	Bus Device ID (BDID)	R	#7	#6	#5	#4	#3	#2	#1	#0	0
		W	SCSI Bus Device ID					ID4	ID2	ID1	—
1	SPC Control (SCTL)	R	Reset & Disable	Control Reset	Diag Mode	ARBIT Enable	Parity Enable	Select Enable	Reselect Enable	INT Enable	P
		W									
2	Command (SCMD)	R	Command Code			RST Out	Intercept Xfer	Transfer PRG Xfer	Modifier 0	Term Mode	P
		W									
3	Transfer Mode (TMOD) Command (SCMD)	R	Sync. Xfer	Max. Transfer Offset			Min. Transfer Period		0	0	P
		W		4	2	1	2	1			
4	Interrupt Sense (INTS)	R	Selected	Reselected	Disconnect	Command Complete	Service Required	Time Out	SPC Hard Error	Reset Condition	P
		W	Reset Interrupt								—
5	Phase Sense (PSNS)	R	REQ	ACK	ATN	SEL	BSY	MSG	C/D	I/O	P
		W	—								
	SPC Diag Control (SDGC)	R	—								
		W	Diag. REQ	Diag. ACK	XFER Enable		Diag. BSY	Diag. MSG	Diag. C/D	Diag. I/O	—
6	SPC Status (SSTS)	R	Connected		SPC	XFER In Progress	SCSI	TC=0	DREG	Status	P
			INIT	TARG	BSY		RST		Full	Empty	
		W	—								—
7	SPC Error Status (SERR)	R	Data Error		0	0	TC	Phase	Short	Offset	P
			SCSI	SPC	XFER Out		Parity Error	Error	Period	Error	
		W	—								—

Table 2. Bit Assignments For Internal Registers (Continued)

HEX Address	Register and Mnemonic	R/W Operation	7 (MSB)	6	5	4	3	2	1	0 (LSB)	Parity
8	Phase Control (PCTL)	R	Bus Free Interrupt Enable	0				Transfer Phase			P
		W						MSG Out	C/D Out	I/O Out	
9	Modified Byte Counter (MBC)	R	0				Bit3	MBC Bit2	Bit1	Bit0	
		W	—							—	
A	Data Register (DREG)	R	Internal Data Register (8 Byte FIFO)								P
		W	Bit7	6	5	4	3	2	1	0	
B	Temporary Register (TEMP)	R	Temporary Data (Input: From SCSI)								P
			Bit7	6	5	4	3	2	1	0	
		W	Temporary Data (Output: to SCSI)								P
			Bit7	6	5	4	3	2	1	0	
C	Transfer Counter High (TCH)	R	Transfer Counter High (MSB)								P
		W	Bit23	22	21	20	19	18	17	16	
D	Transfer Counter Mid. (TCM)	R	Transfer Counter High (2nd Byte)								P
		W	Bit15	14	13	12	11	10	9	8	
E	Transfer Counter Low (TCL)	R	Transfer Counter High (LSB)								P
		W	Bit7	6	5	4	3	2	1	0	
F	External Buffer (EXBF)	R	External Buffer								P
		W	Bit7	6	5	4	3	2	1	0	

DC CHARACTERISTICS

(Recommended operating conditions unless otherwise specified)

Parameter	Designator	Condition	Values			Unit
			Min	Typ	Max	
Power Supply Current	t_{DS}	Steady State[1]			100	mA
Power Dissipation	P_D			300		mW
Output High Voltage	V_{OH}	$I_{OH} = -0.4mA$	4.2		V_{DD}	V
Output Low Voltage	V_{OL}	$I_{OL} = 3.2mA$	V_{SS}		0.4	V
Input High Voltage	VIH		2.2			V
Input Low Voltage	V_{IL}				0.8	V
Input Leakage Current	I_{LI}	$V_I = 0 -$	−10		10	mA
Input Leakage Current	I_{LZ}	3-state $V_I = 0 - V_{DD}$			10	mA

Note:

1. $V_{IH} = V_{DD} = V_{IL} = V_{SS}$

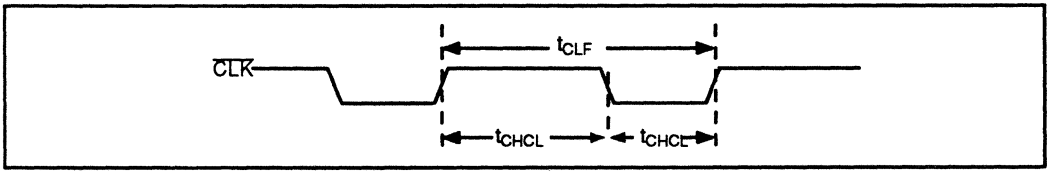
AC CHARACTERISTICS

(Recommended operating conditions unless otherwise noted)

MPU INTERFACE

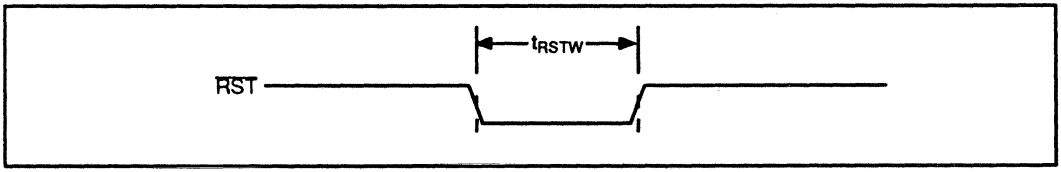
Clock Signal

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
Clock Cycle	t_{CLF}	125		200	ns
Clock High	t_{CHCL}	50			ns
Clock Pulse Width (Low)	t_{CLCH}	40			ns



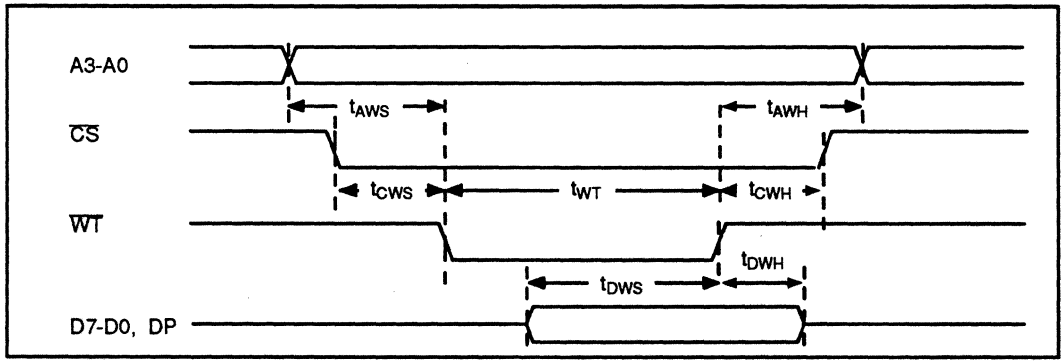
Reset Signal

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
Reset Pulse Width	t_{RSTW}	50			ns



Write Cycle (Registers other than EXBF)

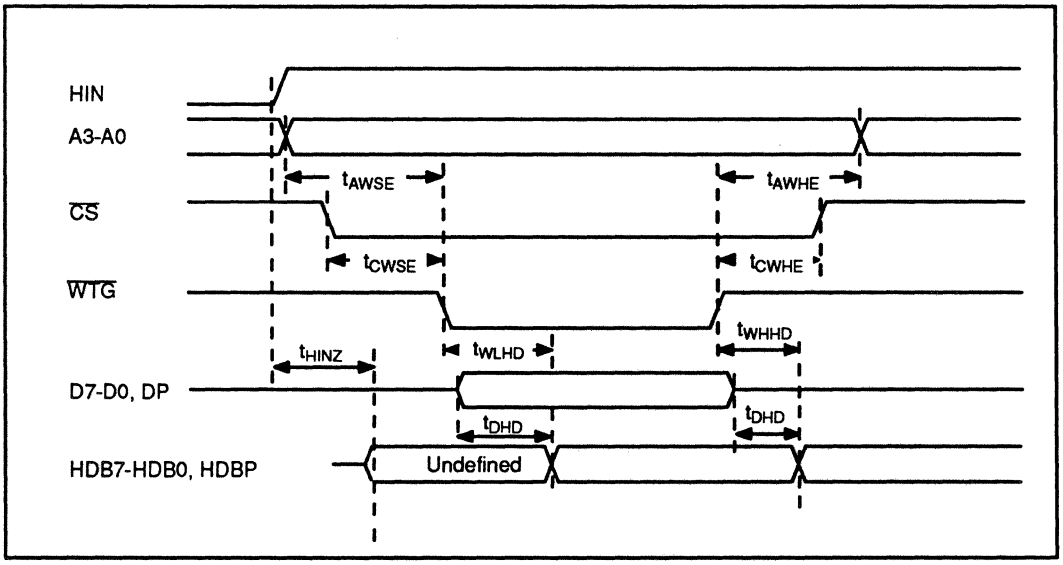
Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
Address Setup	t_{AWS}	40			ns
Address Hold	t_{AWH}	5			ns
$\overline{CS}$ Setup	t_{CWS}	25			ns
$\overline{CS}$ Hold	t_{CWH}	10			ns
Data Bus Setup	t_{DWS}	25			ns
Data Bus Hold	t_{DWH}	20			ns
$\overline{WT}$ Pulse Width	t_{WT}	50			ns



AC CHARACTERISTICS (Continued)

Write Cycle (EXBF Registers)

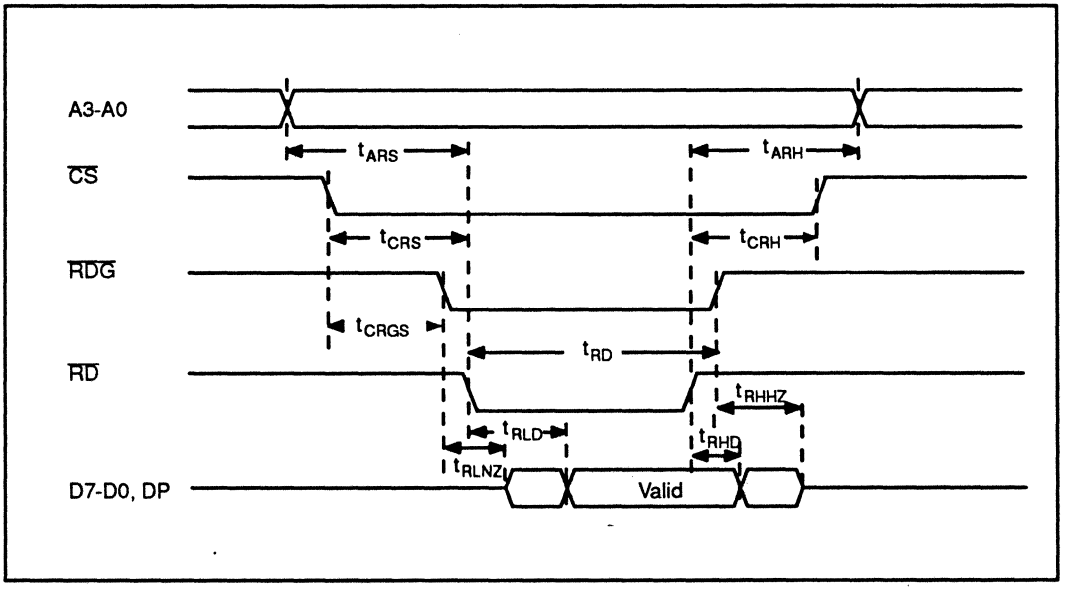
Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
Address Setup	t_{AWSE}	40			ns
Address Hold	t_{AWHE}	5			ns
$\overline{CS}$ Setup	t_{CWSE}	25			ns
$\overline{CS}$ Hold	t_{CWHE}	10			ns
$\overline{WTG}$ Low to DMA Data Bus HDB7-HDB0, HDBP	t_{WLHD}		40	60	ns
$\overline{WTG}$ High to DMA Data Bus HDB7-HDB0, HDBP	t_{WHHD}	10	30		ns
MPU Data Bus (D7-D0, DP) DMA Data Bus (HDB7-HDB0, HDBP)	t_{DHD}	5	25	50	ns
HIN High to DMA Data Bus (HDB7–HDB0, HDBP)	t_{HINZ}	10		40	ns



AC CHARACTERISTICS (Continued)

Read Cycle (Registers other than EXBF)

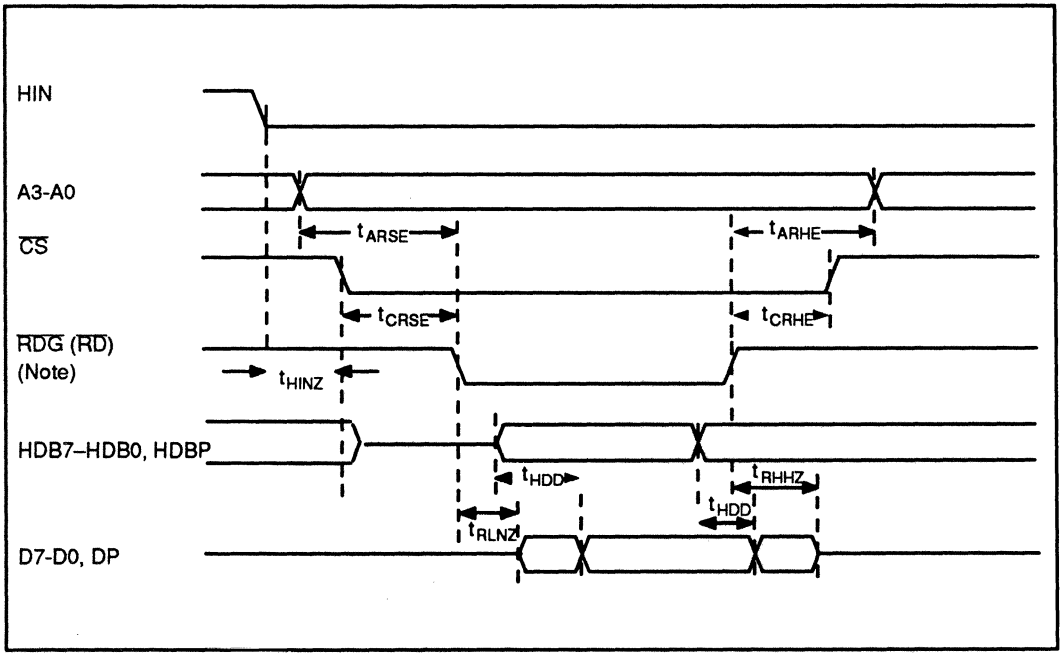
Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
Address Setup	t_{ARS}	40			ns
Address Hold	t_{ARH}	5			ns
$\overline{CS}$ Setup (RD)	t_{CRS}	10			ns
$\overline{CS}$ Hold	t_{CRH}	5			ns
$\overline{RD}$ Pulse Width	t_{RD}	50			ns
$\overline{RDG}$ Low to Data Output	t_{RLNZ}	10		45	ns
$\overline{RDG}$ High to D7-D0, DP High Z	t_{RHHZ}			40	ns
$\overline{RD}$ Low to Data Establish	t_{RLD}			85	ns
$\overline{RD}$ High to Data Hold	t_{RHD}	10			ns
CS Setup ($\overline{RDG}$)	t_{CRGS}	5			ns



AC CHARACTERISTICS (Continued)

Read Cycle (EXBF Register)

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
Address Setup	t_{ARSE}	40			ns
Address Hold	t_{ARHE}	5			ns
$\overline{CS}$ Setup	t_{CRSE}	10			ns
$\overline{CS}$ Hold	t_{CRHE}	10			ns
$\overline{RDG}$ Low to Data Output	t_{RLNZ}	10		45	ns
$\overline{RDG}$ High to D7-D0, DP High Z	t_{RHHZ}			40	ns
DMA Data Bus (HDB7-HDB0 HDBP) to MPU Data Bus (D7-D0, DP)	t_{HDD}	5		50	ns
HIN Low to HDB7–HDB0, HDBP High	t_{HINZ}			40	



Note: 1. These two signals may be applied simultaneously.

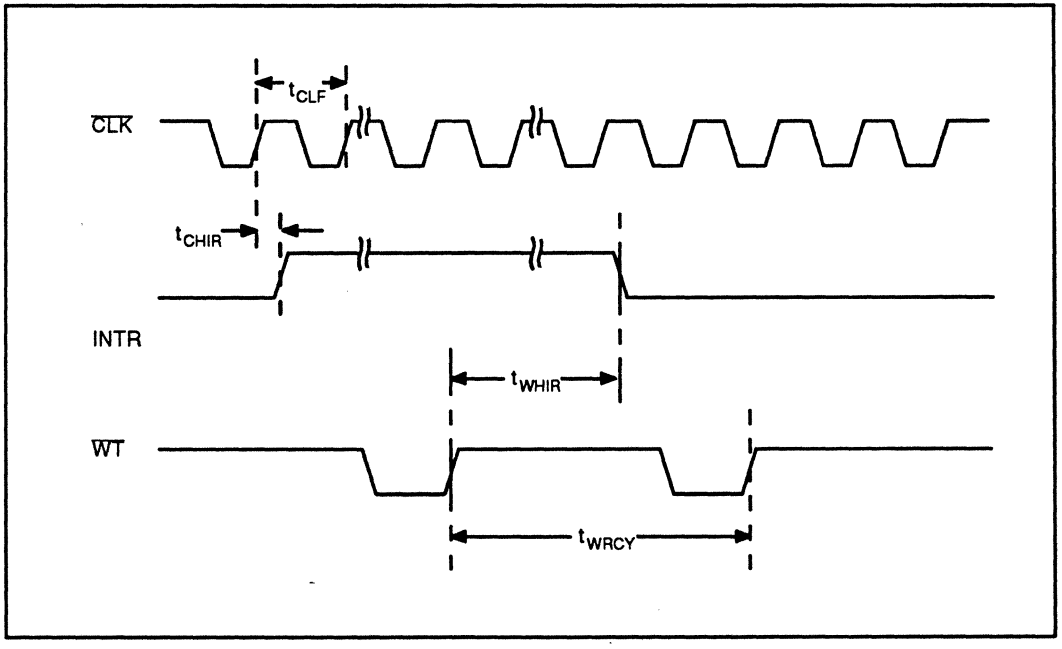
AC CHARACTERISTICS (Continued)

Interrupt

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
$\overline{CLK}$ High to INTR High	t_{CHIR}	5		55	ns
$\overline{WT}$ High to INTR Low (Interrupt Reset) Interrupt Reset Cycle Time[2]	t_{WHIR}	$t_{CLF}1 + 10$		$2t_{CLF} + 80$	ns ns
	t_{WRCY}	$4t_{CLF}$			

Notes:

1. Refer to "Clock Signal" timing for definition of t_{CLF}.
2. Cycle time for $\overline{WT}$ when interrupt is continuous.



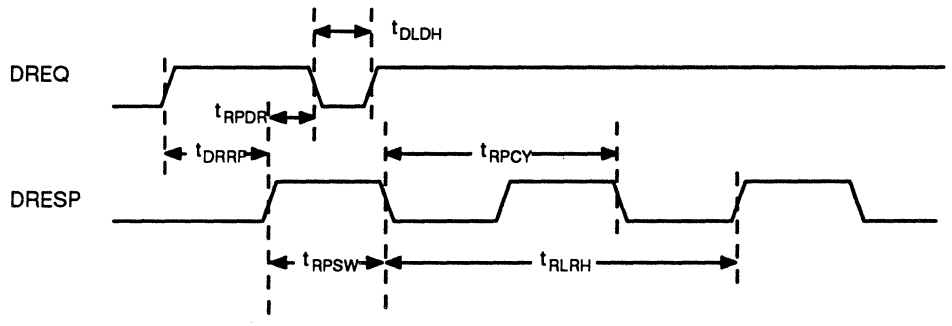
AC CHARACTERISTICS (Continued)

DMA Interface

Access Timing

Parameter	Designator	Values			Unit
		Min	Typ	Max	
DREQ High to DRESP High	t_{DRRP}	t_{CLF} (Note)			ns
DRESP High to DREQ Low	t_{RPDR}	10	45	80	ns
DREQ Low to DREQ High	t_{DLDH}	0			ns
DRESP Pulse Width	t_{RPSW}	50			ns
DRESP Cycle Time (1)	t_{RPCY}	$2t_{CLF}$			ns
DRESP Cycle Time (2)	t_{RLRH}	$3t_{CLF}$			ns

Note:
1. Refer to "Clock Signal" timing for definition of t_{CLF}

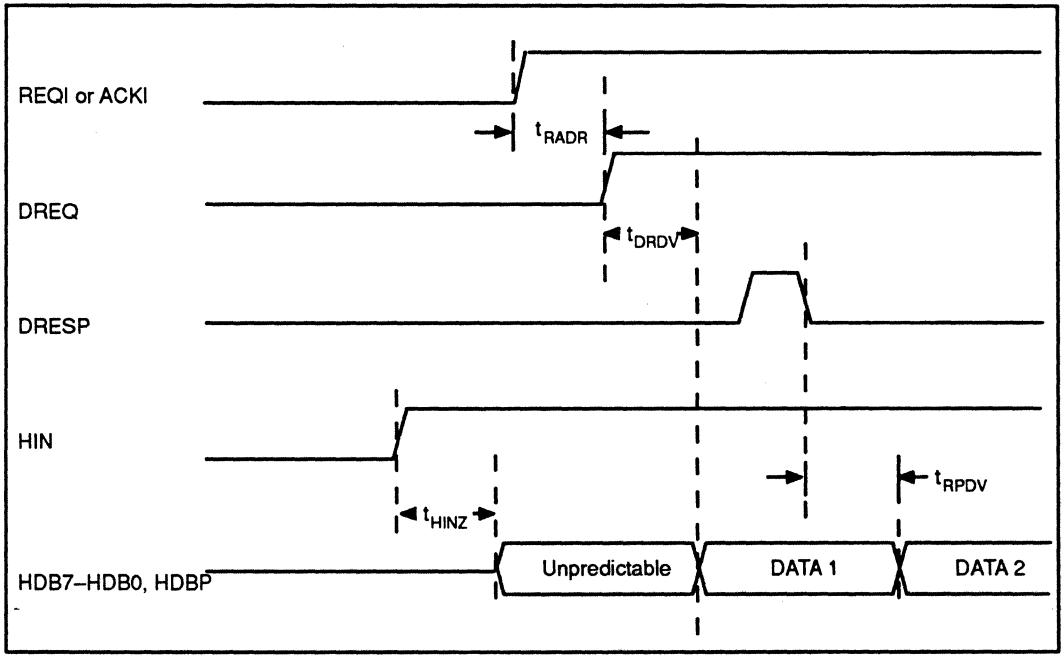


AC CHARACTERISTICS (Continued)

Input Operation (SPC to External Data Buffer)

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
HIN High to HDB7-HDB0, HDBP Data Output	t_{HINZ}	10		40	ns
DREQ High to Data Establish	t_{DRDV}			60	ns
DRESP Low to Data Change	t_{RPDV}	15	15	90	ns
REQI or ACKI High to DREQ High→	t_{RADR}		55	$3t_{CLF}$[2] + 70	ns

Notes:
1. When SPC receives REQ (Initiator) or ACK (Target) with an empty FIFO during DMA (hardware) transfer.
2. Refer to "Clock Signal" timing for definition of t_{CLF}



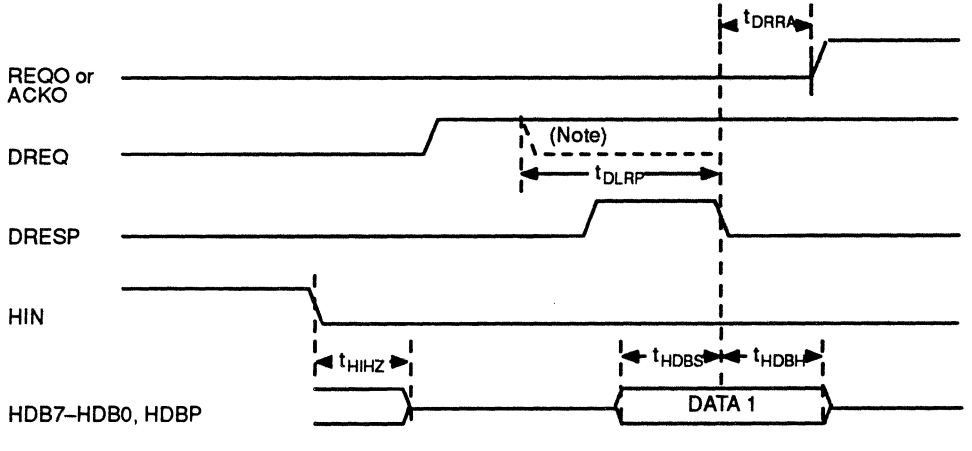
AC CHARACTERISTICS (Continued)

Output Operation (External Data Buffer to SPC)

Parameter	Designator	Values			Unit
		Min	Typ	Max	
HIN Low to HDB7-HDB0, HDBP (High Z)	t_{HIHZ}			40	ns
Data Bus Setup	t_{HDBS}	20			ns
Data Bus Hold	t_{HDBH}	20			ns
DREQ Low to DRESP (Note)	t_{DLRP}			$5t_{CLF}$[1]	ns
DRESP Low to REQO or ACKO High[2]	t_{DRRA}			$4t_{CLF}$[2] + 115	ns

Notes:

1. Refer to "Clock Signal" timing for definition of t_{CLF}.
2. The indicated timing is invoked if SPC receives DRESP when the internal data buffer is empty during a DMA (hardware) transfer. The timing parameter is waived for ACKO when the last byte is transferred with the SPC serving as an initiator.



Note: In the following cases,the DREQ signal becomes inactive asynchronously with the DRESP signal to stop refetching data during an output operation:

- The Transfer Pause command is issued when the SPC is serving as a TARGET
- The Transfer Phase is changed when the SPC is serving as an INITIATOR

In these cases, the last DRESP signal response must not exceed t_{DLRP}

AC CHARACTERISTICS (Continued)

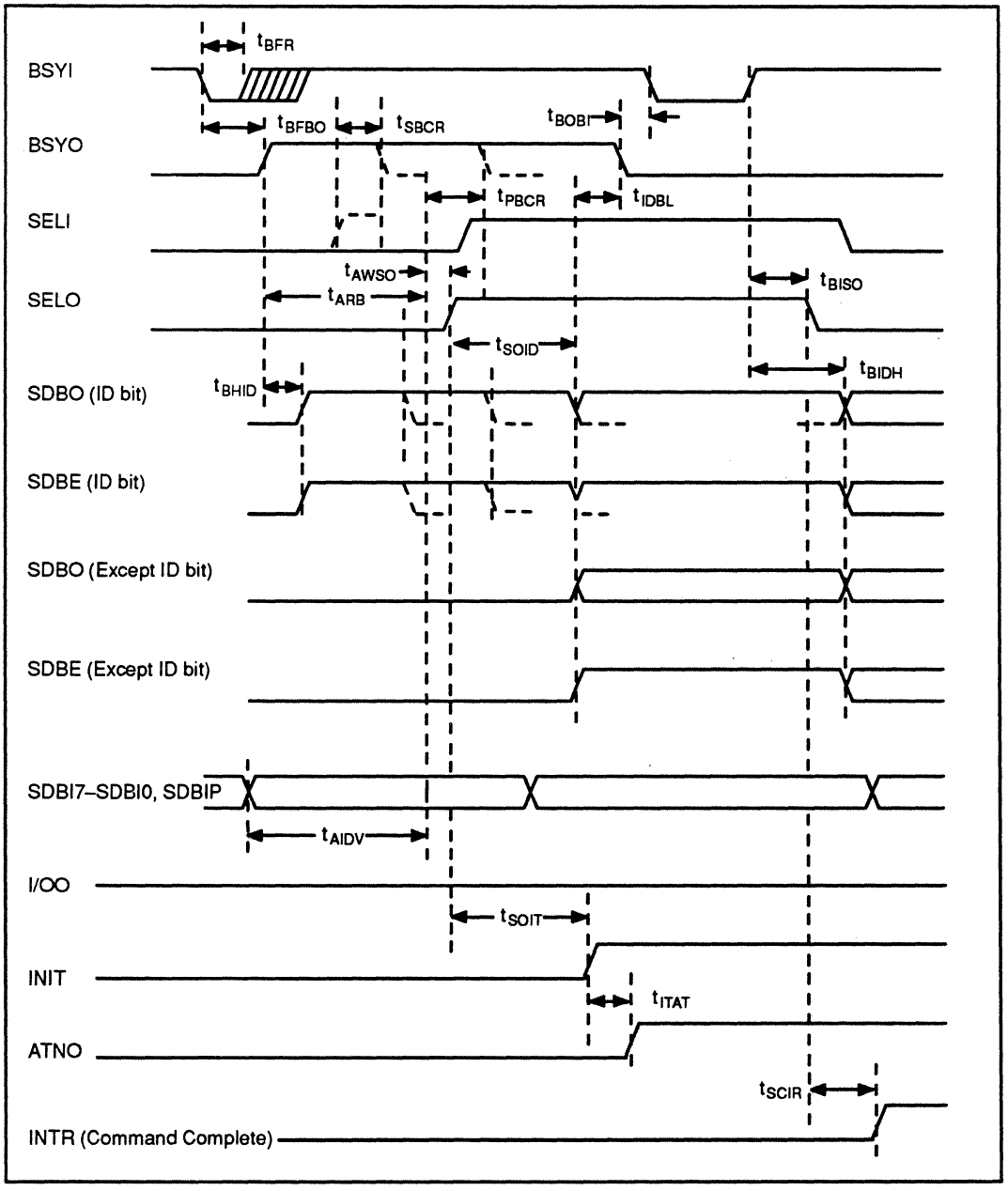
SCSI Interface (Selection Phase)

INITIATOR with Arbitration

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
Bus Free Time	t_{BFR}	$4t_{CLF}$+ 50			ns
BSYI Low to BSYO High (Start of arbitration)	t_{BFBO}	(6 + n) x t_{CLF} + 5		(7 + n) x t_{CLF} + 65	ns
BSYO High to ID Bit High	t_{BHID}	CLθ	20	55	ns
BSYO High to Prioritize	t_{ARB}	$32t_{CLF}$− 40			ns
Data Bus Valid (High Priority Bit) to Prioritize	t_{AIDV}	70			ns
Data Bus Valid (Low Priority Bit) to Prioritize		5			ns
Bus Usage Permission Granted to SELO High	t_{AWSO}	0		45	ns
SELO High to Data Bus (ID) Send	t_{SOID}	$11t_{CLF}$− 30	$11t_{CLF}$+15	$11t_{CLF}$+45	ns
SELO High to INIT High	t_{SOIT}	$11t_{CLF}$− 30	$11t_{CLF}$− 10	$11t_{CLF}$+ 40	ns
INIT High to ATNO High	t_{ITAT}		5	5	ns
Data Bus (ID) Send to BSYO Low	t_{IDBL}	$2t_{CLF}$− 50	$2t_{CLF}$− 10	$2t_{CLF}$+25	ns
BSYO Low to BSYI Low	t_{BOBI}	0		t_{CLF}	ns
BSYI High to SELO Low	t_{BISO}	$2t_{CLF}$+ 5			ns
BSYI High to Data Bus (ID) Hold	t_{BIDH}	$2t_{CLF}$+ 5			ns
SELO Low to INTR High	t_{SCIR}		0	35	ns
SELI High to BSYO, ID Bit Low	t_{SBCR}			$3t_{CLF}$+ 115	ns
Prioritize to BSYO, ID Bit Low	t_{PBCR}			125	ns

Notes:
1. Refer to "Clock Signal" timing for definition of t_{CLF}.
2. n = TCL register set value

AC CHARACTERISTICS (Continued)

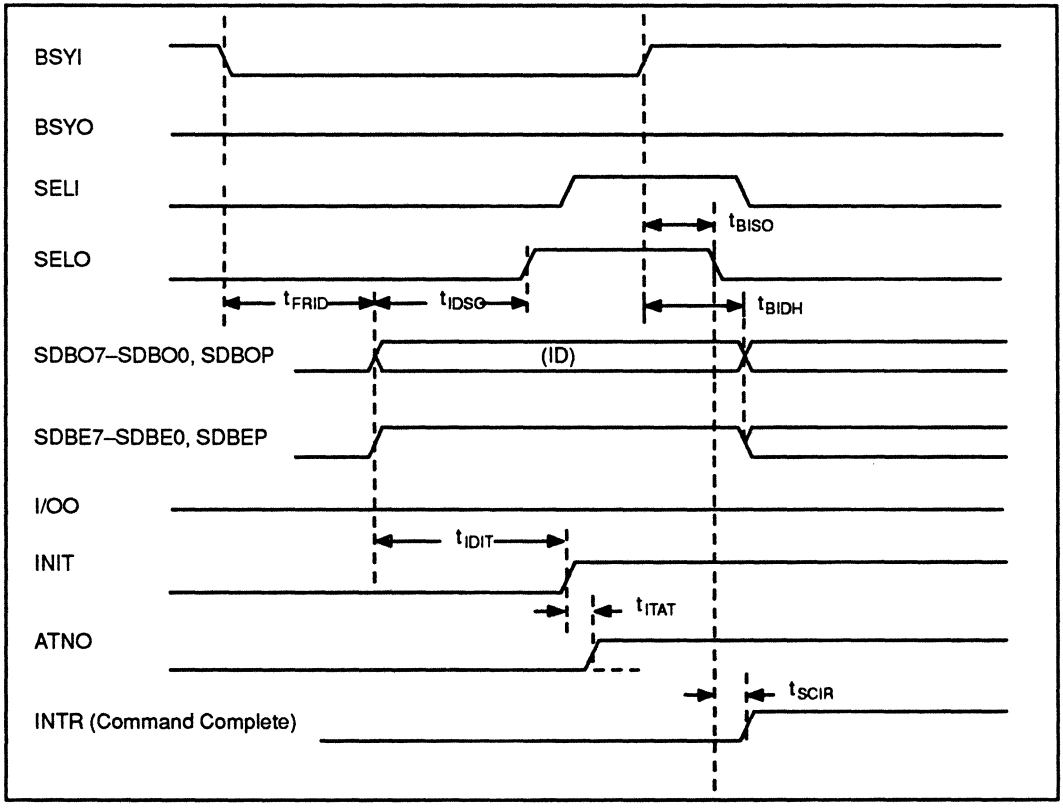


AC CHARACTERISTICS (Continued)

INITIATOR without Arbitration

Parameter	Designator	Values			Unit
		Min.[1,2]	Typ.	Max.	
Bus Free to Data Bus (ID) Send	t_{FRID}	(6 + n) x t CLF + 5		(7 + n) x t CLF- 85	ns
ID Send to SELO High	t_{IDSO}	$11t_{CLF}$– 50	$11t_{CLF}$– 15	$11t_{CLF}$+ 25	ns
ID Send to INIT High	t_{IDIT}	$11t_{CLF}$– 50	$11t_{CLF}$	$11t_{CLF}$+ 40	ns
INIT hIgh to ATNO High	t_{ITAT}	–5	5	25	ns
BSYI High to SELO Low	t_{BISO}	$2t_{CLF}$+ 5			ns
BSYI High to Data Bus (ID) Hold	t_{BIDH}	$2t_{CLF}$+ 5			ns
SELO Low to INTR High	t_{SCIR}		0	35	ns

Notes: 1. Refer to "Clock Signal" timing for definition of t_{CLF}.
2 n = TCL register set value



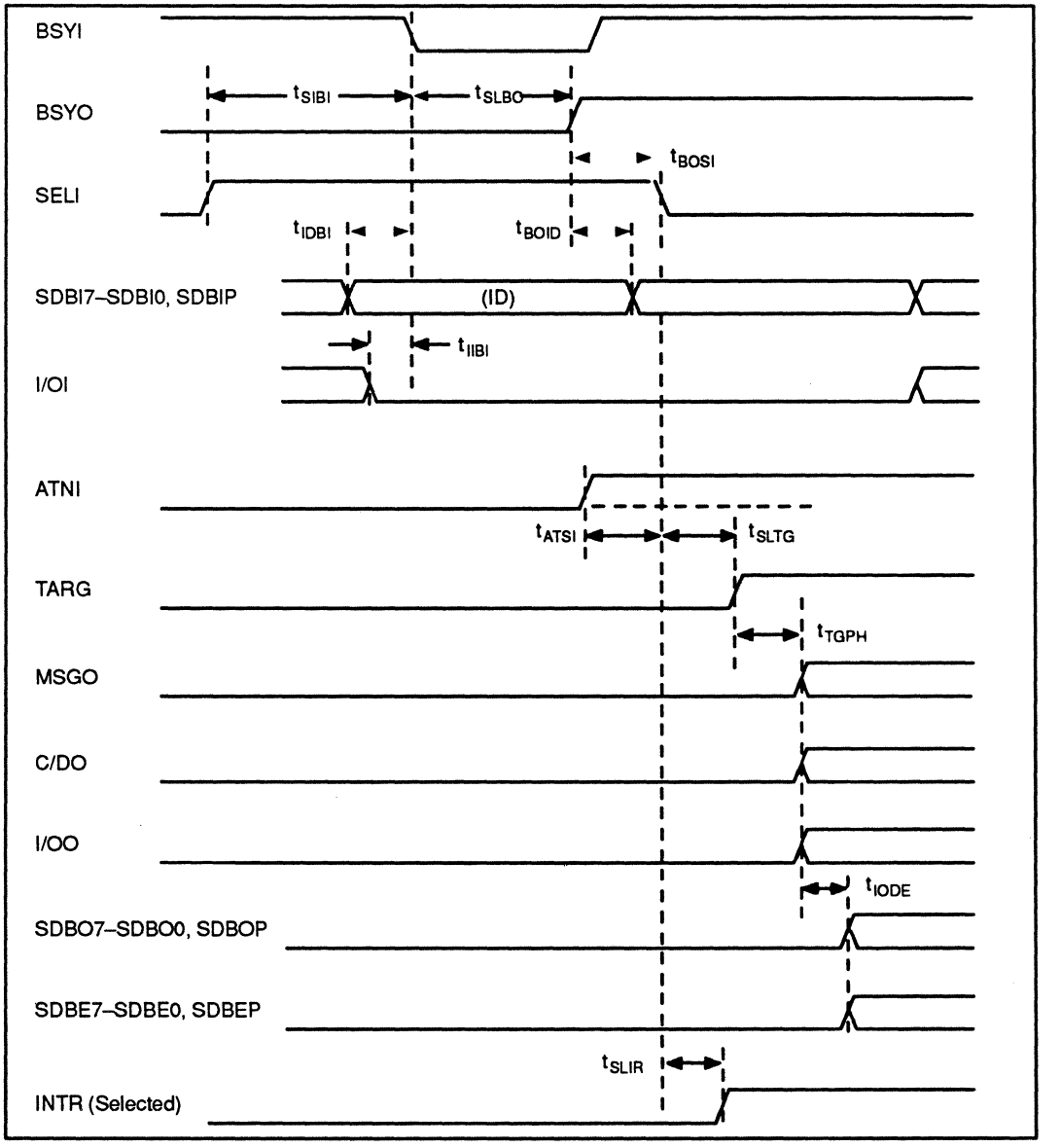
AC CHARACTERISTICS (Continued)

TARGET with Arbitration

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
SELI High to BSYI Low	t_{SIBI}	0			ns
Data Bus (ID) Valid to BSYI Low	t_{IDBI}	0			ns
I/OI Low to BSYI Low	t_{IIBI}	0			ns
BSYI Low to BSYO High (Response time)	t_{SLBO}	$4t_{CLF}$[1]+5		$5t_{CLF}$+ 60	ns
BSYO High to Data Bus (ID) Hold	t_{BOID}	20			ns
BSYO High to SELI Low	t_{BOSI}	0			ns
ATNI High to SELI Low	t_{ATSI}	0			ns
SELI Low to TARG High	t_{SLTG}	$3t_{CLF}$+ 5		$4t_{CLF}$+ 60	ns
TARG High to Phase Signal Output	t_{TGPH}	–5	10	30	ns
I/OO High to Data Bus Enable[2]	t_{IODE}	$4t_{CLF}$– 30	$4t_{CLF}$+ 20	$4t_{CLF}$+ 70	ns
SELI Low to INTR High	t_{SLIR}			$3t_{CLF}$+ 65	ns

Notes:
1. Refer to "Clock Signal" timing for definition of t_{CLF}
2. In case of bit 0 (I/O out) of PCTL register is set in advance.

AC CHARACTERISTICS (Continued)



AC CHARACTERISTICS (Continued)

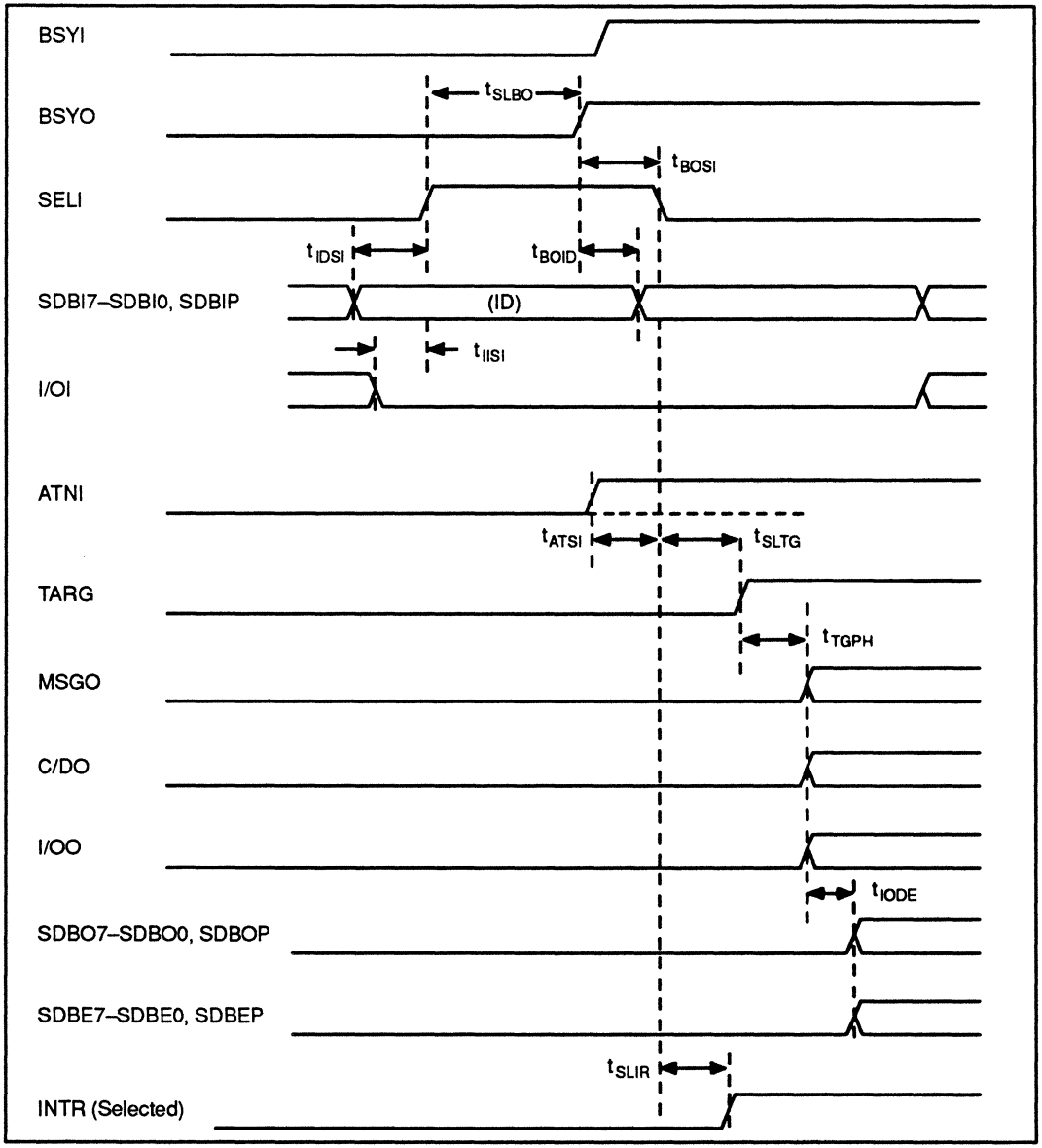
TARGET without Arbitration

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
Data Bus (ID) Valid to SELI High	t_{IDSI}	0			ns
I/OI Low to SELI High	t_{IISI}	0			ns
SELI High to BSYO High (Response time)	t_{SLBO}	$2t_{CLF}$[1] + 5		$3t_{CLF}$ + 65	ns
BSYO High to Data Bus (ID) Hold	t_{BOID}	20			ns
BSYO High to SELI Low	t_{BOSI}	0			ns
ATNI High to SELI Low	t_{ATSI}	0			ns
SELI Low to TARG High	t_{SLTG}	$3t_{CLF}$ + 5		$4t_{CLF}$ + 60	ns
TARG High to Phase Signal Output[2]	t_{TGPH}	−5	10	30	ns
I/OO High to Data Bus Enable	t_{IODE}	$4t_{CLF}$ − 30	$4t_{CLF}$ + 20	$4t_{CLF}$ + 70	ns
SELI Low to INTR High	t_{SLIR}			$3t_{CLF}$ + 65	ns

Notes:

1. Refer to "Clock Signal" timing for definition of t_{CLF}
2. In case bit 0 (I/O) of PCTL register is set in advance.

AC CHARACTERISTICS (Continued)



AC CHARACTERISTICS (Continued)

SCSI Interface (Reselection Phase)

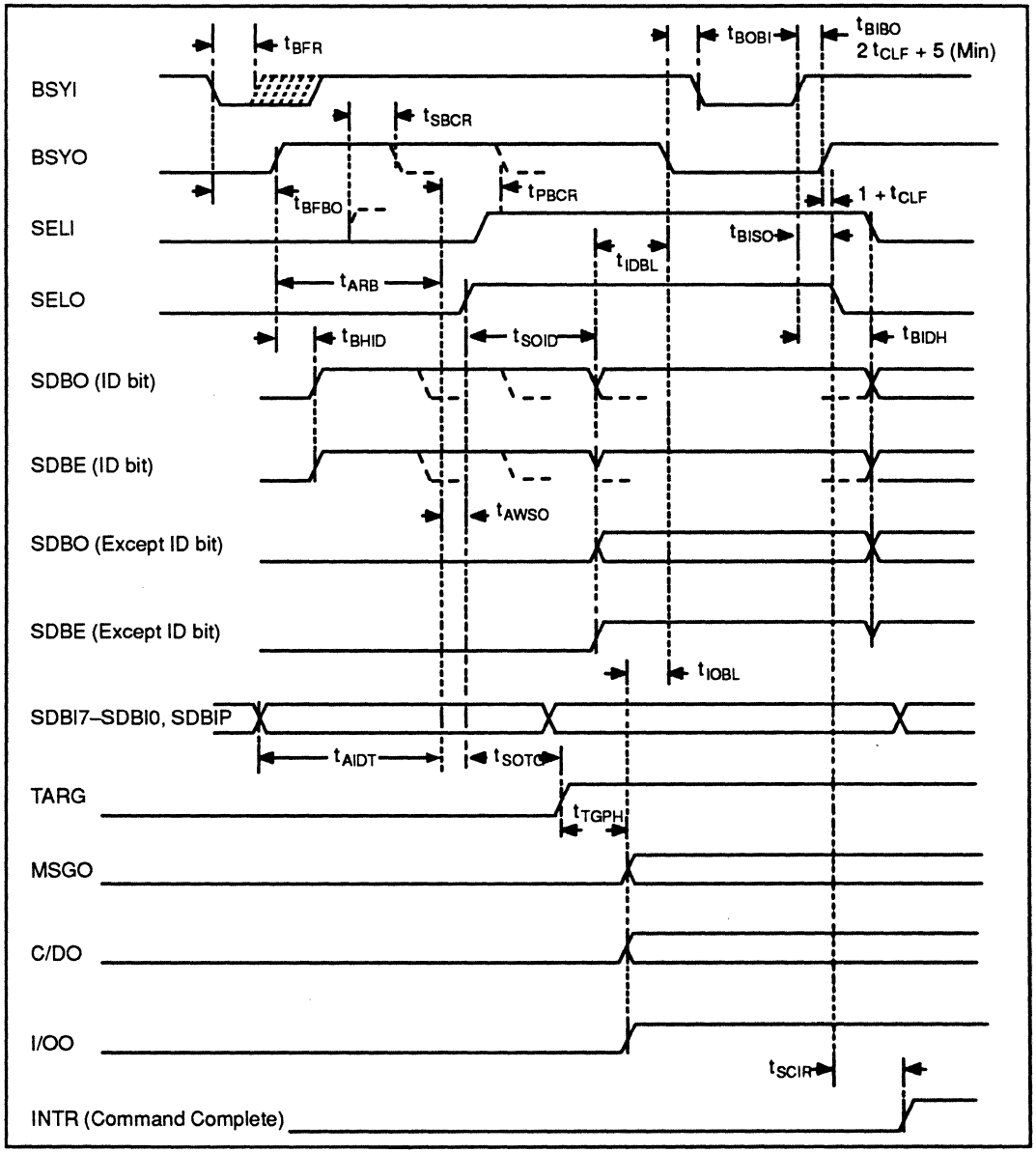
TARGET

Parameter	Designator	Values			Unit
		Min.[1,2]	Typ.	Max.	
Bus Free Time	t_{BFR}	$4t_{CLF}+50$			ns
BSYI Low to BSYO High (Start of arbitration)	t_{BFBO}	(6 + n) x t_{CLF} + 5		(7 + n) x t_{CLF} + 65	ns
BSYO High to ID Bit High	t_{BHID}	0	20	55	ns
BSYO High to Prioritize	t_{ARB}	$3t_{CLF}-40$			ns
Data Bus Valid (High Priority Bit) to Prioritize	t_{AIDT}	70			ns
Data Bus Valid (Low Priority Bit) to Prioritize		5			ns
Bus Usage Permission Grant to SELO High	t_{AWSO}	0		45	ns
SELO High to Data Bus (ID) Send	t_{SOID}	$11t_{CLF}-30$	$11t_{CLF}+15$	$11t_{CLF}+45$	ns
SELO High to TARG High	t_{SOTG}	$11t_{CLF}-30$	$11t_{CLF}+5$	$11t_{CLF}+30$	ns
TARG High to Phase Signal Send	t_{TGPH}	−5	10	30	ns
I/OO High to BSYO Low	t_{IOBL}	$2t_{CLF}-40$	$2t_{CLF}-10$	$2t_{CLF}+20$	ns
Data Bus (ID) Send to BSYO Low	t_{IDBL}	$2t_{CLF}-50$	$2t_{CLF}-10$	$2t_{CLF}+25$	ns
BSYO Low to BSYI Low	t_{BOBI}	0		t_{CLF}	ns
BSYI High to SELO Low	t_{BISO}	$3t_{CLF}+5$			ns
BSYI High to Data Bus (ID) Hold	t_{BIDH}	$3t_{CLF}+5$			ns
SELO Low to INTR High	t_{SCIR}		0	35	ns
SELI High to BSYO, ID Bit Low	t_{SBCR}			$3t_{CLF}+115$	ns
Prioritize to BSYO, ID Bit Low	t_{PBCR}			125	ns
BSYI High to BSYO High	t_{BIBO}	$2t_{CLF}+5$			ns

Notes:

1. Refer to "Clock Signal" timing for definition of t_{CLF}
2. n = TCL register set value

AC CHARACTERISTICS (Continued)

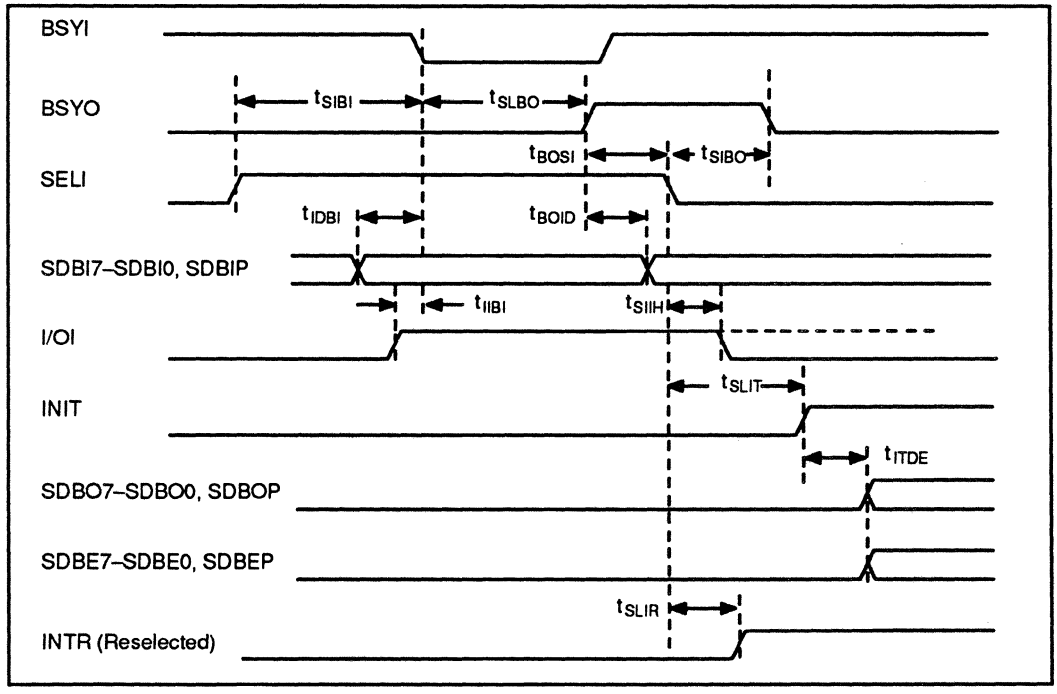


AC CHARACTERISTICS (Continued)

INITIATOR

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
SELI High to BSYI Low	t_{SIBI}	0			ns
Data Bus (ID) Valid to BSYI Low	t_{IDBI}	0			ns
I/OI Low to BSYI Low	t_{IIBI}	0			ns
BSYI Low to BSYO High (Response time)	t_{SLBO}	$4t_{CLF}+5$ (Note)		$5t_{CLF}+60$	ns
BSYO High to Data Bus (ID) Hold	t_{BOID}	20			ns
BSYO High to SELI Low	t_{BOSI}	0			ns
SELI Low to BSYO Low	t_{SIBO}	$2t_{CLF}+5$		$3t_{CLF}+60$	ns
SELI Low to I/OI Hold	t_{SIIH}	$4t_{CLF}+20$			ns
SELI Low to INTR High	t_{SLIR}			$3t_{CLF}+65$	ns
SELI Low to INIT High	t_{SLIT}	$3t_{CLF}+5$		$4t_{CLF}+65$	ns
INIT High to Data Bus Enable (With I/OI at low level)	t_{ITDE}	10			ns

Note: 1. Refer to "Clock Signal" timing for definition of t_{CLF}.



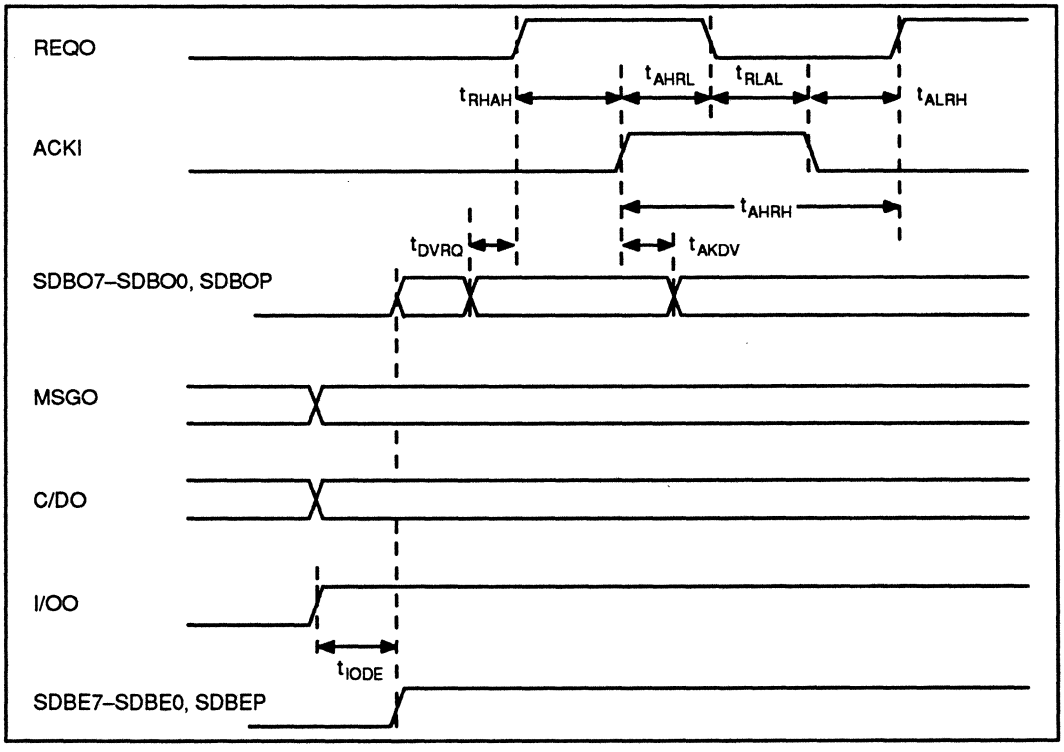
AC CHARACTERISTICS (Continued)

SCSI Interface (Transfer Phase)

Asynchronous Transfer Output (TARGET)

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
I/OO High to Data Bus Enable	t_{IODE}	$3t_{CLF}$		$4t_{CLF}+100$	ns
Data Bus Valid to REQO High	t_{DVRQ}	$2t_{CLF}-80$			ns
ACKI High to Data Bus Hold	t_{AKDV}	15	55		ns
REQO High to ACKI High	t_{RHAH}	20			ns
ACKI High to REQO Low	t_{AHRL}	10	30	55	ns
REQO Low to ACKI Low	t_{RLAL}	0			ns
ACKL Low to REQO High	t_{ALRH}		0	35	ns
ACKI High to REQO High	t_{AHRH}	$2t_{CLF}+5$			ns

Note: 1. Refer to "Clock Signal" timing for definition of t_{CLF}.

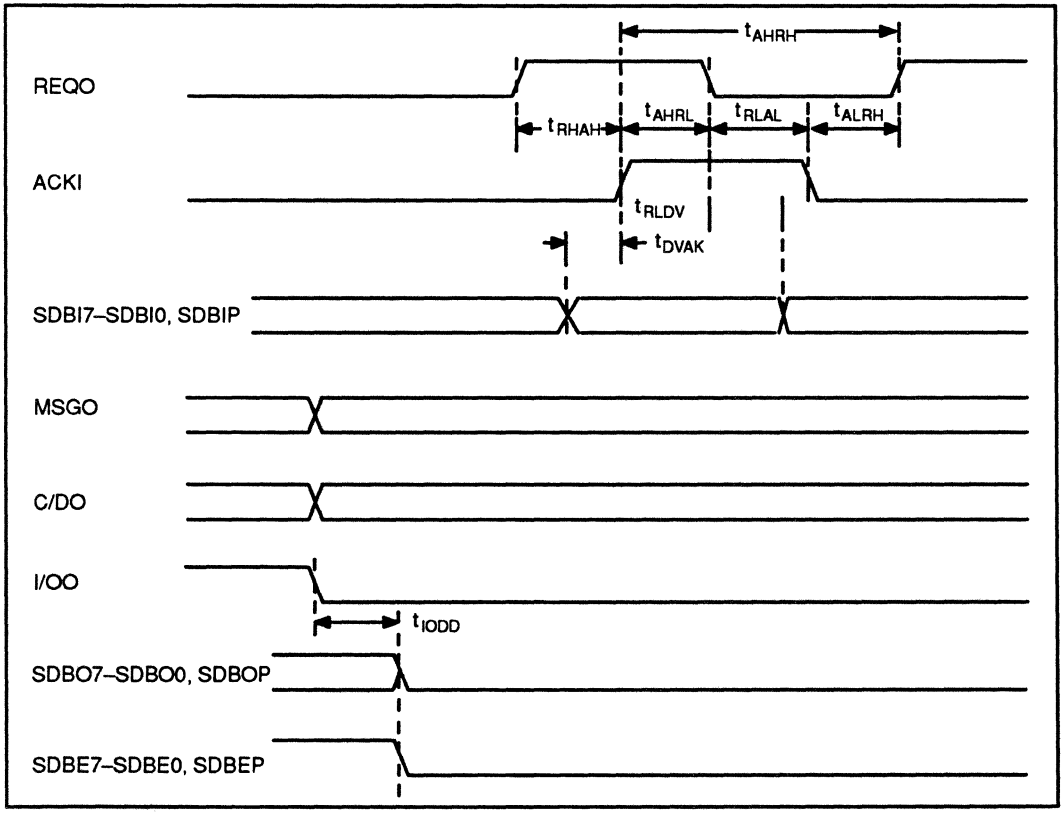


AC CHARACTERISTICS (Continued)

Asynchronous Transfer Input (TARGET)

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
I/OO Low to Data Bus Disable	t_{IODD}		20	55	ns
Data Bus Valid to ACKI High10	t_{DVAK}				ns
REQO Low to Data Bus Hold	t_{RLDV}	25			ns
REQO High to ACKI High	t_{RHAH}	20			ns
ACKI High to REQO Low	t_{AHRL}	10	30	55	ns
REQO Low to ACKI Low	t_{RLAL}	0			ns
ACKI Low to REQO High	t_{ALRH}	0	35		ns
ACKI High to REQO High	t_{AHRH}	$2t_{CLF}+5$			ns

Note: 1. Refer to "Clock Signal" timing for definition of t_{CLF}

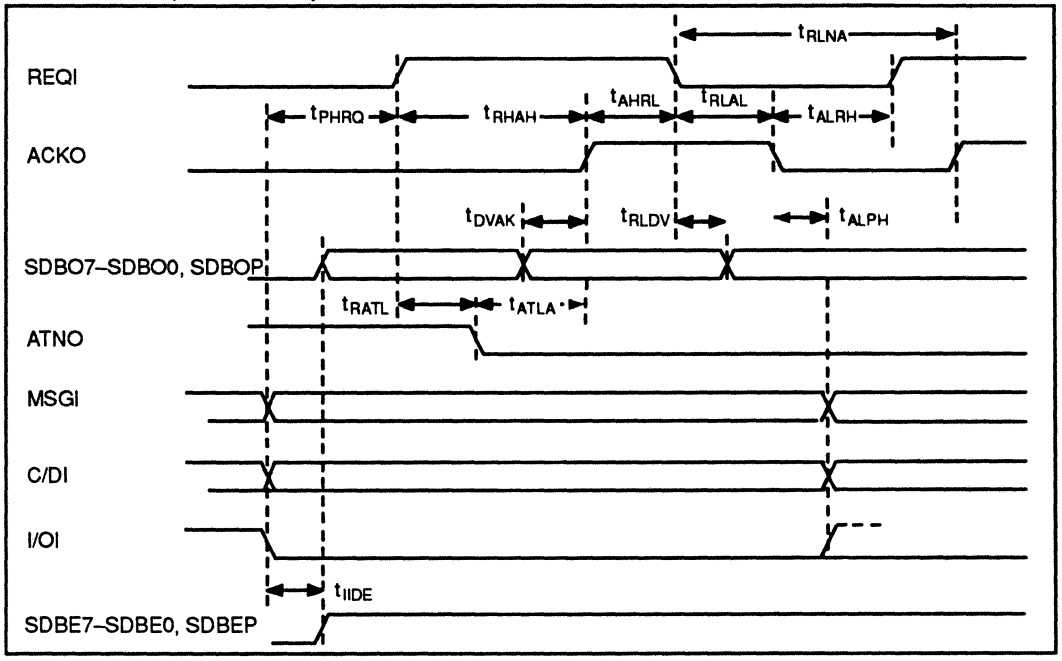


AC CHARACTERISTICS (Continued)

Asynchronous Transfer Output (INITIATOR)

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
I/OI Low to Data Bus Enable	t_{IIDE}	10	50	90	ns
Phase Specify to REQI High	t_{PHRQ}	100			ns
ACKO Low to Phase Change	t_{ALPH}	10			ns
REQI High to ATNO Low[1]	t_{RATL}	$2t_{CLF}+5$			ns
ATNO Low to ACKO High[1]	t_{ATLA}	$t_{CLF}-20$			ns
Data Bus Valid to ACKO High	t_{DVAK}	$2t_{CLF}-80$			ns
REQI Low to Data Bus Hold	t_{RLDV}	20	60		ns
REQI High to ACKO High	t_{RHAH}	20	45	75	ns
ACKO High to REQI Low	t_{AHRL}	0			ns
REQI Low to ACKO Low[3]	t_{RLAL}	10	45	75	ns
ACKO Low to REQI High	t_{ALRH}	10			ns
REQI Low to ACKO High	t_{RLNA}	$2t_{CLF}+5$			ns

Notes: 1. With these timing parameters, the ATNO signal is reset only when the last byte is sent at the MESSAGE OUT phase.
2. Refer to "Clock Signal" timing for definition of t_{CLF}
3. Except for the last byte.

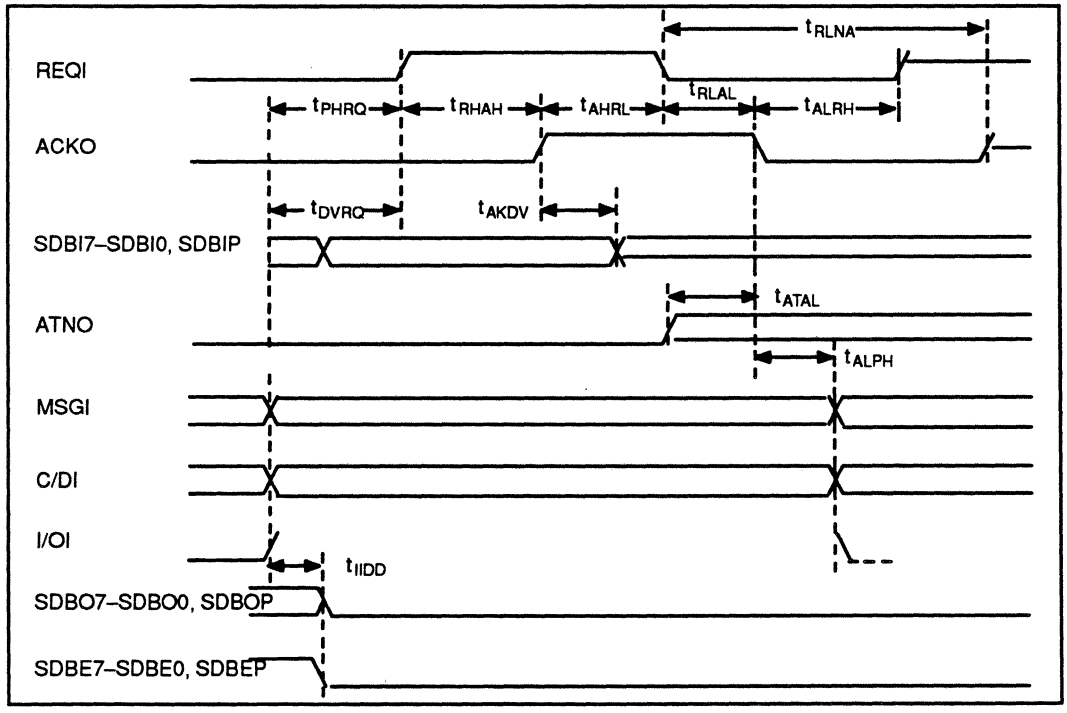


AC CHARACTERISTICS (Continued)

Asynchronous Transfer Input (INITIATOR)

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
I/OI High to Data Bus Disable	t_{IIDD}		45	75	ns
Phase Specify to REQI High	t_{PHRQ}	100			ns
ACKO Low to Phase Change	t_{ALPH}	10			ns
Data Bus Valid to REQI High	t_{DVRQ}	10			ns
ACKO High to Data Bus Hold	t_{AKDV}	15			ns
REQI High to ACKO High	t_{RHAH}	15	40	70	ns
ACKO High to REQI Low	t_{AHRL}	0			ns
REQI Low to ACKO Low	t_{RLAL}	10	45	75	ns
ACKO Low to REQI High	t_{ALRH}	10			ns
REQI Low to ACKO High	t_{RLNA}	t_{CLF}[1] + 5	t_{CLF}[1] + 15		ns
ATNO High to ACKO Low[2]	t_{ATAL}	t_{CLF} - 20			ns

Notes: 1. Refer to "Clock Signal" timing for definition of t_{CLF}
2. With these timing parameters, the ATNO signal is sent only when parity checking is enabled and a parity error is detected in the input data.

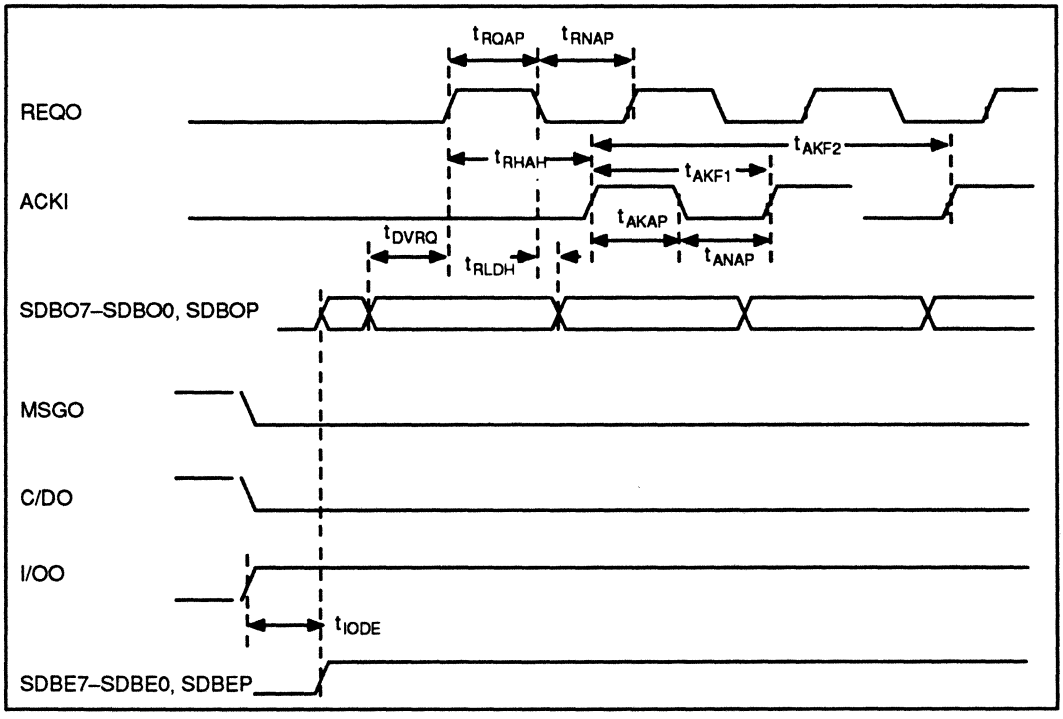


AC CHARACTERISTICS (Continued)

Synchronous Transfer Output (TARGET)

Parameter	Designator	Values			Unit
		Min.[1,2]	Typ.	Max.	
I/OO High to Data Bus Enable	t_{IODE}	$3t_{CLF}$		$4t_{CLF} + 100$	ns
Data Bus Valid to REQO High	t_{DVRQ}	t_{CLF}			ns
REQO Assertion Period	t_{RQAP}	$t_{CLF} - 10$	t_{CLF}		ns
REQO Nonassertion Period	t_{RNAP}	$nt_{CLF} - 10$	nt_{CLF}		ns
REQO Low to Data Bus Hold	t_{RLDH}	0	5		ns
REQO High to ACKI High	t_{RHAH}	0			ns
ACKI cycle time (1)	t_{AKF1}	t_{CLF}			ns
ACKI cycle time (2)	t_{AKF2}	$3t_{CLF}$			ns
ACKI Assertion Period	t_{AKAP}	50			ns
ACKI Nonassertion Period	t_{ANAP}	50			ns

Notes: 1. Refer to "Clock Signal" timing for definition of t_{CLF}.
2. n = TMOD register set value

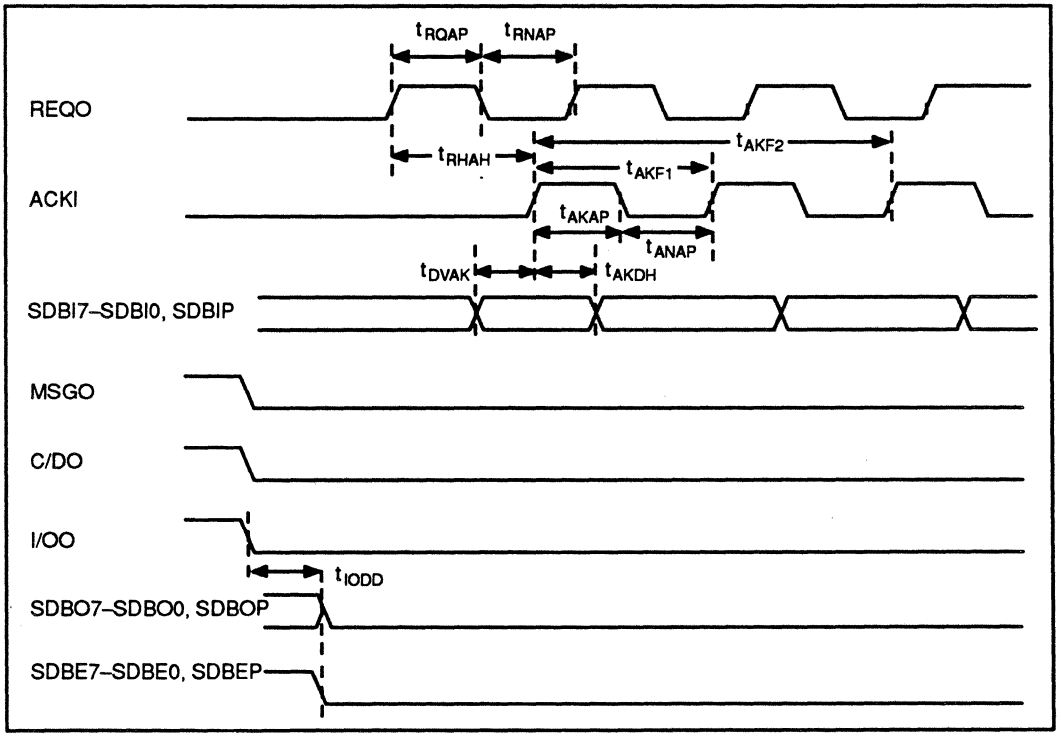


AC CHARACTERISTICS (Continued)

Synchronous Transfer Input (TARGET)

Parameter	Designator	Values			Unit
		Min.[1,2]	Typ.	Max.	
I/OO Low to Data Bus Disable	t_{IODD}		20	55	ns
REQO Assertion Period	t_{RQAP}	t_{CLF} – 10	t_{CLF}		ns
REQO Nonassertion Period	t_{RNAP}	$n\,t_{CLF}$ – 10	$n\,t_{CLF}$		ns
REQO High to ACKI High	t_{RHAH}	0			ns
ACKI Assertion Period	t_{AKAP}	50			ns
ACKI Nonassertion Period	t_{ANAP}	50			ns
ACKI Cycle time (1)	t_{AKF1}	t_{CLF}			ns
ACKI Cycle time (2)	t_{AKF2}	$3\,t_{CLF}$			ns
Data Bus Valid to ACKI High	t_{DVAK}	10			ns
ACKI High to Data Bus Hold	t_{AKDH}	40			ns

Notes: 1. Refer to "Clock Signal" timing for definition of t_{CLF}
2. n = TMOD register set value

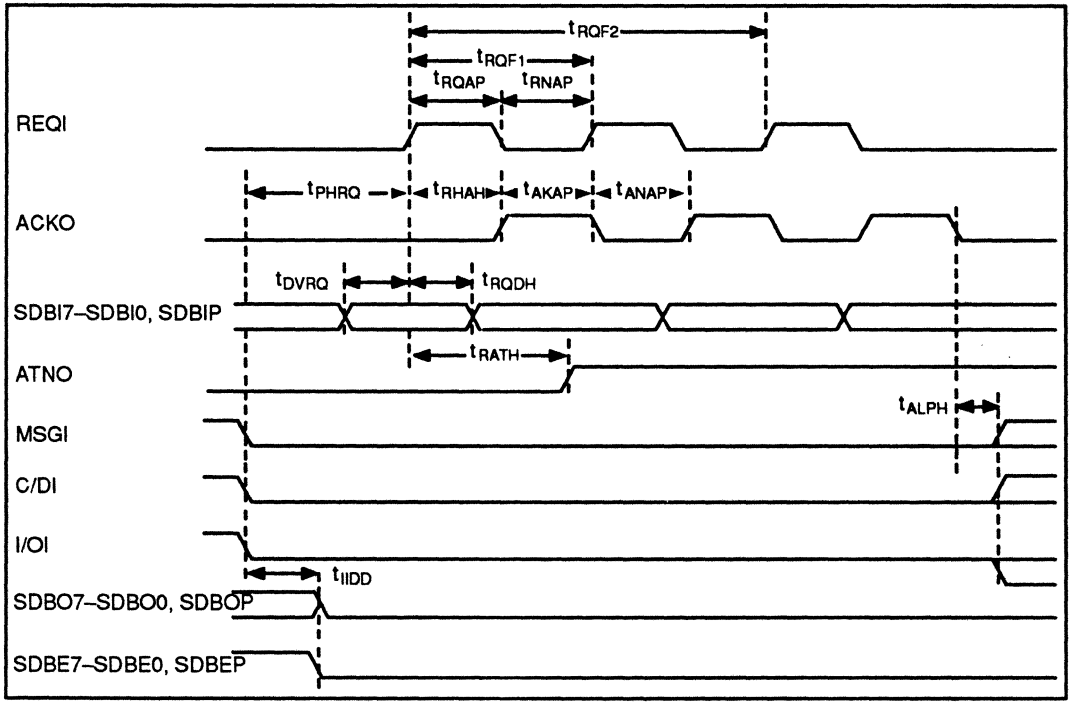


AC CHARACTERISTICS (Continued)

Synchronous Transfer Output (INITIATOR)

Parameter	Designator	Values			Unit
		Min.[1,2]	Typ.	Max.	
I/OI Low to Data Bus Enable	t_{IIDE}	10	50	90	ns
Phase Specify to REQI High	t_{PHRQ}	100			ns
ACKO Low to Phase Change	t_{ALPH}	10			ns
REQI Assertion Period	t_{RQAP}	50			ns
REQI Nonassertion Period	t_{RNAP}	50			ns
REQI Cycle time (1)	t_{RQF1}	t_{CLF}			ns
REQI Cycle time (2)	t_{RQF2}	$3t_{CLF}$			ns
REQI High to ACKO High	t_{RHAH}	$3t_{CLF}$			ns
ACKO Assertion Period	t_{AKAP}	$t_{CLF}-10$	t_{CLF}		ns
ACKO Nonassertion Period	t_{ANAP}	$nt_{CLF}-10$	nt_{CLF}		ns
Data Bus Valid to ACKO High	t_{DTAK}	t_{CLF}			ns
ACKO Low to Data Bus Hold	t_{ALDH}	0	5		ns

Notes: 1. Refer to "Clock Signal" timing for definition of t_{CLF}.
2. n = TMOD register set value



AC CHARACTERISTICS (Continued)

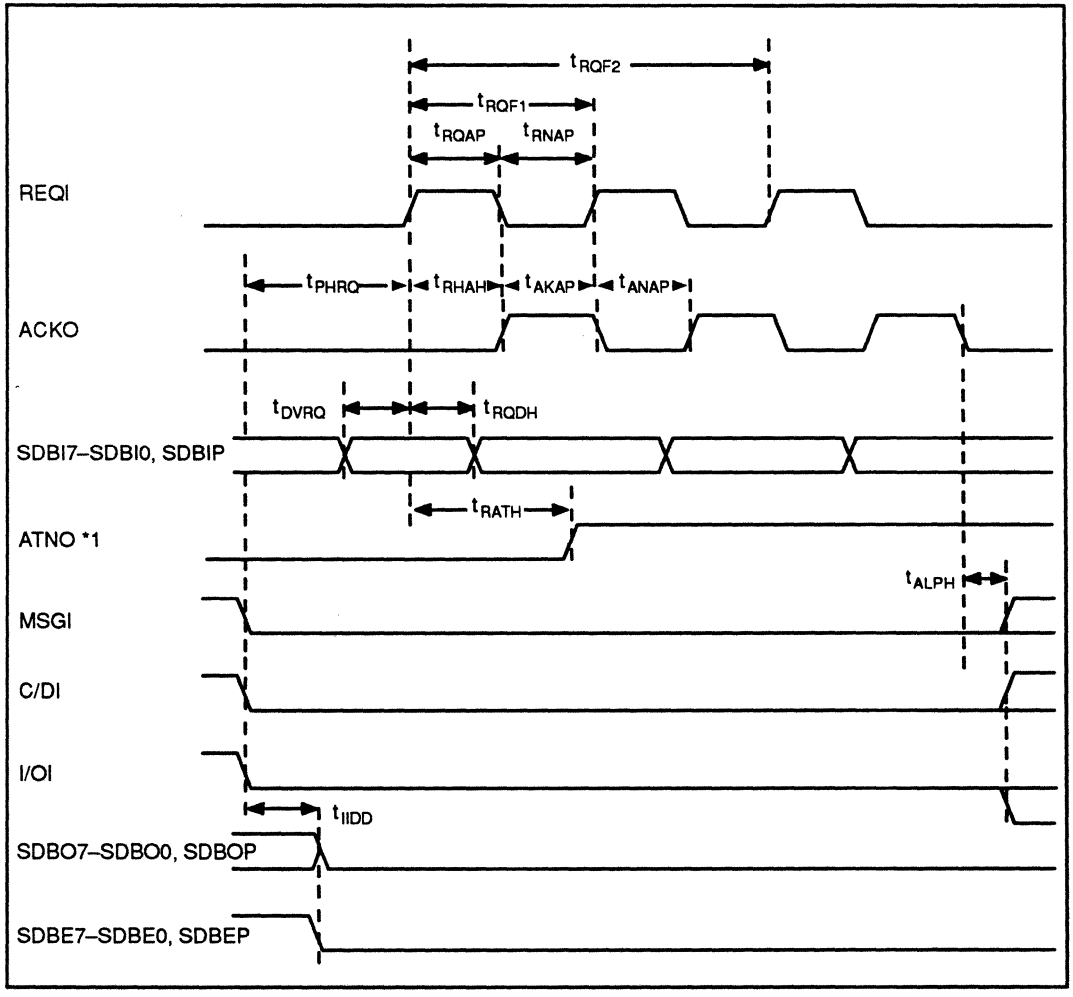
Synchronous Transfer Input (INITIATOR)

Parameter	Designator	Values			Unit
		Min.[1,2]	Typ.	Max.	
I/OI High to Data Bus Disable	t_{IDD}		45	75	ns
Phase Specify to REQI High	t_{PHRQ}	100			ns
ACKO Low to Phase Change	t_{ALPH}	10			ns
Data Bus Valid to REQI High	t_{DVRQ}	10			ns
REQI High to Data Bus Hold	t_{RQDH}	40			ns
REQI Assertion Period	t_{RQAP}	50			ns
REQI Nonassertion Period	t_{RNAP}	50			ns
REQI Cycle time (1)	t_{RQF1}	t_{CLF}			ns
REQI Cycle time (2)	t_{RQF2}	$3t_{CLF}$			ns
REQI High to ACKO High	t_{RHAH}	$6t_{CLF}+5$			ns
ACKO Assertion Period	t_{AKAP}	$t_{CLF}-10$	t_{CLF}		ns
ACKO Nonassertion Period	t_{ANAP}	$nt_{CLF}-10$	$n\,t_{CLF}$		ns
REQI High to ATNO High[3]	t_{RATH}	$3t_{CLF}+5$	$4\,t_{CLF}-30$		ns

Notes:

1. Refer to "Clock Signal" timing for definition of t_{CLF}.
2. n = TMOD register set value
3. With these timing parameters, the ATNO signal is sent only when parity checking is enabled and a parity error is detected in the input data.

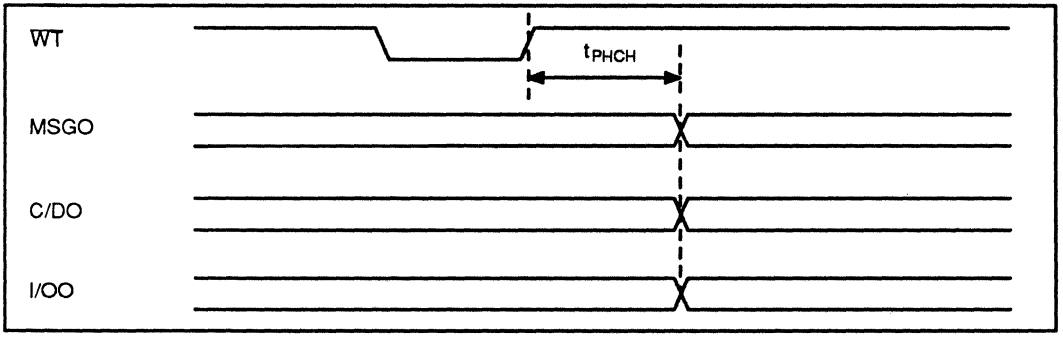
AC CHARACTERISTICS (Continued)



AC CHARACTERISTICS (Continued)

Transfer Phase Change (TARGET)

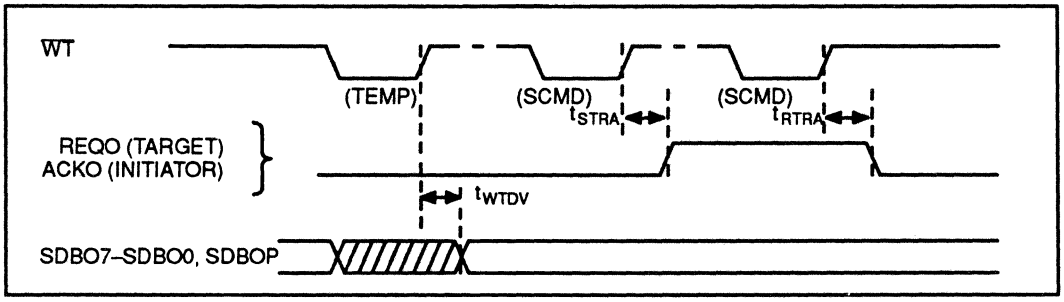
Parameter	Designator	Values			Unit
		Min	Typ	Max	
$\overline{WT}$ High to MSGO, C/DO, I/OO (PCTL register)	t_{PHCH}	10	40	65	ns



MANUAL TRANSFER[1]

Parameter	Designator	Values			Unit
		Min..[2]	Typ.	Max.	
$\overline{WT}$ High to Data Bus Valid (TEMP register output)	t_{WTDV}		40	60	ns
$\overline{WT}$ High to REQO, ACKO High (Set ACK/REQ command)	t_{STRA}	$2t_{CLF}+5$		$3t_{CLF}+80$	ns
$\overline{WT}$ High to REQO, ACKO Low (Reset ACK/REQ command)	t_{RTRA}	$2t_{CLF}+5$		$3t_{CLF}+80$	ns

Notes: 1. Timing sequences not shown here conform to the asynchronous transfer timing sequence.
2. Refer to "Clock Signal" timing for definition of t_{CLF}.

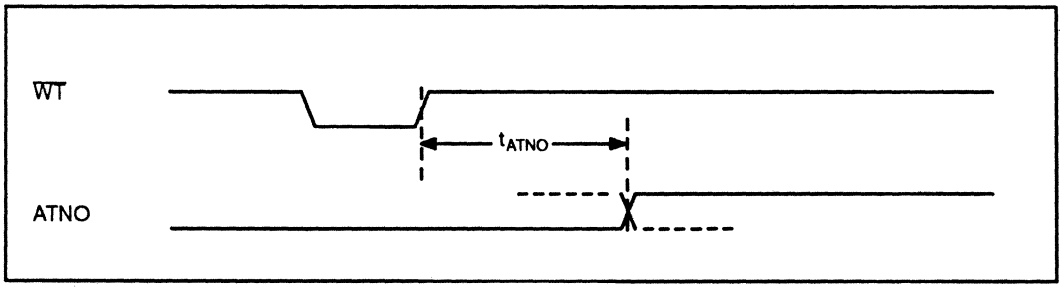


AC CHARACTERISTICS (Continued)

Attention Condition (INITIATOR)

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
$\overline{WT}$ High to ATNO High/Low (Set ATN command and Reset ATN command)	t_{ATNO}	$2t_{CLF}+5$ (Note)		$3t_{CLF}+90$	ns

Note:
1. Refer to "Clock Signal" timing for definitibn of t_{CLF}.



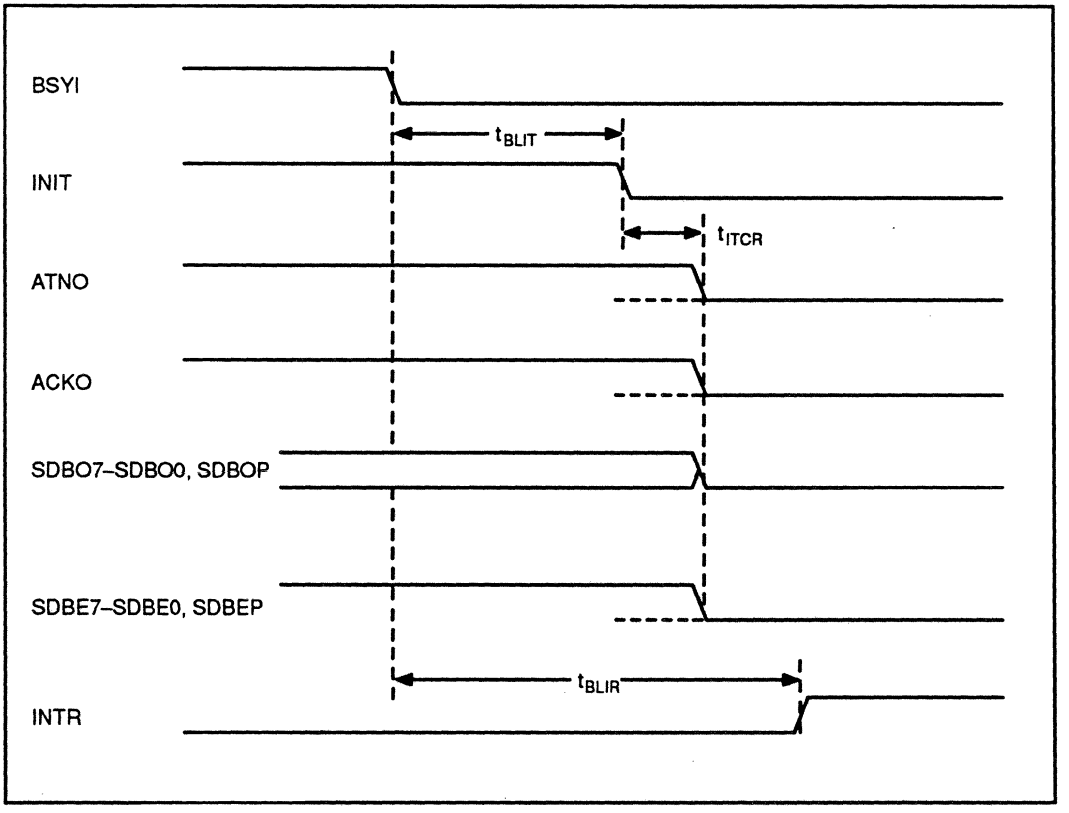
AC CHARACTERISTICS (Continued)

SCSI Interface (Bus Free)

INITIATOR (Disconnected)

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
BSYI Low to INIT Low	t_{BLIT}			$5\,t_{CLF}+70$ (Note)	ns
INIT Low to Bus Clear	t_{ITCR}		20	50	ns
BSYI Low to INTR High	t_{BLIR}			$6\,t_{CLF}+75$	ns

Note:
1. Refer to "Clock Signal" timing for definition of t_{CLF}.

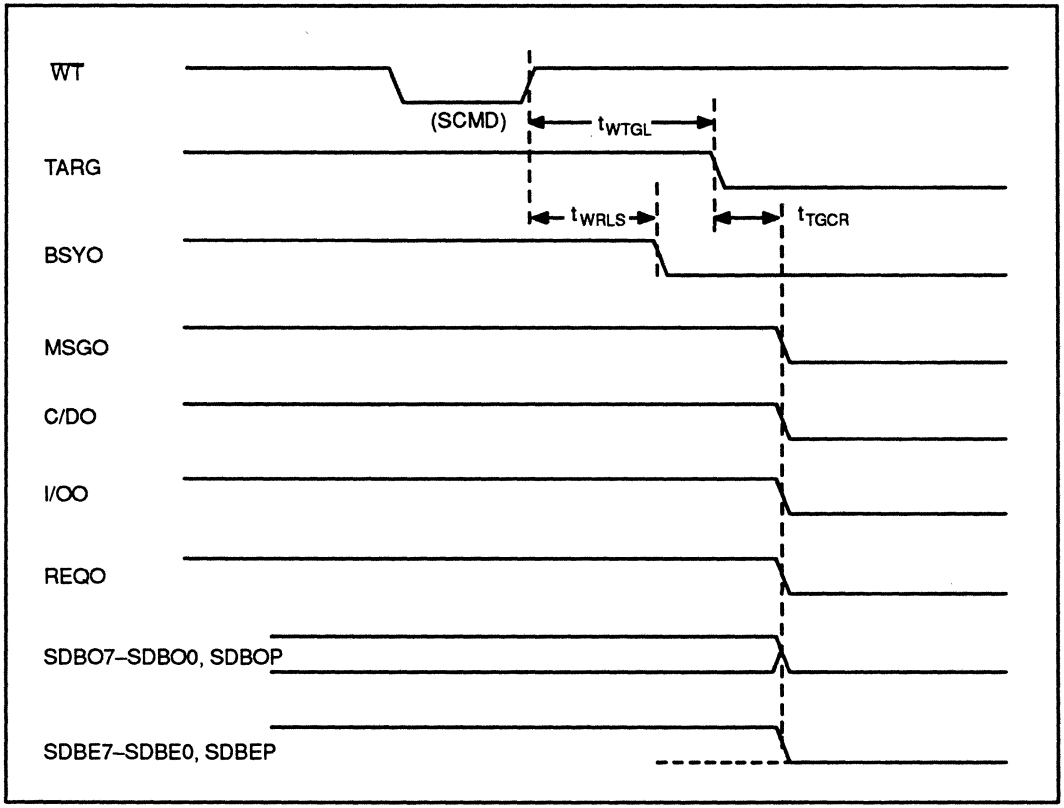


AC CHARACTERISTICS (Continued)

TARGET (Bus Release Command)

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
$\overline{WT}$ High to BSYO Low (SCMD register)	t_{WRLS}			$3t_{CLF}+80$ (Note)	ns
$\overline{WT}$ High to TARG Low (SCMD register)	t_{WTGL}			$3t_{CLF}+80$	ns
TARG Low to Bus Clear	t_{TGCR}		20	50	ns

Note:
1. Refer to "Clock Signal" timing for definitibn of t_{CLF}.

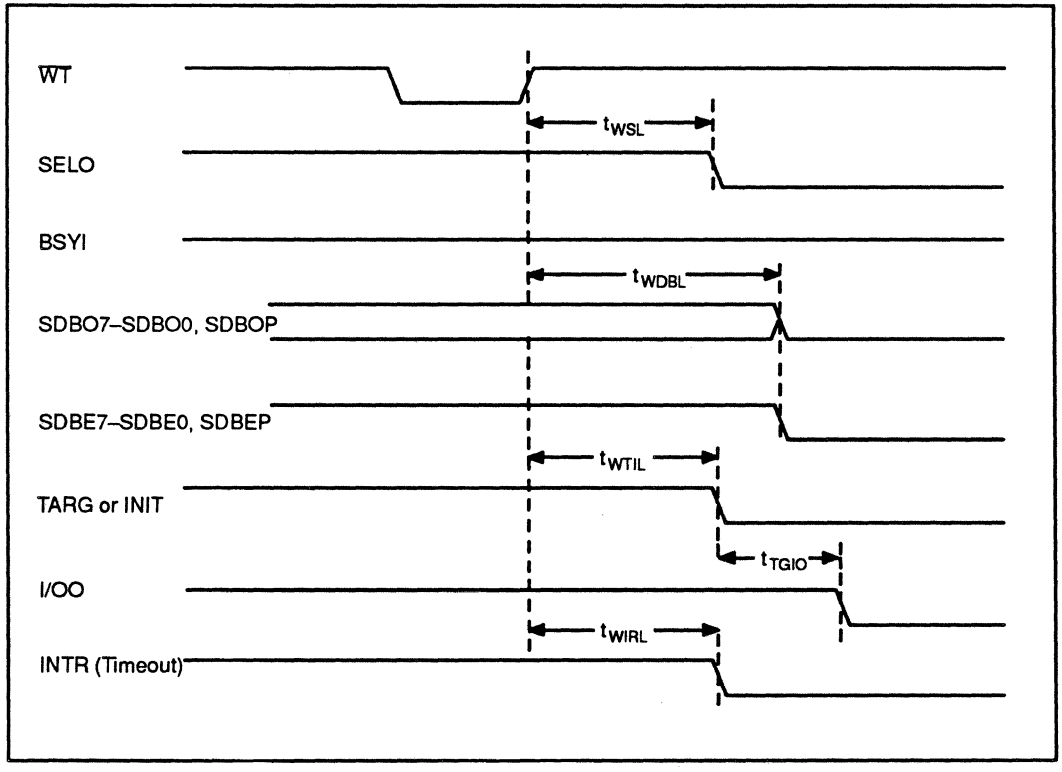


AC CHARACTERISTICS (Continued)

SELECTION/RESELECTION Phase Stop (Time-Out)

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
$\overline{WT}$ High to SELO Low (INTS register)	t_{WSL}			$3t_{CLF}$+ 80 (Note)	ns
$\overline{WT}$ High to Data Bus Disable (INTS register)	t_{WDBL}			$3t_{CLF}$+ 105	ns
$\overline{WT}$ High to TARG or INIT Low (INTS register)	t_{WTIL}			$3t_{CLF}$+ 80	ns
TARG Low to I/OO Low	t_{TGIO}		10	30	ns
$\overline{WT}$ High to INTR Low (INTS register)	t_{WIRL}			$3t_{CLF}$+ 105	ns

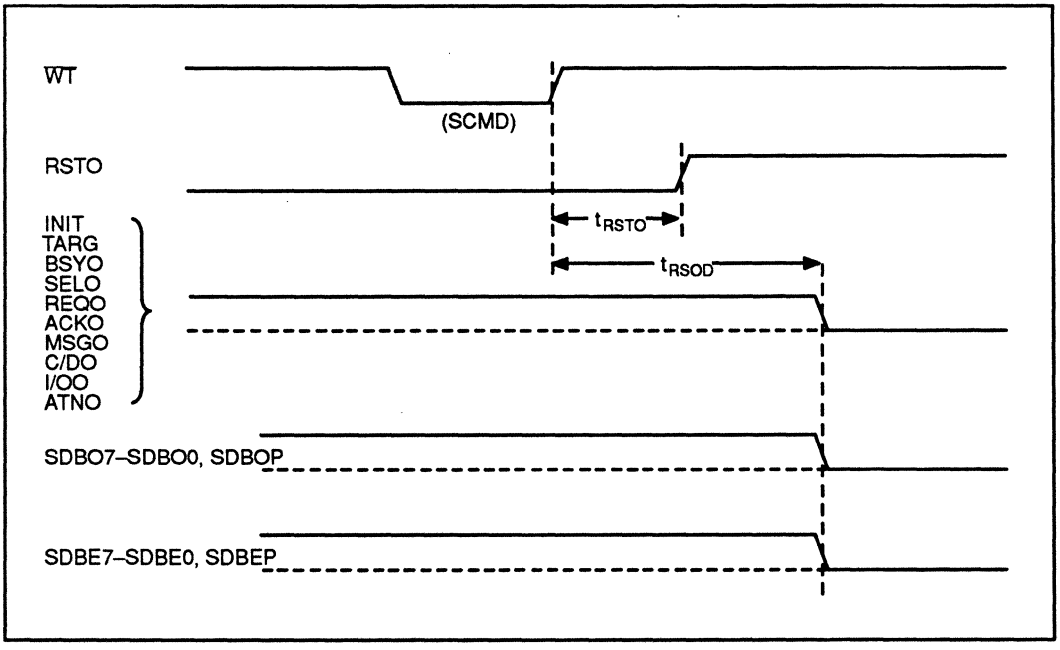
Note:
1. Refer to "Clock Signal" timing for definition of t_{CLF}



AC CHARACTERISTICS (Continued)

SCSI Interface (Reset Condition))

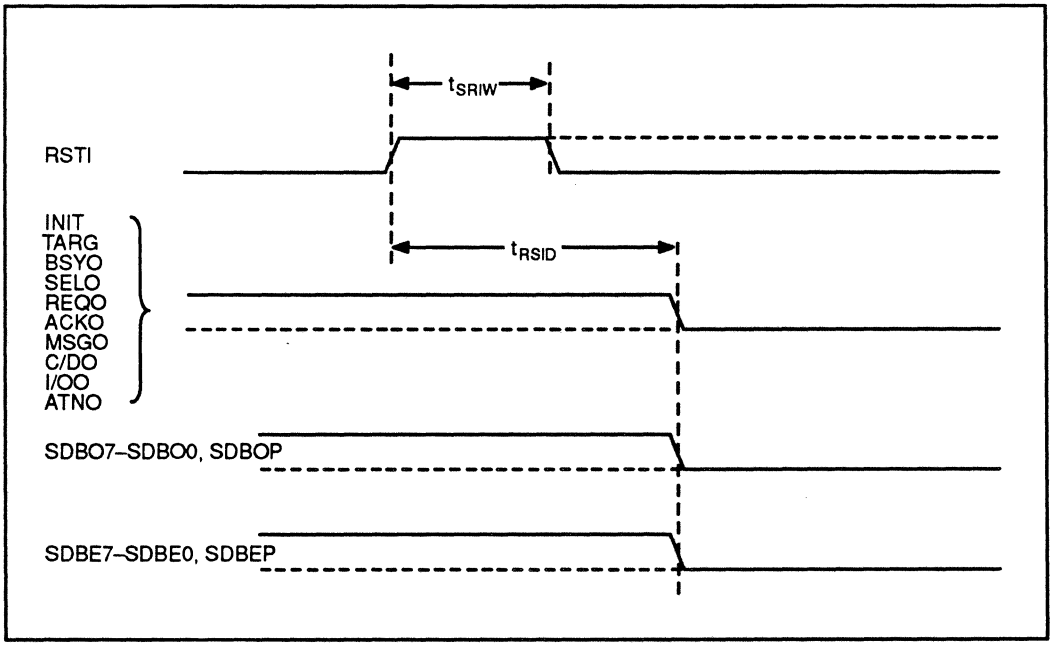
RST Signal Sending					
Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
WT High (bit 4 of SCMD register) to RSTO	t_{RSTO}	5	35	55	ns
Reset Delay	t_{RSOD}		70	115	ns



AC CHARACTERISTICS (Continued)

RST Signal Receiving					
Parameter	**Designator**	**Values**			**Unit**
		Min.[1]	**Typ.**	**Max.**	
RSTI Pulse Width	t_{SRIW}	$3\,t_{CLF}$			ns
Reset Delay	t_{RSID}			$4\,t_{CLF}+115$	ns

Note:
1. Refer to "Clock Signal" timing for definition of t_{CLF}.



CAPACITANCE (T_A = 25°C, V_{DD} = 0V, f_I = 1 MHz)

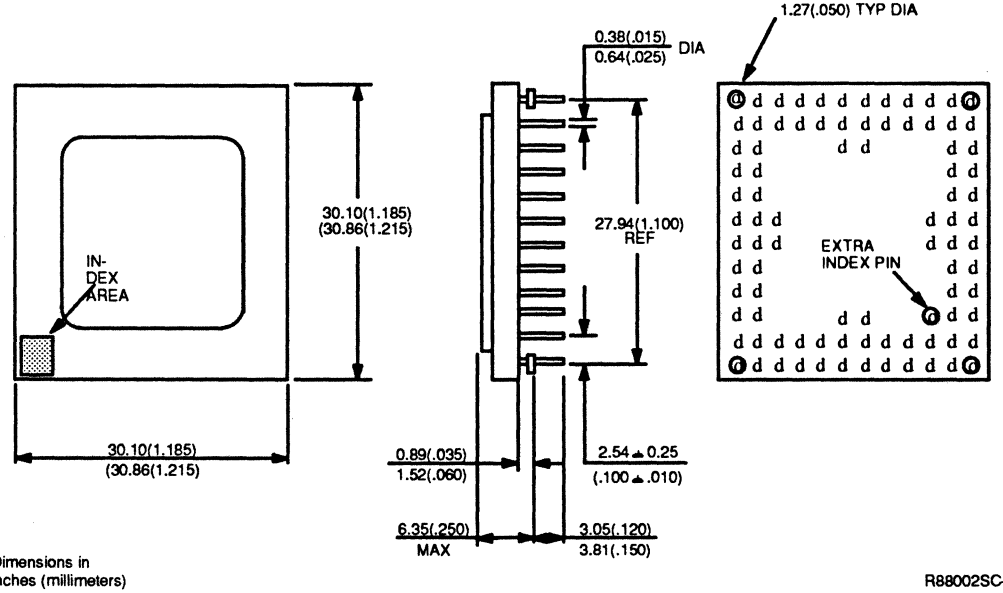
Parameter	**Designator**	**Values**			**Unit**
		Min	**Typ**	**Max**	
Input Pin Capacitance	C_{IN}			9	pF
Output Pin Capacitance	C_{OUT}			9	pF
I/O Pin Capacitance	$C_{I/O}$			11	pF

GRID ARRAY

88–Lead Ceramic Pin Grid Array

PGA–88C–A01

88–LEAD CERAMIC PIN GRID ARRAY
(CASE No. PGA–88C–A01)

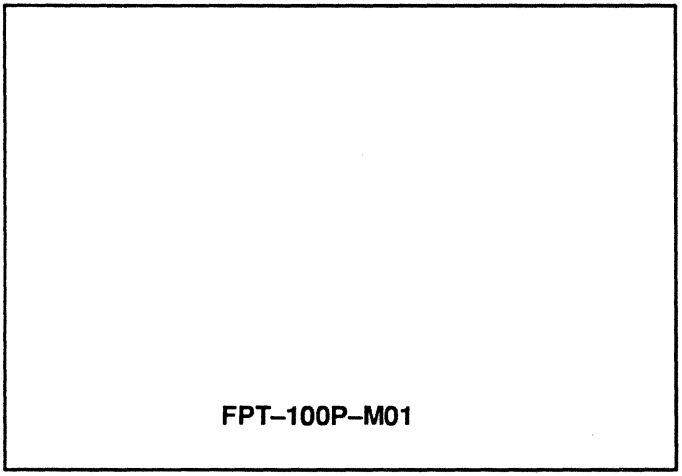


Dimensions in
inches (millimeters)

R88002SC–3C

GRID ARRAY (Continued)

100–Lead Plastic Flat Package



100–LEAD PLASTIC FLAT PACKAGE
(CASE NO.: FPT–100P–MO1)

MB87033B
SCSI Protocol Controller (SPC) with On-Chip Drivers/Receivers

GENERAL DESCRIPTION

The MB87033 SCSI Protocol Controller (SPC) is a CMOS LSI circuit specifically designed to control a Small Computer Systems Interface (SCSI). The MB87033 rounds out Fujitsu's SPC family of protocol controllers by providing software enhancements and other functional features that will meet all facets of the SCSI specification (ANSI X 3.131-1986).

To achieve optimum performance and interface flexibility, the MB87033 provides an 8-byte First-In First-Out (FIFO) data buffer register and a 28-bit transfer byte counter which allows burst transfers of up to 256 megabytes. To improve programming requirements, "Attention Detect" and "Arbitration Fail" interrupts are provided and on-chip driver/receiver circuits simplify interface connections. Data transfers can be executed in either the asynchronous or synchronous mode with a maximum offset of 8-bytes.

SCSI Compatibility
- Supports all commands.
- Software compatible with MB87030/31
- Serves as either INITIATOR or TARGET

Data Bus
- Independent buses for CPU and DMA controller

Transfer Modes
- Asynchronous
- Synchronous mode transfers with programmable offset of up to eight bytes (8-Byte FIFO)

Data Transfer Speed
- Up to a maximum of 5-megabytes/sec

Selectable Operating Modes
- DMA transfer
- Program transfer
- Manual transfer
- Diagnostic

Interface
- On-chip single ended Drivers/Receivers
- Guaranteed to sink 48mA regardless of the number of outputs simultaneously asserted.

Enhancements
- On-chip parity generation
- Attention condition detect interrupt
- Arbitration fail interrupt
- 8 byte FIFO
- 28 bit counter

Clock Requirements
- 10 MHz clock

Technology/Power Requirements
- Silicon-gate CMOS
- Single +5V power supply

Available Packaging
- 84-pin plastic leadless chip carrier
- 80-pin plastic flat package

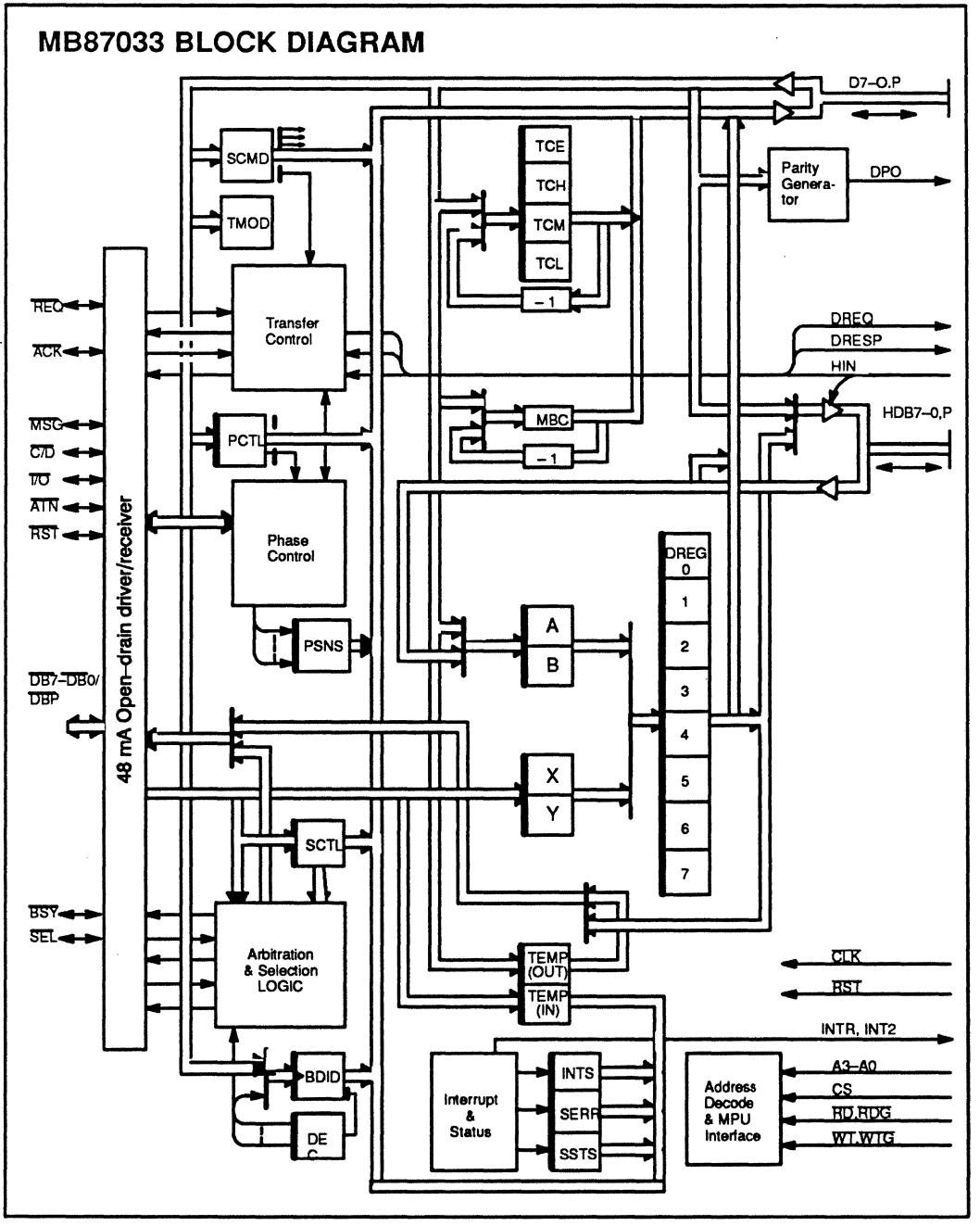
ABSOLUTE MAXIMUM RATINGS[1]

Rating	Symbol	Values		Unit
		Min	Max	
Supply Voltage	V_{DD}	V_{SS}[2] –0.3	6.0	V
Input Voltage	V_I	V_{SS}[2] –0.3	V_{DD} + 0.3	V
Output Voltage[2]	V_O	V_{SS}[2] –0.3	V_{DD} + 0.3	V
Storage Temperature (Ceramic)	T_{STG}	–65	+125	°C

NOTES:

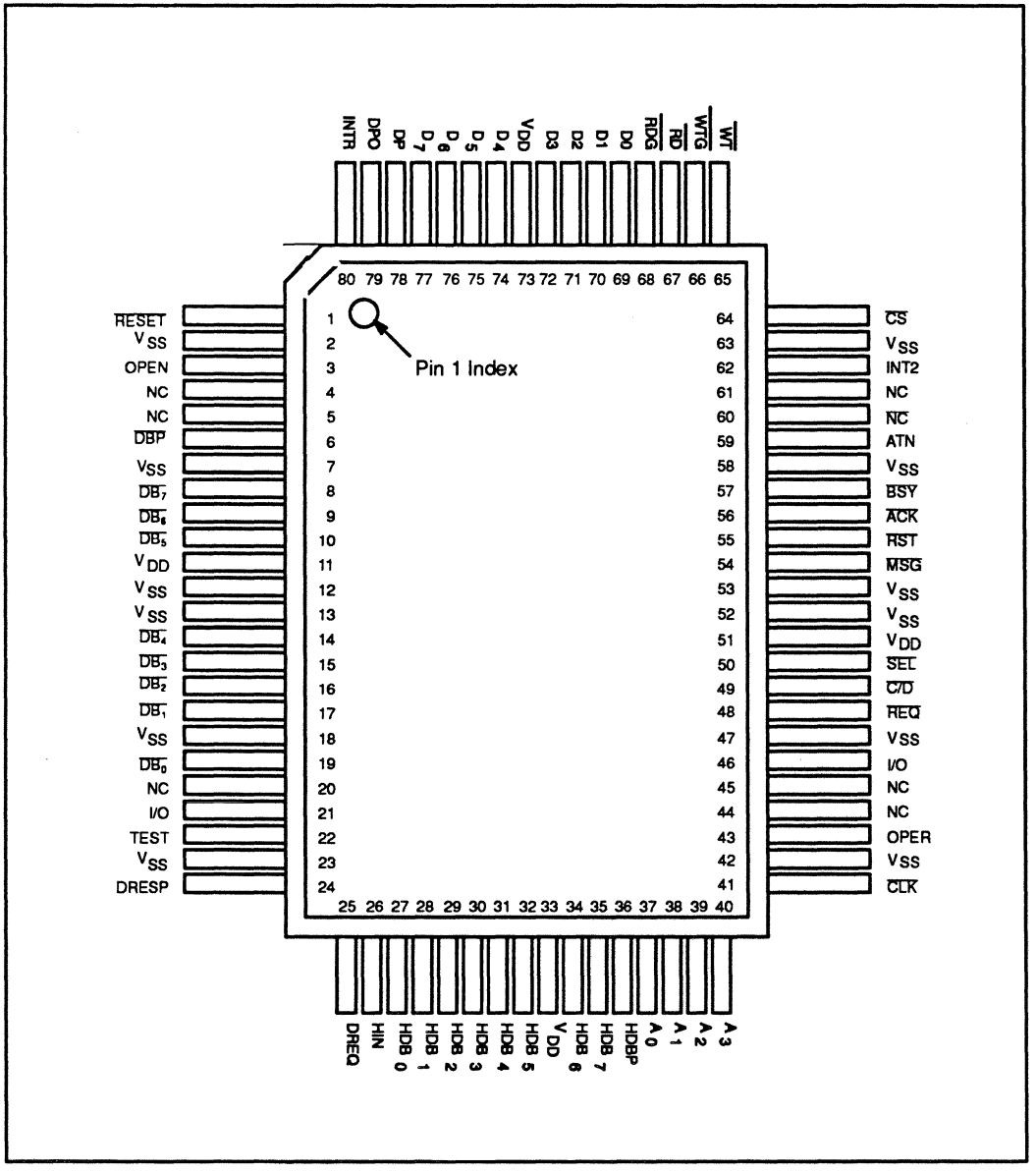
1. Permanent device damage may occur if the above **Absolute Maximum Ratings** are exceeded. Functional operation should be restricted to the conditions as detailed in the operational sections of this data sheet. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.
2. V_{SS} = 0V.
3. Not more than one output may be shorted at a time for a maximum duration of one second.

MB87033 BLOCK DIAGRAM



Transfer PIN ASSIGNMENTS

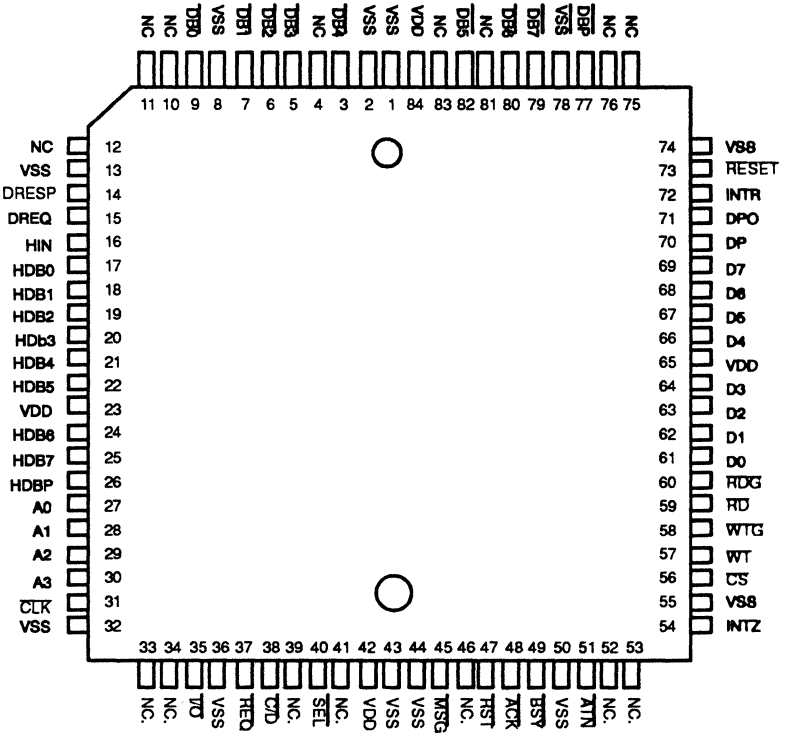
80–Pin Plastic Flat Package



PIN ASSIGNMENTS

84–Pin Plastic Leadless Chip Carrier

(Top View)



PIN DESCRIPTIONS

Pin Number		Designator	Function
FPT	PLCC		
1	73	$\overline{\text{RESET}}$	When set to the active-low state, an asynchronous reset signal that clears all internal circuits of the SPC.
2.7, 12,13 18,23 42,47 52,53 58,63	1,2 8,13 32,36 43,44 50,55 74,78	V_{SS}	Power supply ground.
3,43	—	Open	Reserved. (Note: Do not make external connections to these pins.)
4,5 20,21 44,45 60,61	4,10 11,12 33,34 39,41 46,52 53,75 76,81,83	NC	No internal connection. (Note: These pins must be open connections at all times.)
8 9 10 14 15 16 17 19 6	79 80 82 3 5 6 7 9 77	$\overline{\text{DB7}}$ $\overline{\text{DB6}}$ $\overline{\text{DB5}}$ $\overline{\text{DB4}}$ $\overline{\text{DB3}}$ $\overline{\text{DB2}}$ $\overline{\text{DB1}}$ $\overline{\text{DB0}}$ $\overline{\text{DBP}}$	Inputs/outputs for the SCSI data bus; these I/O pins connect directly to the SCSI connector. $\overline{\text{DB7}}$ is the MSB; $\overline{\text{DB0}}$ is the LSB. $\overline{\text{DBP}}$ is an odd parity bit.
11,33 51,73	2,21,22,42 44,63,65,84	VDD	+5V power supply.
22		TEST	Reserved for test functions. (Note. Do not make external connections to these pins.)
24	14	DRESP	DRESP is a response signal to the data transfer request signal DREQ. The DRESP pin must be refreshed with an applied pulse after each byte of data is transferred. In the DMA mode, DRESP is used as a timing signal for completion of a data-byte transfer.

PIN DESCRIPTIONS

Pin Number		Designator	Function
FPT	PLCC		
25	15	DREQ	When executing a data transfer cycle in the DMA mode, DREQ is used to indicate a request for a data transfer between the SPC and external buffer memory. In the DMA mode, routing of data is as shown below. **Output Operation:** External buffer memory to SPC to SCSI bus. **Input Operation:** SCSI bus to SPC to external buffer memory. In an output operation, DREQ becomes active to request a data transfer to the external buffer memory when the FIFO contains valid data.
26	16	HIN	Indicates direction of transmission along data bus lines HDB0-HDB7 and HDBP in the DMA transfer mode. To be executed, direction of transmission must be properly coordinated with internal operation of the SPC. When HIN is low (inactive), the data bus lines are placed in the high-impedance state (input mode). When HIN is high (active) all bus lines are switched to the output mode.
35 34 32 31 30 29 28 27 36	25 24 22 21 20 19 18 17 26	HDB7 HDB6 HDB5 HDB4 HDB3 HDB2 HDB1 HDB0 HDBP	3-state bidirectional data bus for transferring data to or from the external buffer memory in the DMA mode. As shown below, the direction of data transmission depends on the HIN input signal. **HIN** **HDBn** **Operation** L Input Mode Output H Output Mode Input
37–40	30–27	A3–A0	Address input signals for selecting an internal register in the SPC. The MSB is A3; the LSB is A0. When $\overline{CS}$ is active low, read/write is enabled and an internal register is selected by these address inputs via data bus lines D0-D7 and DP.
41	31	CLK	Input clock for controlling internal operations and data-transfer speeds of SPC.

PIN DESCRIPTIONS (Continued)

Pin Number		Designator	Function
FPT	PLCC		
57 50 48 56 54 49 46 59 55	49 40 37 48 45 38 35 51 47	$\overline{BSY}$ $\overline{SEL}$ $\overline{REQ}$ $\overline{ACK}$ $\overline{MSG}$ $\overline{C/D}$ $\overline{I/O}$ $\overline{ATN}$ $\overline{RST}$	Input/output control signals for SCSI bus.
62	54	INT2	A non-maskable interrupt request signal that indicates a reset condition on the SCSI bus.
64	56	CS	Selection enable signal for accessing an internal register in the SPC. When CS is active low, the following signals are valid: $\overline{RD}$, $\overline{WT}$, A0–A3, D0–D7 and DP.
65	57	$\overline{WT}$	Input strobe used for writing data into an internal register of the SPC; this signal is asserted only if $\overline{CS}$ is active low. On the trailing edge of $\overline{WT}$, data placed on data bus lines D0-D7/DP is loaded into the internal register selected by address inputs A0-A3, except when all address lines are high (A0-A3 = H).
66	58	$\overline{WTG}$	While this signal is active low, data placed on data bus lines D0-D7/DP is output to HDB0-HDB7/HDBP provided the following input conditions are satisfied: CS and HIN = H $\overline{A0}$–A3 = H
67 68	59 60	$\overline{RD}$ $\overline{RDG}$	Input strobes used for reading out contents of internal register; strobes are effective only when $\overline{CS}$ is active low. When $\overline{RDG}$ is active low, the contents of an internal register selected by address inputs A0–A3 are placed on data bus lines D0–D7/DP. For a data transfer cycle in the program transfer mode, the trailing edge of RD is used as a timing signal to indicate the end of data read.

PIN DESCRIPTIONS (Continued)

Pin Number		Designator	Function
FPT	PLCC		
69 70 71 72 74 75 76 77	61 62 63 64 66 67 68 69	D0 D1 D2 D3 D4 D5 D6 D7	Used for writing or reading data into or from an internal register of the SPC; these bus lines are 3-state and bidirectional. The MSB is D7; the LSB is D0. DP is an odd parity bit.
78	49	DP	When the $\overline{CS}$ and $\overline{RDG}$ inputs are active low, contents of the internal register are output to the data bus (read operation). In operations other than read, these bus lines are kept in a high-impedance state.
79	50	DPO	An odd parity output for data byte D7–D0. DPO represents an output when D7–D0/DP are placed in a high-impedance state; DPO is in a high-impedance state when D7–D0/DP serve as outputs. If a parity bit is not generated for external memory, DPO can be used as an input for DP.
80	51	INTR	When active high, requests an interrupt to indicate completion of an SPC internal operation or the occurrence of an error; the INTR interrupt request can be masked by user software. When an interrupt request is honored, INTR remains in the active state until cause of the interrupt is cleared.

ADDRESSING OF INTERNAL REGISTERS

The MB87033 SPC contains16 byte-wide registers that are externally accessible. These registers are used to control internal operations of the SPC and also to indicate processing/result status. A unique address, identified by address bits A3–A0, is assigned to each of the sixteen registers. These addresses are defined in Table 1.

Table 1. Internal Register Addressing

Register	Mnemonic	Operation	Chip Select ($\overline{CS}$)	Address Bits A3	A2	A1	A0
Bus Device ID	BDID	R W	0	0	0	0	0
SPC Control	SCTL	R W	0	0	0	0	1
Command	SCMD	R W	0	0	0	1	0
Transfer Mode	TMOD	R W	0	0	0	1	1
Interrupt Sense Reset Interrupt	INTS	R W	0	0	1	0	0
Phase Sense SPC Diagnostic Control	PSNS SDGC	R W	0	0	1	0	1
SPC Status	SSTS	R	0	0	1	1	0
SPC Error Status	SERR	R	0	0	1	1	1
Phase Control	PCTL	R W	0	1	0	0	0
Modified Byte Count Extended Transfer Count	MBC	R W	0	1	0	0	1
Data Register	DREG	R W	0	1	0	1	0

Register	Mnemonic	Operation	Chip Select ($\overline{CS}$)	Address Bits A3	A2	A1	A0
Bus Device ID	BDID	R	0	0	0	0	0
		W					
SPC Control	SCTL	R	0	0	0	0	1
		W					
Command	SCMD	R	0	0	0	1	0
		W					
Transfer Mode	TMOD	R	0	0	0	1	1
		W					
Interrupt Sense	INTS	R	0	0	1	0	0
Reset Interrupt		W					
Phase Sense	PSNS	R	0	0	1	0	1
SPC Diagnostic Control	SDGC	W					
SPC Status	SSTS	R	0	0	1	1	0
SPC Error Status	SERR	R	0	0	1	1	1
Phase Control	PCTL	R	0	1	0	0	0
		W					
Modified Byte Count	MBC	R	0	1	0	0	1
Extended Transfer Count		W					
Data Register	DREG	R	0	1	0	1	0
		W					

Table 1. Internal Register Addressing (Continued)

Register	Mnemonic	Operation	Chip Select ($\overline{CS}$)	Address Bits A3	A2	A1	A0
Temporary Register	TEMP	R	0	1	0	1	1
		W					
Transfer Counter (High)	TCH	R	0	1	1	0	0
		W					
Transfer Counter (Middle)	TCM	R	0	1	1	0	1
		W					
Transfer Counter (Low)	TCL	R	0	1	1	1	0
		W					
External Buffer	EXBF	R	0	1	1	1	1
		W					

BIT ASSIGNMENTS FOR INTERNAL REGISTERS

Table 2 lists the bit assignments for the sixteen internal registers defined in Table 1. In most cases, bit assignments for the MB87033 SPC are identical to those for the MB87030/31; however, in the MB87033, some features are expanded and others are added to improve overall performance. These modifications and additions are summarized in the following descriptions.

MPU Bus Parity Generator: An odd parity bit is output from DPO (pin 79 in FPT, pin 50 in PLCC) for each data byte (D7-D0). DPO is a 3-state pin and is placed in a high-impedance state when data from D7-D0 is output to the MPU. If the MPU interface does not contain a parity generator, the output of DPO can be connected to the DP input pin of the SPC (pin 78 in FPT, pin 49 in PLCC).

Reset Condition Interrupt Request Signal

The INT2 output (pin 62 in FPT, pin 33 in PLCC) is a non-maskable interrupt request that, when driven High, notifies the SPC when a reset condition is detected on the SCSI bus. Bit 4 (Reset Condition Interrupt Mask Enable) of the Phase Control (PCTL) register does not affect the INT2 output pin.

When a bus reset condition is detected, the INTR output (pin 80 in FPT, pin 51 in PLCC) also is driven to the high state; however, the state of INTR can be masked by bit 4 of the PCTL register:

Bit 4 = 0: INTR goes high when a reset condition is detected.

Bit 4 = 1: INTR does not go high when a reset condition is detected.

Lost Arbitration Interrupt Request

If bit 6 (Lost Arbitration Interrupt Enable) of the phase control (PCTL) register is set to "1", a COMMAND COMPLETE interrupt is generated when the SPC (serving as initiator or target) loses in the ARBITRATION process. To determine the cause of a COMMAND COMPLETE interrupt (completion of SELECTION, RESELECTION, or lost ARBITRATION), refer to bits 6 (target) and 7 (initiator) of the SPC status (SSTS) register. If both bits are set to "0", the COMMAND COMPLETE interrupt is a result of lost arbitration.

Attention Condition Interrupt

If bit 5 (Attention Condition Interrupt Enable) of the phase control (PCTL) register is set to "1" and the SPC serves as a target, a service required interrupt occurs. To reset the service

required interrupt, set bit 3 of the interrupt sense (INTS) register to "1" or revoke the current target role of the SPC.

Expansion of Transfer Byte Counter

If bit 0 of the transfer mode (TMOD) register is set to "1", the transfer byte counter is expanded to 28-bits. In the expanded mode, the high nibble (bits 24 through 27) are entered into the four most significant bit positions (7 through 4) of the modified byte count (MBC) register.

Note: When a hardware data transfer or execution of a SELECT command is in process, access to the TMOD register is forbidden.

Bit 0 of the TMOD register =1: To access the highest four bits (bits 27 through 24) of the transfer byte counter, data reads or writes are addressed to the high nibble of the modified byte counter (MBC) register. When a TRANSFER or SELECT command is issued, the transfer byte count (or t_{WAIT}) should be placed in the high nibble of the MBC register rather than the TCH, TCM, and TCL registers.

Bit 0 of the TMOD register = 0: The transfer byte counter is not expanded to 28–bits; hence, reading the high nibble of MBC yields a "0" even though some particular value is written into the register. In this case, t_{WAIT} or the transfer byte count is based on a 24-bit transfer byte counter (identical to the MB87030/MB87031).

During read/write access of an internal register, the following rules are invoked:

- Internal registers include only those registers identified in Table 2.
- A write command to a read–only register is ignored.
- For write operations, all bit positions with a "—" (blank) designator can be written as "0" or "1".
- All bit positions with an assigned "0" are always read as a zero (0).

Table 2. Bit Assignments For Internal Registers

HEX Address	Register and Mnemonic	R/W Operation	7 (MSB)	6	5	4	3	2	1	0 (LSB)	Parity
0	Bus Device ID (BDID)	R	#7	#6	#5	#4	#3	#2	#1	#0	0
		W	SCSI Bus Device ID					ID4	ID2	ID1	—
1	SPC Control (SCTL)	R	Reset & Disable	Control Reset	Diag Mode	ARBIT Enable	Parity Enable	Select Enable	Reselect Enable	INT Enable	P
		W									
2	Command (SCMD)	R	Command Code			RST Out	Intercept Xfer	Transfer PRG Xfer	Modifer 0	Term Mode	P
		W									
3	Transfer Mode (TMOD)	R	Sync. Xfer	Max. Transfer Offset			Min. Transfer Period			Xfer Counter Expand	P
		W		4	2	1	2	1	0		
4	Interrupt Sense (INTS)	R	Selected	Reselected	Disconnect	Command Complete	Service Required	Time Out	SPC Hard Error	Reset Condition	P
		W	Reset Interrupt								—
5	Phase Sense (PSNS)	R	REQ	ACK	ATN	SEL	BSY	MSG	C/D	I/O	P
	SPC Diag Control (SDGC)	W	Diag REQ	Diag ACK	—	—	Diag BSY	Diag MSG	Diag C/D	Diag I/O	—
6	SPC Status (SSTS)	R	Connected		SPC	XFER In Progress	SCSI	TC=0	DREG	Status	P
			INIT	TARG	BSY		RST		Full	Empty	
		W	—								—
7	SPC Error Status (SERR)	R	Data Error		0	0	TC Parity Error	Phase Error	Short Period	Offset Error	P
			SCSI	SPC							
		W	—								—

Table 2. Bit Assignments For Internal Registers (Continued)

HEX Address	Register and Mnemonic	R/W Operation	7 (MSB)	6	5	4	3	2	1	0 (LSB)	Parity
8	Phase Control (PCTL)	R	Bus Free Interrupt Enable	Arbitration Fail Interrupt Enable	Attention Condition Interrupt Enable	Reset Condition Interrupt mask	0	Transfer Phase			P
		W						MSG Out	C/D Out	I/O Out	
9	Modified Byte Counter (MBC)	R	Extended Transfer Counter				MBC				
			Bit 27	Bit 26	Bit 25	Bit 24	Bit3	Bit2	Bit1	Bit0	
A	Data Register (DREG)	R	Internal Data Register (8 Byte FIFO)								P
		W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
B	Temporary Register (TEMP)	R	Temporary Data (Input: From SCSI)								P
			Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
		W	Temporary Data (Output: to SCSI)								P
			Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
C	Transfer Counter High (TCH)	R	Transfer Counter High (MSB)								P
		W	Bit23	Bit22	Bit21	Bit20	Bit19	18Bit	17Bit	16Bit	
D	Transfer Counter Mid. (TCM)	R	Transfer Counter High (2nd Byte)								P
		W	Bit15	14Bit	13Bit	12Bit	11Bit	10Bit	9Bit	8Bit	
E	Transfer Counter Low (TCL)	R	Transfer Counter High (LSB)								P
		W	Bit7	6Bit	5Bit	4Bit	3Bit	2Bit	1Bit	0Bit	
F	External Buffer (EXBF)	R	External Buffer							P	
		W	Bit7	6Bit	5Bit	4Bit	3Bit	2Bit	1Bit	0Bit	

DC CHARACTERISTICS
(Recommended operating conditions unless otherwise specified)

SCSI Bus Pins

Parameter	Designator	Condition	Values			Unit
			Min.	Typ.	Max.	
Input Voltage	V_{IH}		2.0		5.25	V
	V_{IL}		0.0		0.8	V
Input Current	I_{IH}	V_{IH} = 5.25V			10	mA
	I_{IL}	V_{IL} = 0.0V			–10	mA
Output Voltage	V_{OL}	V_{DD} = 4.75V I_{OL} = 48 mA			0.5	V
Input Hysteresis Width	V_{HW}		0.2			V

Other than SCSI Bus Pins

Parameter	Designator	Condition	Values			Unit
			Min.	Typ.	Max.	
Input Voltage	V_{IH}		2.2		V_{DD} + 0.3	V
	V_{IL}		V_{SS}–0.3		0.8	V
Output Voltage	V_{OH}	I_{OH} = –0.4mA	4.2		V_{DD}	V
	V_{OL}	I_{OL} = 3.2mA	V_{SS}		0.4	
Input Leakage Current	I_{LIN}	V_{IH} = 5.25V			10	mA
	I_{LIN}	V_{IL} = 0.0V			–10	mA
Input/Output Leakage Current	I_{LIN}	V_{IH} = 5.25V			10	mA
	I_{LIN}	V_{IL} = 0.0V			–10	mA
Supply Current	I_{DD}	10 MHz Clock Outputs Open			30	mA

TA = 0 – 70°C, V_{DD} + 5V ± 5%

DC CHARACTERISTICS (Continued)
(Recommended operating conditions unless otherwise specified)

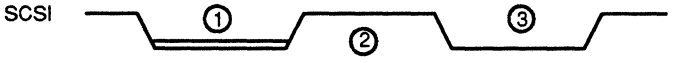
Capacitance

Parameter	Designator	Condition	Values			Unit
			Min.	Typ.	Max.	
Input pin capacitance	C_{IN}	T_A = 25C, V_{DD}= V_I=OV, f – 1MHz			9	pf
Output capacitance	C_{OUT}	T_A = 25C, V_{DD}= V_I=OV, f – 1MHz			9	pf
I/O pin capacitance *2	$C_{I/O}$	T_A = 25C, V_{DD}= V_I=OV, f – 1MHz			11	pf
I/O pin capacitance *3	$C_{I/O}$	T_A = 25C, V_{DD}= V_I=OV, f – 1MHz			30	pf

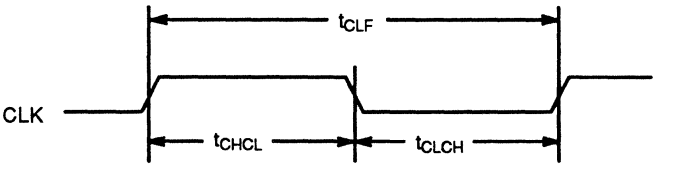
Notes: 1. SCSI bus Pins are $\overline{DB7}$–$\overline{DB0}$, $\overline{DBP}$, $\overline{RST}$, $\overline{SEL}$, $\overline{I/O}$, $\overline{C/D}$, $\overline{MSG}$, $\overline{ATN}$, $\overline{REQ}$, $\overline{ACK}$, and $\overline{BSY}$.
2. For all I/O pins except SCSI bus pins.
3. For SCSI bus pins only (see note 1.)

AC CHARACTERISTICS

SCSI signal timing chart is described as follows:

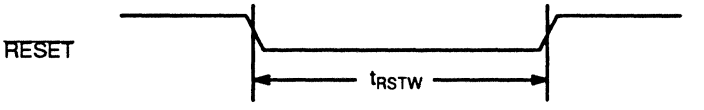


Notes: 1. Output "L"
2. No device is outputting "L".
3. The other device is outputting "L".



Clock

Parameter	Designator	Values			Unit
		Min	Typ	Max	
Clock cycle time	t_{CLF}	100		200	ns
Clock "H" Pulse width	t_{CHCL}	50			ns
Clock "L" Pulse width	t_{CLCH}	40			ns

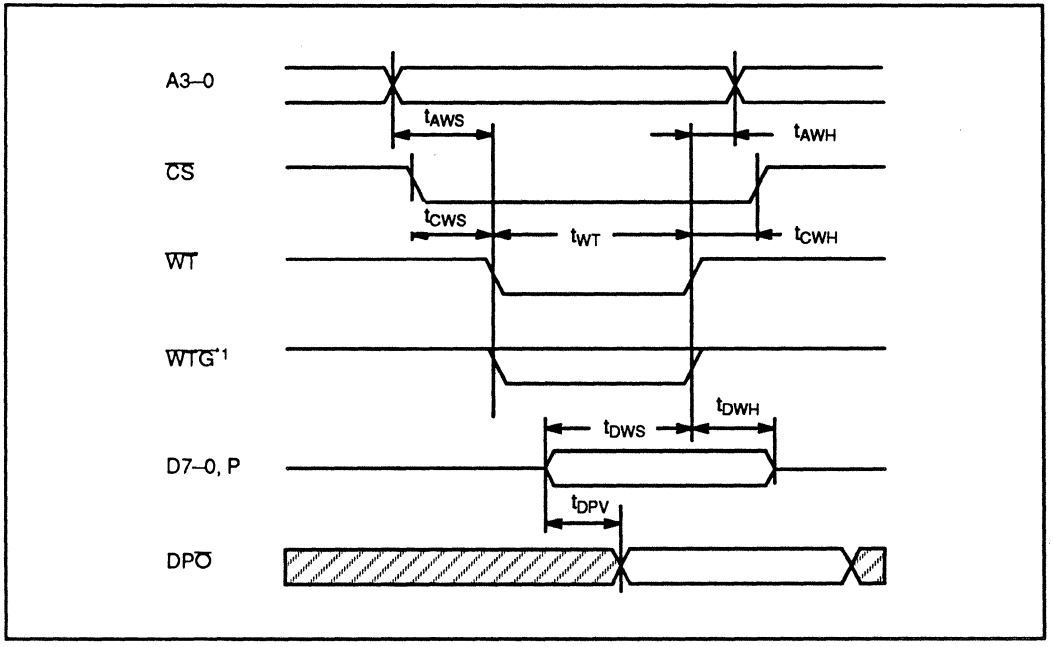


Hardware Reset

Parameter	Designator	Values			Unit
		Min	Typ	Max	
Reset Pulse Width	t_{RSTW}	50			ns

AC CHARACTERISTICS (Continued)

Register Write (except EXBF)



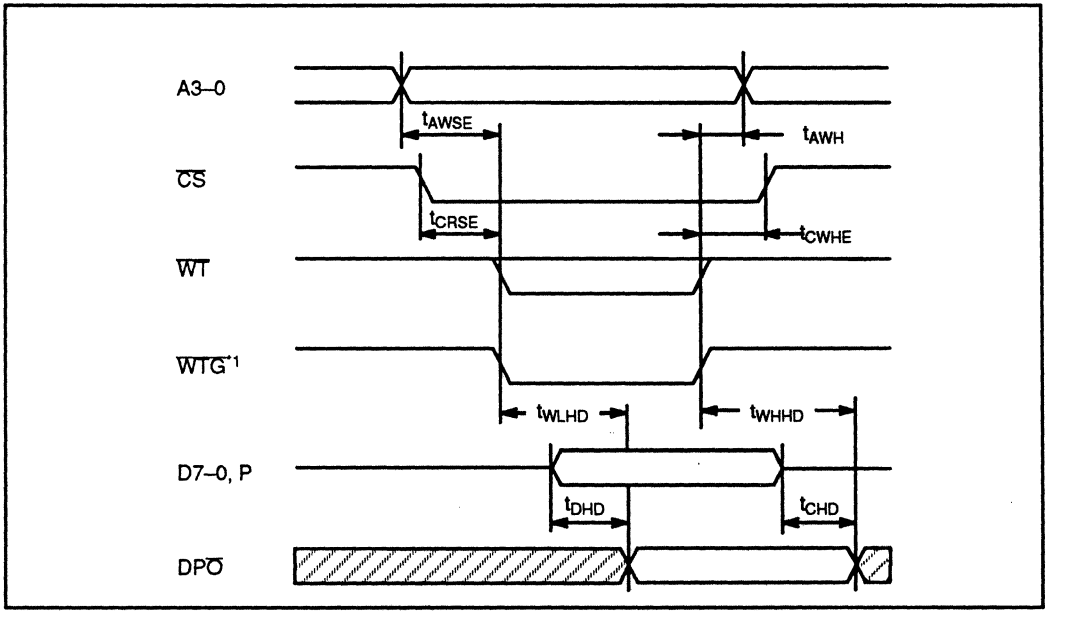
Register Write

Parameter	Designator	Values			Unit
		Min	Typ	Max	
Address set-up	t_{AWS}	35			ns
Address hold	t_{AWH}	5			ns
$\overline{CS}$ set-up	t_{CWS}	20			ns
$\overline{CS}$ hold	t_{CWH}	10			ns
Data bus set-up	t_{DWS}	25			ns
Data bus hold	t_{DWH}	20			ns
$\overline{WT}$ Pulse width	t_{WT}	50			ns
Data bus valid → DPO valid	t_{DPV}			55	ns

Note: *1 $\overline{WTG}$ is input at the same timing as $\overline{WT}$ or held to "H".

AC CHARACTERISTICS (Continued)

Register Write (EXBF)



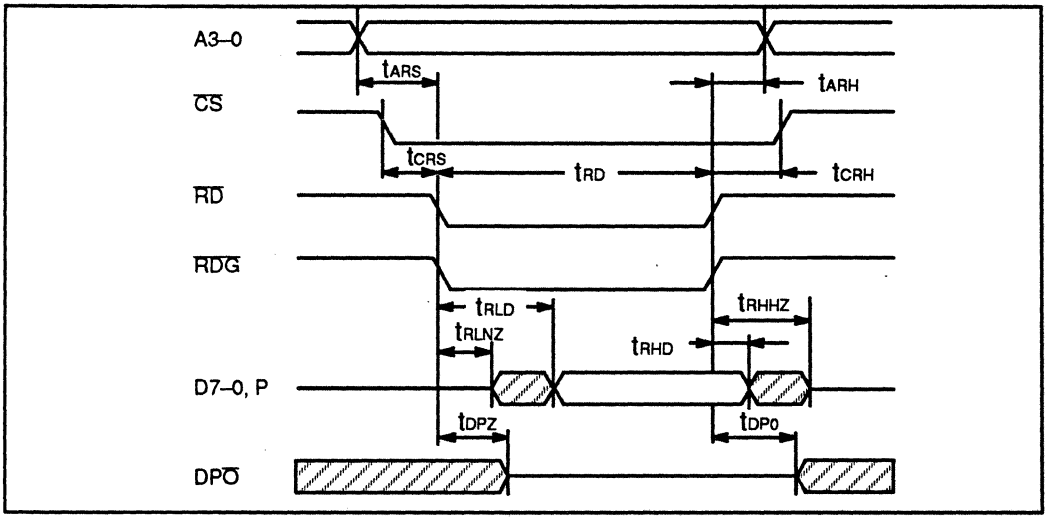
Register Write

Parameter	Designator	Values			Unit
			Min	Typ	Max
Address set-up	t_{aws}	35			ns
Address hold	t_{awhe}	5			ns
$\overline{CS}$ set-up	t_{cwse}	20			ns
$\overline{CS}$ hold	t_{cwhe}	10			ns
$\overline{WTG}$ "L" → DMA pulse output valid	t_{wlhd}			55	ns
$\overline{WTG}$ "H" → DMA pulse output invalid	twhhd	10			ns
MPU data bus → DMA bus delay	t_{dhd}			50	ns

Note: *1 $\overline{WTG}$ is input at the same timing as $\overline{WT}$ or held to "H".

AC CHARACTERISTICS (Continued)

Register Read (except EXBF)



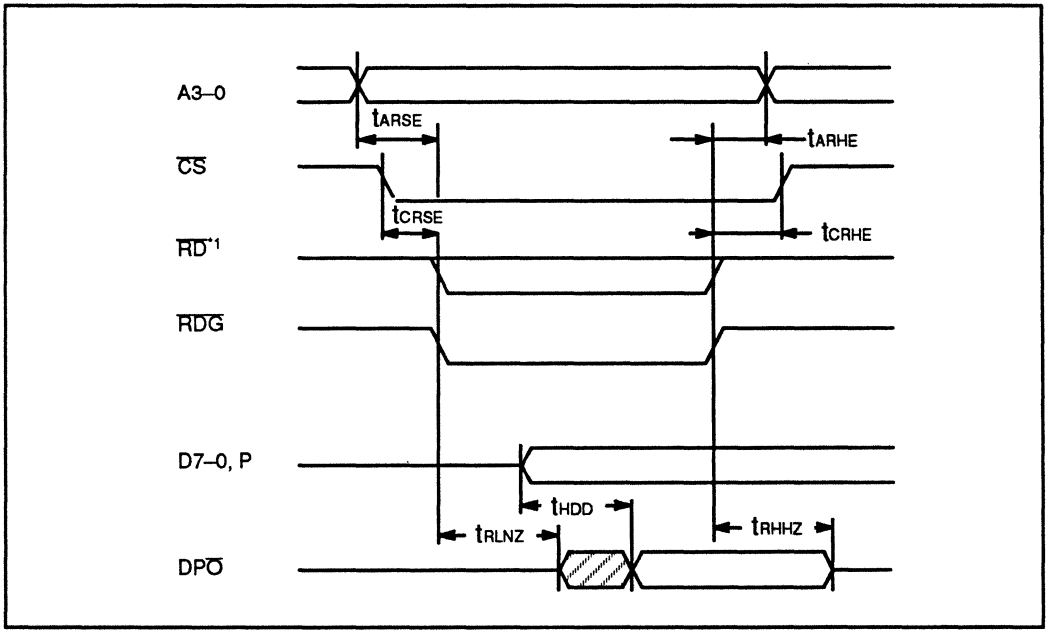
Register Write

Parameter		Designator	Values			Unit
			Min	Typ	Max	
Address set-up		t_{ARS}	35			ns
Address hold		t_{ARH}	5			ns
$\overline{CS}$ set-up		t_{CRS}	20			ns
$\overline{CS}$ hold		t_{CRH}	10			ns
$\overline{RDG}$ "L" → Data bus ourpur valid		t_{RLNZ}	10		40	ns
$\overline{RDG}$ "H" → Data bus ourpur valid		t_{RHHZ}	10		40	ns
$\overline{RD}$ "L" → Data Valid	D7–0	t_{RLD}			70	ns
	DP				85	ns
$\overline{RD}$ "H" → Data Invalid		t_{RHD}	10			ns
$\overline{RD}$ Pulse width		t_{RD}	50			ns
*1 $\overline{RDG}$ "L" → Data High-Z		t_{DPZ}	10		40	ns
*1 $\overline{RDG}$ "H" → DPO		t_{DPO}	10		40	ns

Note: *1 = DPO goes to High-Z when both $\overline{RDG}$ and $\overline{CS}$ are "L".

AC CHARACTERISTICS (Continued)

Register Read (EXBF)



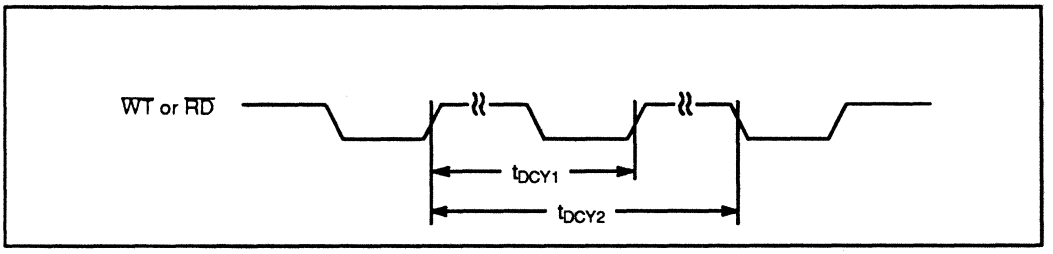
Note: *1:: RD is input at the same timing as RDG or held to "H".

Register Read

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
Address set-up	t_{ARSE}	35			ns
Address hold	t_{ARHE}	5			ns
CS set-up	t_{CRSE}	20			ns
CS hold	t_{CRHE}	10			ns
RDG "L" → Data bus output	t_{RLNZ}	10		40	ns
RDG "H" → Data bus High-Z	t_{RHHZ}	10		40	ns
DMA bus → MPU data bus delay	t_{HDD}			50	ns

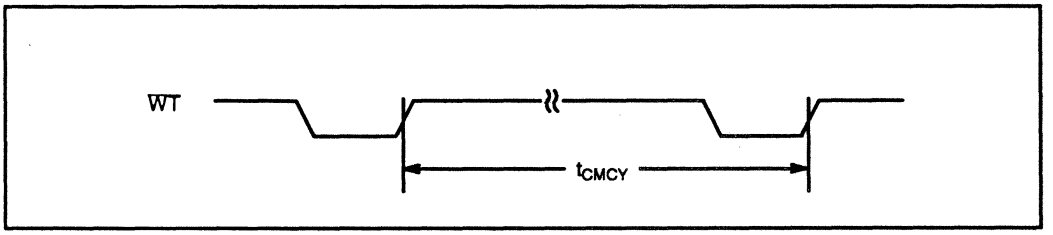
AC CHARACTERISTICS (continued)

DREG Access Cycle Time



DREG

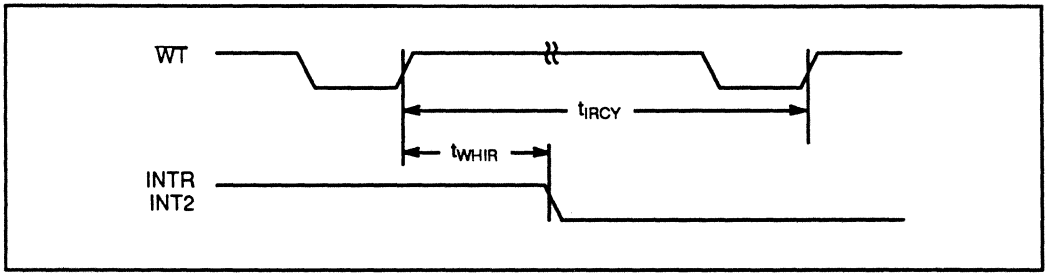
Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
DREG access cycle time (2)	t_{DCY1}	$2t_{CLF}$			ns
DREG access cycle time (1)	t_{DCY2}	$3t_{CLF}$			ns



Command Issue Cycle Time

Parameter	Designator	Values			Unit
		Min.	Typ..	Max	
SCMD register access cycle time	t_{CMCY}	$4t_{CLF}$			ns

AC CHARACTERISTICS (continued)

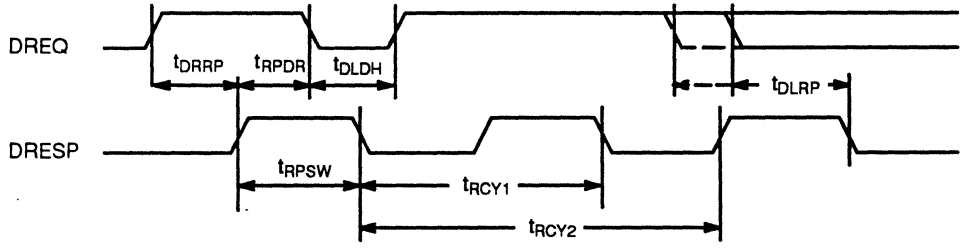


Interrupt Reset

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
$\overline{WT}$ "H" → Interrupt output (INTR.INT2) "L"	t_{WHR}	t_{CLF}		$4t_{CLF} + 80$	ns
INTS register access cycle time	t_{IRCY}	$4t_{CLF}$			ns

AC CHARACTERISTICS (continued)

DMA Access Cycle Time

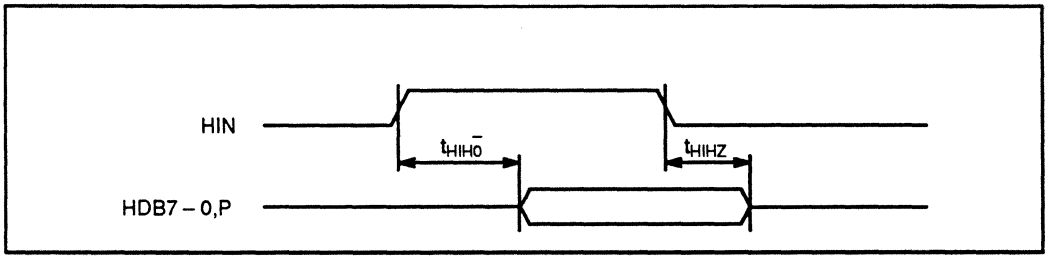


DMA Access Timing

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
DREQ "H" → DRESP "H"	t_{DRRP}	t_{CLF}			ns
DRESP "H" → DREQ "L"	t_{RPDR}	5		70	ns
DREQ "L" → DREQ "H"	t_{DLDH}	0			ns
DRESP Pulse Width	t_{RPSW}	50			ns
DRESP Cycle time (1)	t_{RCY1}	$2t_{CLF}$			ns
DRESP Cycle time (2)	t_{RCY2}	$3t_{CLF}$			ns
DREQ "L" → DRESP "L"	t_{DLRP}			$5t_{CLF}$	ns

Note: *1 : This parameter is applicable when the data buffer hold function or the Transfer Pause command is used and DREQ goes low asynchronously to DRESP. Under these conditions, data cannot be written to the data buffer until tDLRP and tRPSW are satisfied.

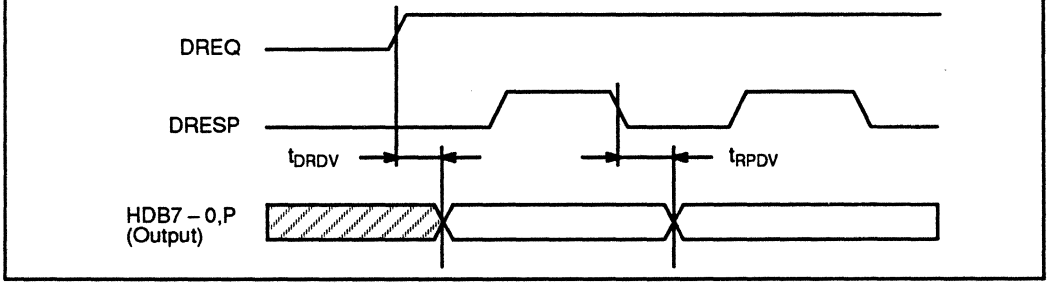
AC CHARACTERISTICS (continued)



DMA Bus Output Control

Parameter	Designator	Values			Unit
		Min	Typ	Max	
HIN "H" → DMA bus output	$t_{HIH\overline{0}}$	5_{CLF}		40	ns
HIN "L" → DMA bus High-Z	t_{HIHZ}	5_{CLF}		40	ns

DMA Transfer (1) – Data Input from SCSI

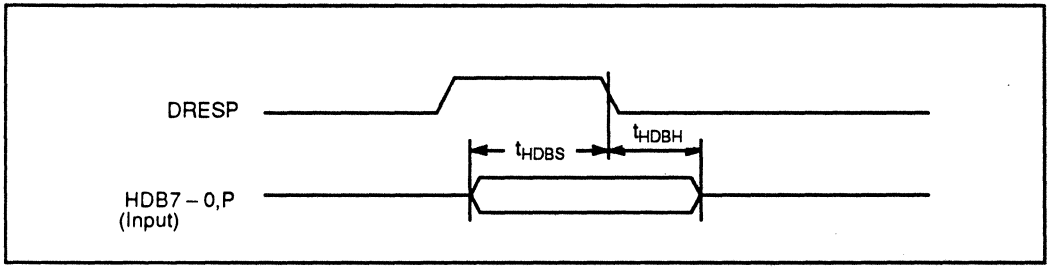


DMA Transfer (1) – Data Input from SCSI

Parameter	Designator	Values			Unit
		Min	Typ	Max	
DREQ "H" → Output data valid	t_{DRDV}			60	ns
DRESP "L" → Output data[1] switch	t_{RPDV}	15		90	ns

Note: *1: This parameter is applied when the internal data buffer goes Not Empty from Empty.

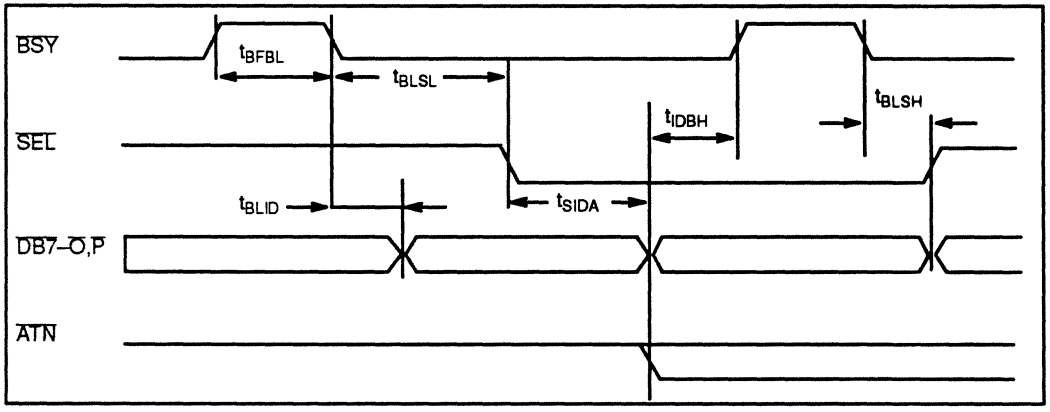
AC CHARACTERISTICS (continued)



DMA Transfer (2) – Data Output to SCSI

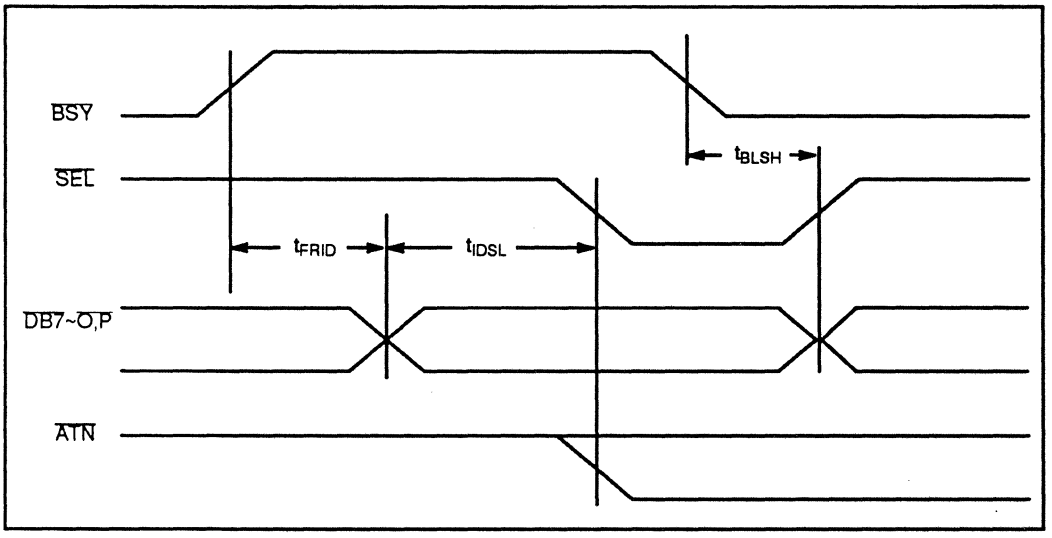
Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
Input data set up	t_{HDBS}	20			ns
Input data hold	t_{HDBH}	20			ns

6.5.14 Selection (1) Initiator (with Arbitration)



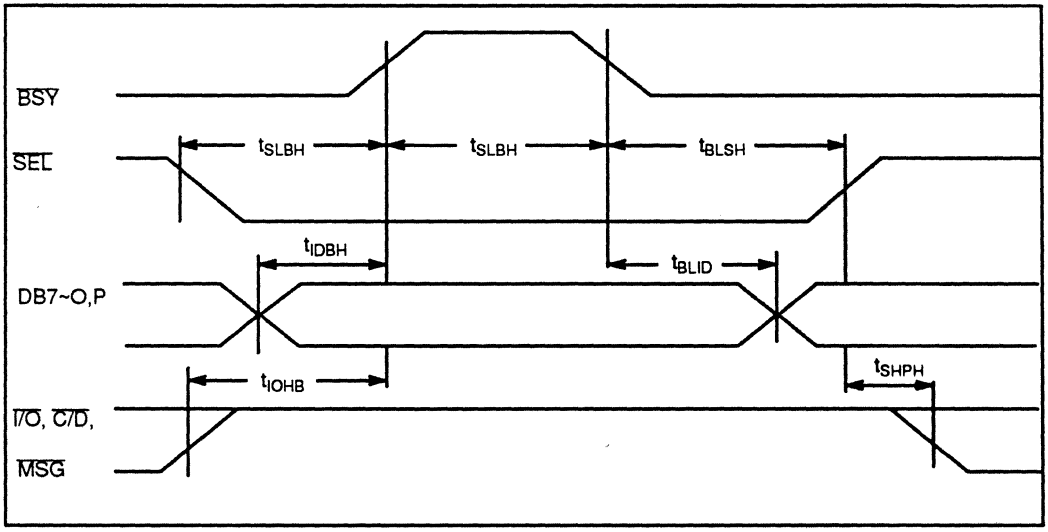
Item	Symbol	Min.	Typ.	Max.	Unit
Bus free→ BSY 'L' *1	t_{BFBL}	(6+n) t_{CLF}		(7+n) t_{CLF}+80	ns
BSY 'L' - sends its own ID bit	t_{BLID}	10		60	ns
BSY 'L' → SEL 'L'	t_{BLSL}	$32t_{CLF}$ – 40		$32t_{CLF}$ + 30	ns
SEL 'L' → ID, ATN send	t_{SIDA}	$11t_{CLF}$ – 30		$11t_{CLF}$ – 60	ns
ID send → BSY 'H'	t_{IDBH}	$2t_{CLF}$ – 60		$2t_{CLF}$ – 30	ns
BSY 'L' → SEL 'H'. ID hold	t_{BLSH}	$2t_{CLF}$		$3t_{CLF}$ + 120	ns
Note *1: n = TCL register setting					

6.5.15 Selection (1) Initiator (with Arbitration)



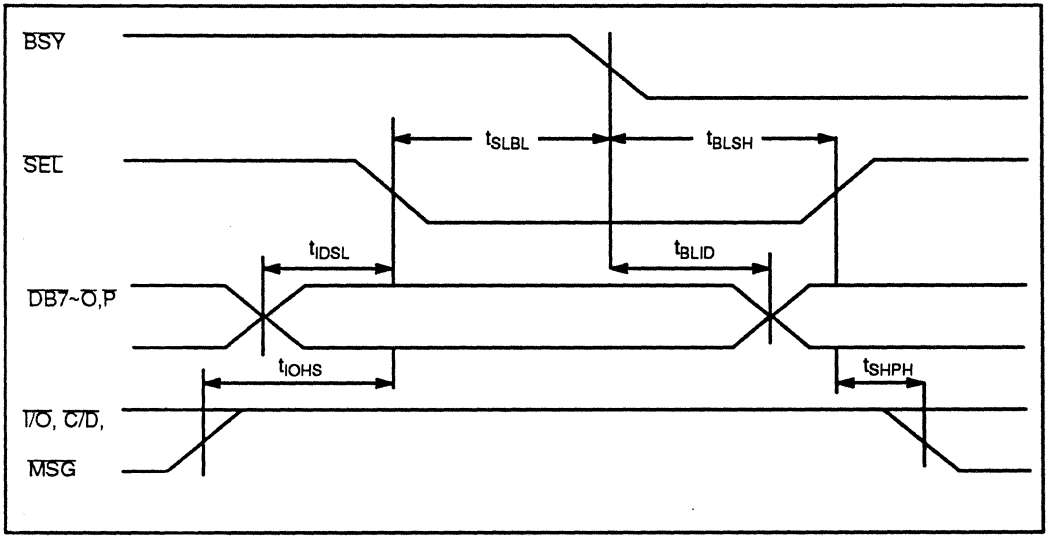
Item	Symbol	Min.	Typ.	Max.	Unit
Bus free → ID send *1	t_{FRID}	(6+n) t_{CLF}		(7+n) t_{CLF} + 100	ns
ID send → SEL 'L', ATN 'L'	t_{IDSL}	11t_{CLF} – 60		11t_{CLF} + 40	ns
BSY 'L'→ SEL 'H'. ID hold	t_{BLSH}	2t $_{CLF}$		3t_{CLF} + 120	ns
Note: *1:n = TCL register setting					

6.5.16 Selection (3) Target (with Arbitration)



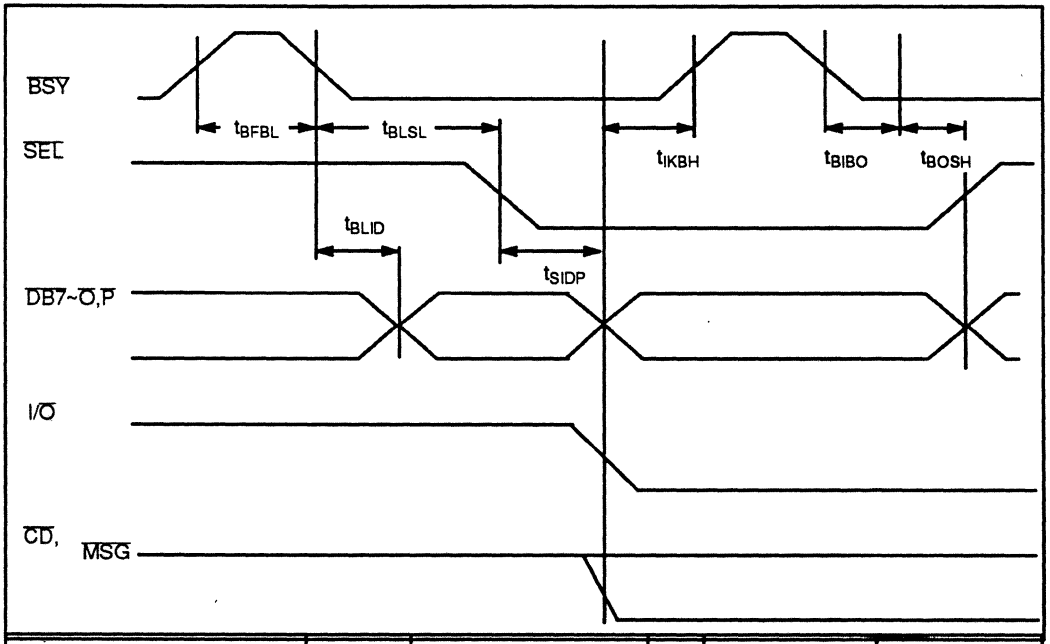
Item	Symbol	Min.	Typ.	Max.	Unit
SEL 'L' → BSY 'H'	t_{SLBH}	0			ns
ID confirmation → BSY 'H'	t_{IDBH}	0			ns
I/O 'H' → BSY 'H'	t_{IOHB}	0			ns
BSY 'H'→ BSY 'L'	t_{BHBL}	$4t_{CLF}$		$5t_{CLF}$ + 80	ns
BSY 'L'→ SEL 'H'	t_{BLSH}	0			ns
BSY 'L' → ID hold	t_{BLID}	30			ns
SEL 'H' phase signal output	t_{SHPH}	$3t_{CLF}$		$4t_{CLF}$ + 100	ns

6.5.17 Selection (4) Target (without Arbitration)



Item	Symbol	Min.	Typ.	Max.	Unit
ID confirmation → SEL 'L'	t_{IDSL}	0			ns
I/O 'H' → SEL 'L'	t_{IOHS}	0			ns
SEL 'L' → BSY 'L'	t_{SLBL}	$2t_{CLF}$		$3t_{CLF}$ + 100	ns
BSY 'L' → SEL 'H'	t_{BLSH}				ns
BSY 'L' → ID hold	t_{BLID}				ns
SEL 'H' → phase signal output	t_{SHPH}	$3t_{CLF}$		$4t_{CLF}$ + 120	ns

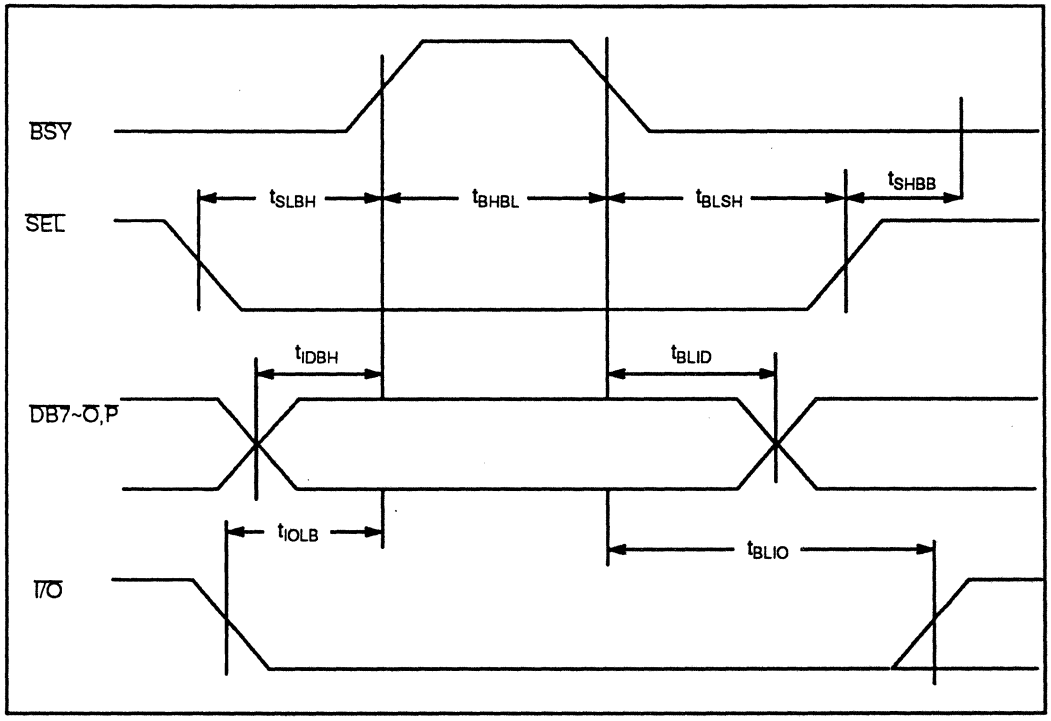
6.5.18 Reselection (1) Target



Item	Symbol	Min.	Typ.	Max.	Unit
Bus free → BSY 'L' *1	t_{BFBL}	(6+n) t_{CLF}		(7+n) t_{CLF} + 80	ns
BSY 'L' – sends its own ID bit	t_{BLID}	10			ns
BSY 'L'→ SEL 'L'	t_{BLSL}	$32t_{CLF}$ – 40		$32t_{CLF}$ + 30	ns
SEL 'L' → ID, phase signal send	t_{SIDP}	$11t_{CLF}$ – 30		$11t_{CLF}$ + 60	ns
ID send → BSY 'H'	t_{IDBH}	$2t_{CLF}$ – 60		$2t_{CLF}$ + 30	ns
BSY 'L'→ BSY 'L' send	t_{BIBO}	$2t_{CLF}$		$3T_{CLF}$ + 90	ns
BSY 'L' send → SEL ID hold	t_{BOSH}	t_{CLF} – 20		t_{CLF} + 60	ns

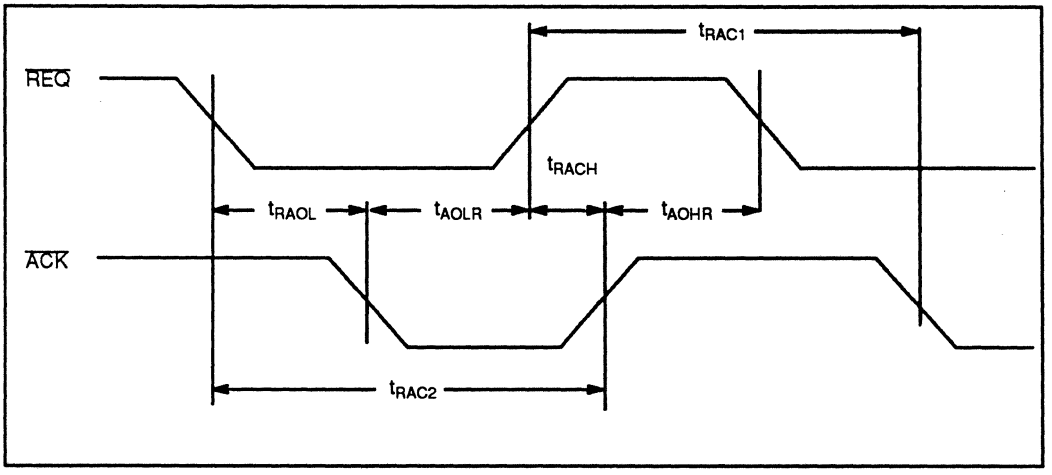
Note: *1: n = TCL register setting

6.5.19 Reselection (2) Initiator



Item	Symbol	Min.	Typ.	Max.	Unit
SEL 'L' → BSY 'H'	t_{SLBH}	0			ns
ID confirmation → BSY 'H'	t_{IDBH}	0			ns
I/O 'L' → BSY 'H'	t_{IOLB}	0			ns
BSY 'H'→ BSY 'L'	t_{BHBL}	$4t_{CLF}$		$5t_{CLF}$ + 80	ns
BSY 'L'→ SEL 'H'	t_{BLSH}	0			ns
BSY 'L' → ID hold	t_{BLID}	30			ns
BSY 'L' → I/O signal hold	t_{BLIO}	20			ns
SEL 'H' → BSY 'L' send stop	t_{SHBO}	$2t_{CLF}$		$3t_{CLF}$ + 100	ns

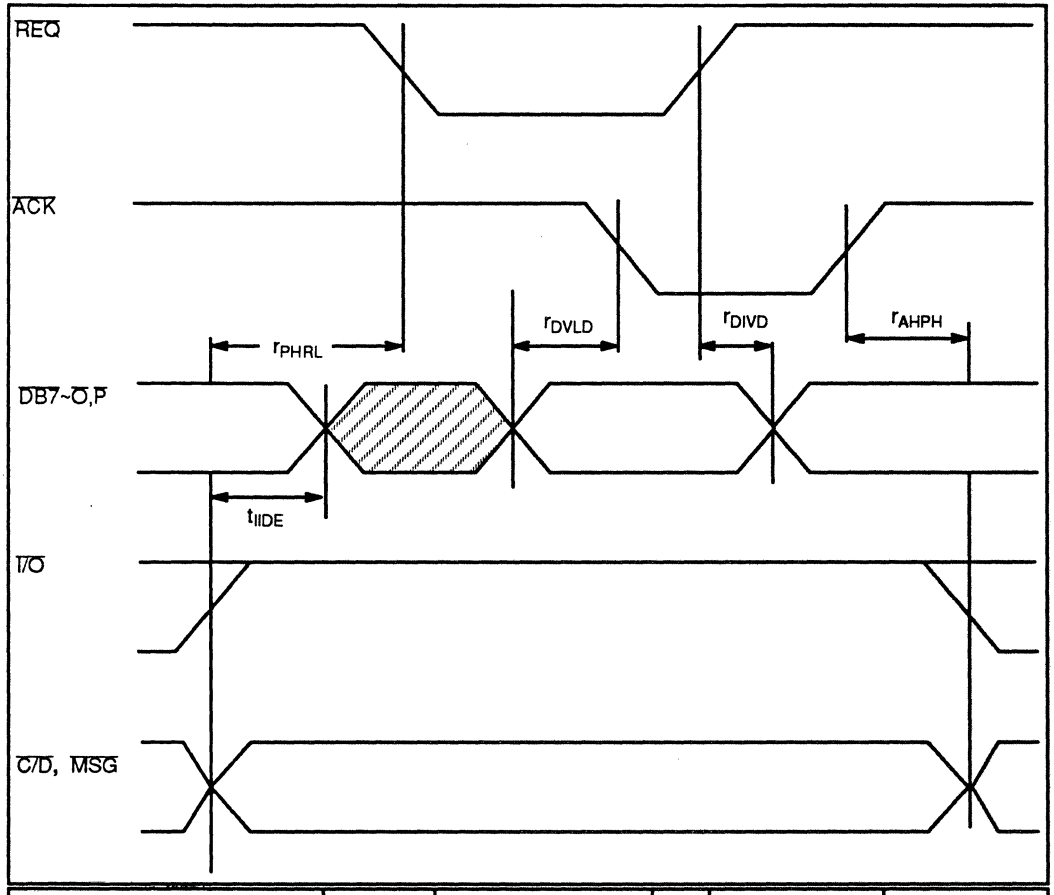
6.5.20 Asynchronous Transfer Initiator (1) REQ-ACK Timing



	Item	Symbol	Min.	Typ.	Max.	Unit
*3	$\overline{REQ}$ 'L' → $\overline{ACK}$ 'L'	t_{RAOL}	15		90	ns
	$\overline{ACK}$ 'L' → $\overline{REQ}$ 'H'	t_{AOLR}	0			ns
*3	$\overline{REQ}$ 'L' → $\overline{ACK}$ 'H'	t_{RACH}	15		100	ns
	$\overline{ACK}$ 'H' → $\overline{REQ}$ 'L'	t_{AOHR}	10			ns
*1,*3	$\overline{REQ}$ 'H' → $\overline{ACK}$ 'L'	t_{RAC1}	$2t_{CLF}$		$3t_{CLF}$ + 120	ns
*1,*3	$\overline{REQ}$ 'H' → $\overline{ACK}$ 'H'	t_{RAC2}	$2t_{CLF}$		$3t_{CLF}$ + 130	ns

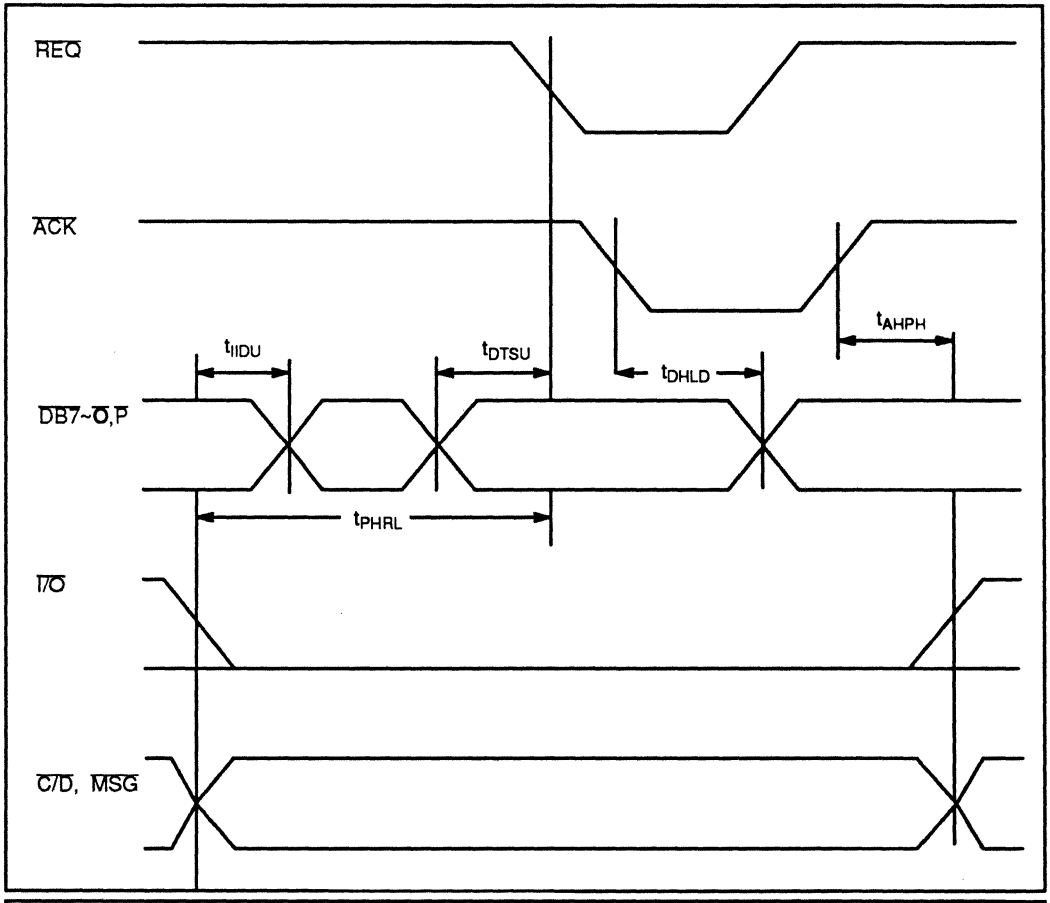
Notes:
*1: The time for $\overline{REQ}$ 'H' → $\overline{ACK}$'L' is determined by the longer of (t_{RACH} + t_{ACHR} + t_{RAOL}) and t_{RAC1}.
*2: Apply for input operations. The time of $\overline{REQ}$'L' → $\overline{ACK}$'H' is determined by the longer of (t_{RAOL} + t_{AOLR} + t_{RACH}) and t_{RAC1}.
*3 The times assigned in this section do not apply in the following cases:
(1) For output operations, if the data buffer is empty.
(2) For input operations, if the data buffer is full.
(3) During transfer of the first or last byte.
(4) During input operations, when the SPC automatically sends the $\overline{ATN}$ signal.

6.5.21 Asynchronous Transfer Initiator (2) Output Operation



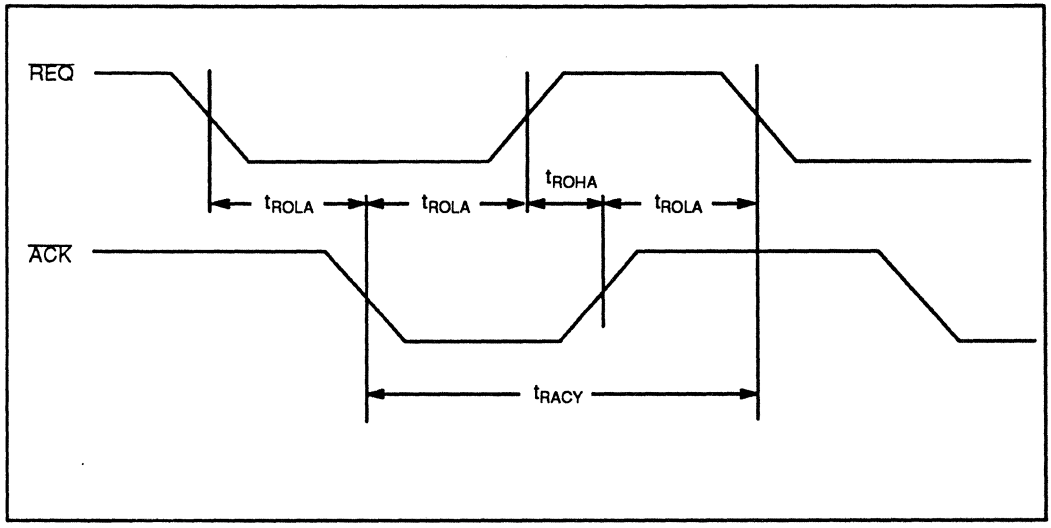
Item	Symbol	Min.	Typ.	Max.	Unit
I/O 'H' → data bus output	t_{IIDE}	10		110	ns
Phase designation → REQ 'L'	t_{PHRL}	100			ns
Data bus output confirmation → ACK 'L'	t_{DVLD}	$2t_{CLF\ 80}$			ns
REQ 'H' → data bus hold	t_{DIVD}	15			ns
ACK 'H' → phase change	t_{AHPH}	10			ns

6.5.22 Asynchronous Transfer Initiator (3) Input Operation



Item	Symbol	Min.	Typ.	Max.	Unit
I/O 'L' → data bus output stop	t_{IIDD}			110	ns
Phase designation → REQ 'L'	t_{PHRL}	100			ns
Data bus confirmation → REQ 'L'	t_{DTSU}	10			ns
ACK 'L' → data bus hold	t_{DHLD}	15			ns
ACK 'H' → phase change	t_{AHPH}	10			ns

6.5.23 Asynchronous Transfer Target (1) REQ-ACK Timing

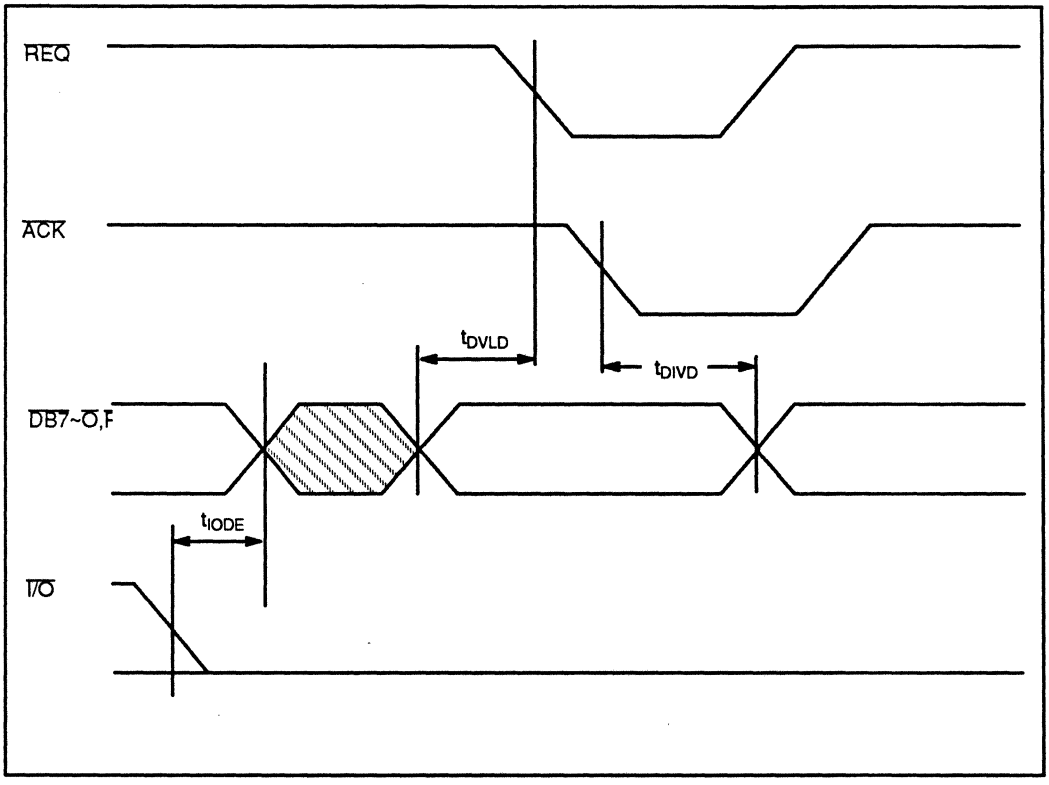


Item	Symbol	Min.	Typ.	Max.	Unit
$\overline{REQ}$ 'L' — $\overline{ACK}$ 'L'	t_{ROLA}	0			ns
$\overline{ACK}$ 'L' → $\overline{REQ}$ 'H'	t_{AROH}	10		90	ns
$\overline{REQ}$ 'H' → $\overline{ACK}$ 'H'	t_{ROHA}	0			ns
$\overline{ACK}$ 'H' → $\overline{REQ}$ 'L' *2	t_{AROL}	10		70	ns
$\overline{ACK}$ 'H' → $\overline{REQ}$ 'L' *1, *2	t_{RACY}	$2t_{CLF}$		$3t_{CLF}$ + 110	ns

Notes: *1: The time for $\overline{ACK}$'L' → REQ'L' is determined by the longer of (t_{AROH} + t_{ROHA} + t_{AROL}) and t_{RACY}.

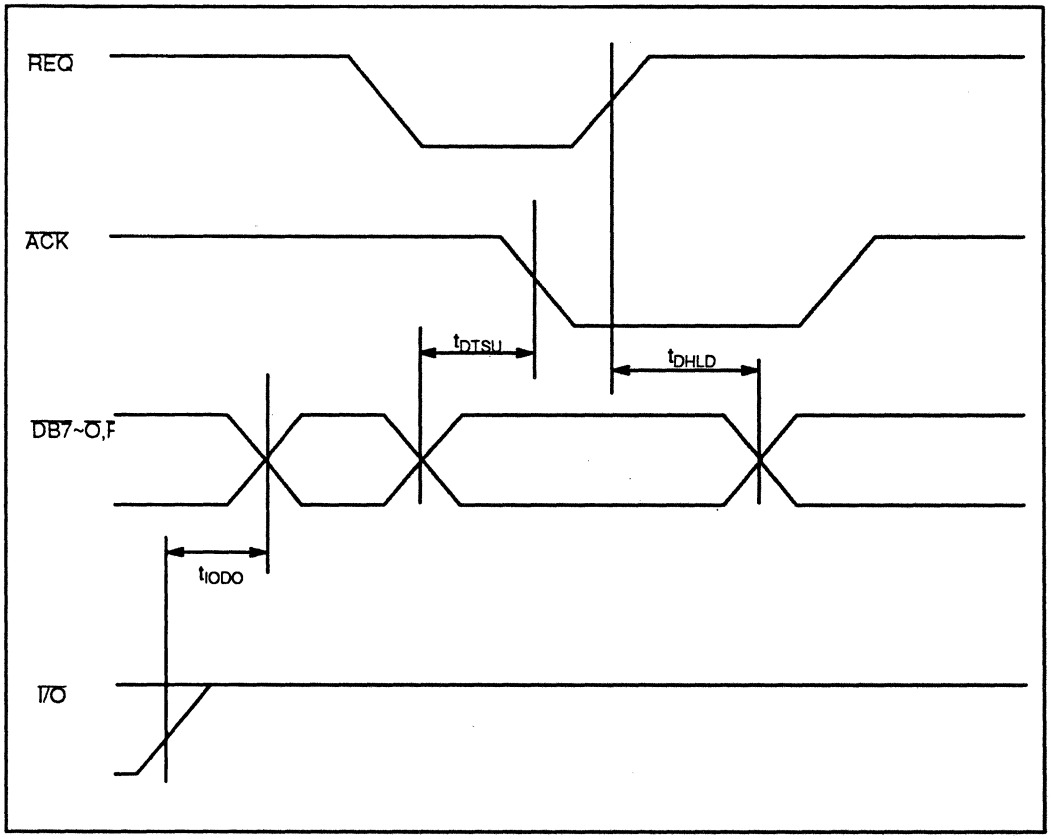
*2 The times assigned in this section do not apply in the following:
(1) For output operations, if the data buffer is empty.
(2) For input operations, if the data buffer is full.

6.5.24 Asynchronous Transfer (2) Output Operation



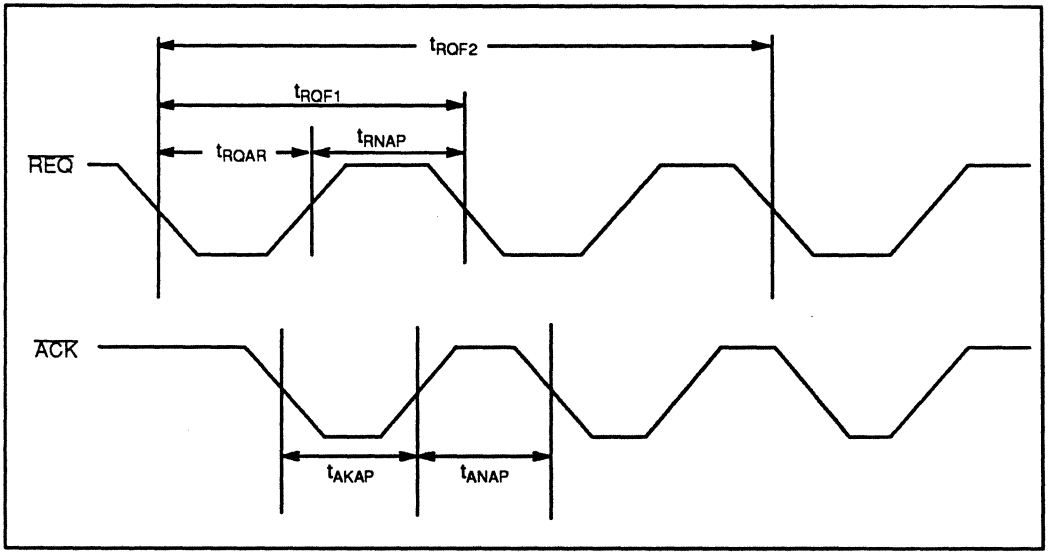
Item	Symbol	Min.	Typ.	Max.	Unit
I/O 'L' → data bus output	t_{IODE}	$7t_{CLF}$		$8t_{CLF}$ + 110	ns
Data bus output confirmation → REQ 'L'	t_{DVLD}	$2t_{CLF}$ - 80			ns
ACK 'L' → data bus hold	t_{DIVD}	15			ns

6.5.25 Asynchronous Transfer Target (3) Input Operation



Item	Symbol	Min.	Typ.	Max.	Unit
I/O 'H' → data bus output stop	t_{IODD}			60	ns
Data bus confirmation → ACK 'L'	t_{DTSU}	10			ns
REQ 'H' → data hold	t_{DHLD}	15			ns

6.5.26 Synchronous Transfer Initiator (1) REQ/ACK Cycle

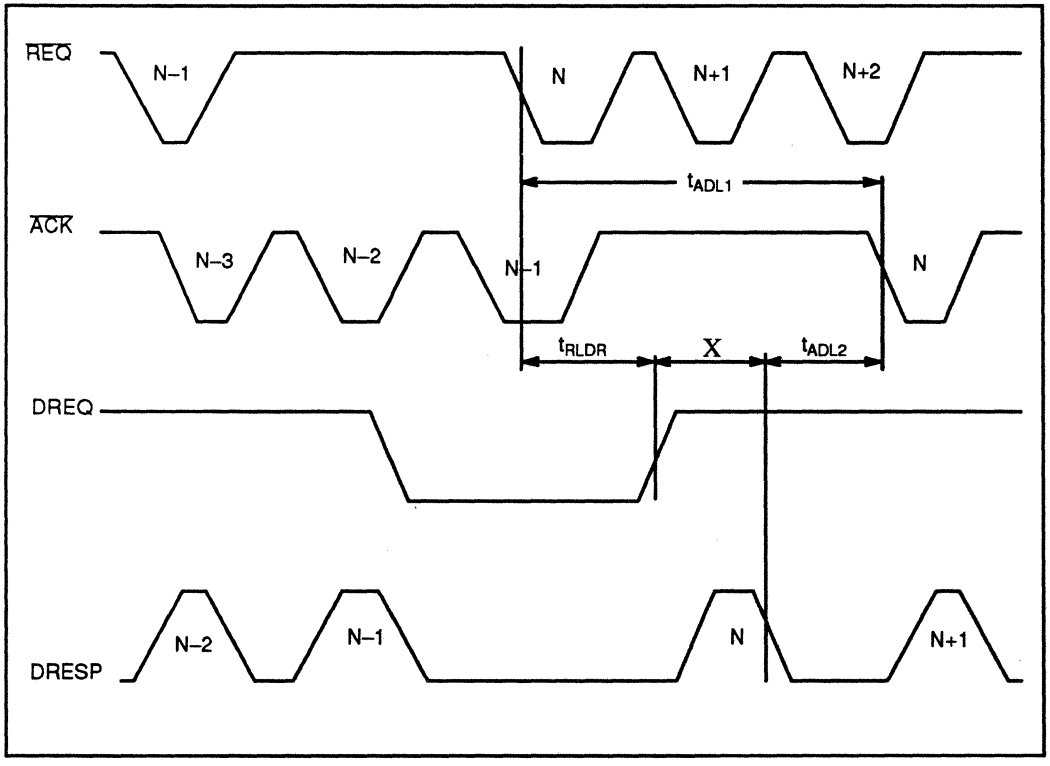


Item	Symbol	Min.	Typ.	Max.	Unit
$\overline{REQ}$ Assertion Period	t_{RQAP}	50			ns
$\overline{REQ}$ Nonassertion Period	t_{RNAP}	50			ns
$\overline{REQ}$ cycle time	t_{RQF1}	t_{CLF}			ns
$\overline{REQ}$ cycle time	t_{RQF2}	$3t_{CLF}$			ns
$\overline{ACK}$ Assertion Period	t_{AKAP}	t_{CLF} - 10			ns
*1 ACK Nonassertion Period	t_{ANAP}	n.t_{CLF} - 30			ns

Note: *1 n depends on the TMOD register setting.

TMOD Register		
Bit 3	Bit 2	n
0	0	1
0	1	2
1	0	3
1	1	4

6.5.27 Synchronous Transfer Initiator (2) REQ/ACK Timingœ



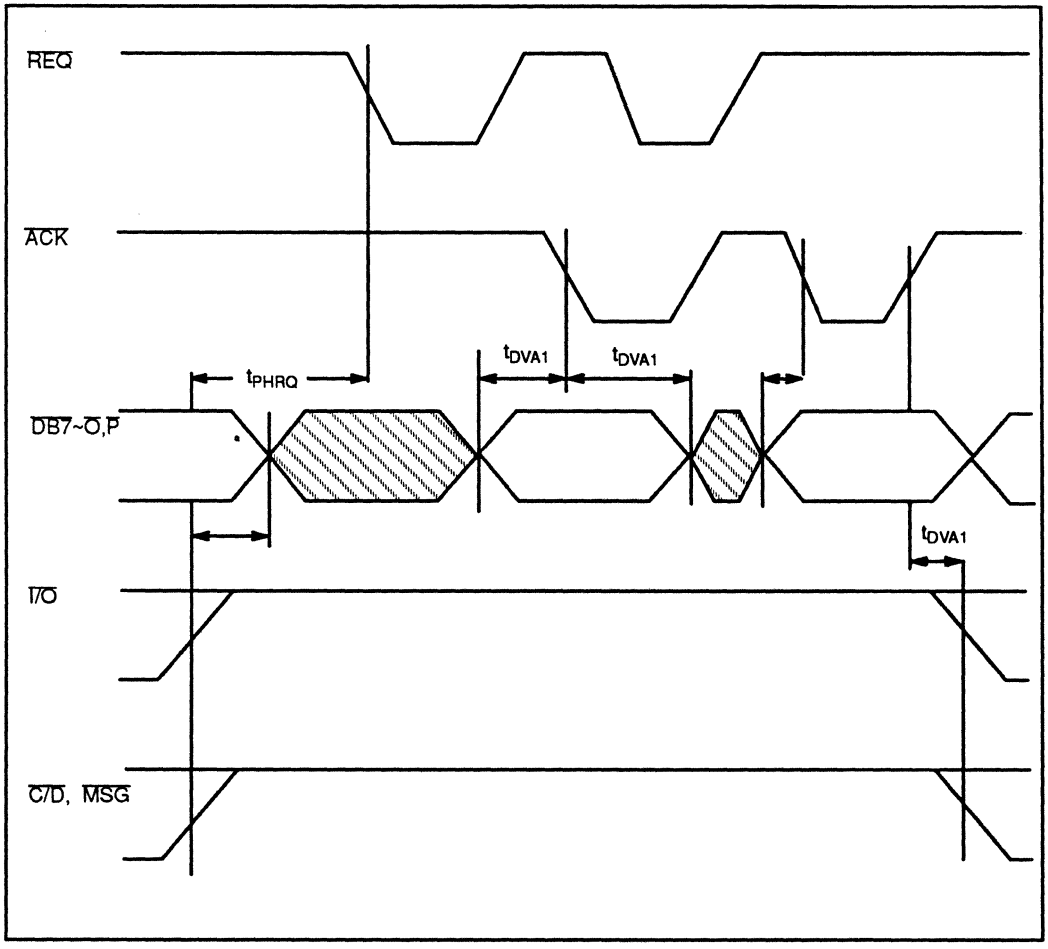
Item	Symbol	Min.	Typ.	Max.	Unit
$\overline{ACK}$ Assertion delay time (1) *1	t_{ADL1}	$3t_{CLF}$		$4t_{CLF}$ + 100	ns
$\overline{REQ}$ 'L' " DREQ 'H' *2	t_{RLDR}	$t_{CLF + 50}$		$3t_{CLF}$ + 60	ns
ACK Assertion delay time (2) *3	t_{ADL2}	$3t_{CLF}$		$4t_{CLF}$ + 120	ns

Notes :*1: Apply to output operations and to input operations when the maximum offset value is 4 or less.

*2: Apply to input operations.

*3 Apply to input operations when the maximum offset value is 5.8.
This is the minimum time after receiving the DRESP of byte N, until the ACK of byte N is transferred. In this case, the minimum time required from receiving the REQ of byte N until transferring the ACK of byte N is (t_{DRESP} + (DRESP assertion time x) + t_{ALD2}).

6.5.28 Synchronous Transfer Initiator (3) Output Operation



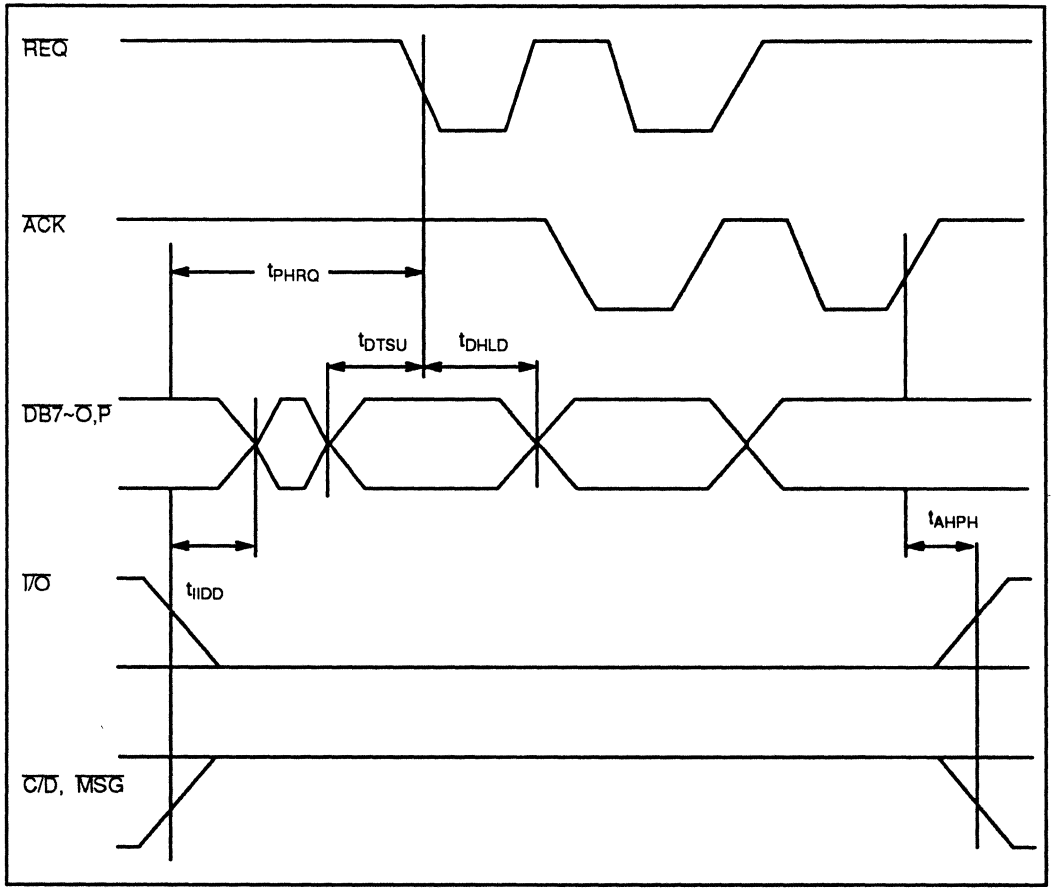
6.5.28 Synchronous Transfer Initiator (3) Output Operation (Continued)

Item	Symbol	Min.	Typ.	Max.	Unit
$\overline{I/O}$ 'H' → data bus output	t_{IIDE}	10		110	ns
Phase designation → $\overline{REQ}$ 'L'	t_{PHRQ}	100			ns
Data bus output *1	t_{DVA1}	$2t_{CLF}$ - 70			ns
confirmation → $\overline{ACK}$ 'L' *2	t_{DVA2}	$n.t_{CLF}$ - 60			
$\overline{ACK}$ 'L' → data bus hold	t_{AKDH}	t_{CLF} - 20			ns
$\overline{ACK}$ 'H' → phase change	t_{AHPH}	10			ns

Notes: *1 n depends on the TMOD register setting.
*2 The time from data bus output confirmation to $\overline{ACK}$ 'L' is set by the shorter of t_{DVA1} and t_{DVA2}.

TMOD Register		
Bit 3	Bit 2	n
0	0	1
0	1	2
1	0	3
1	1	4

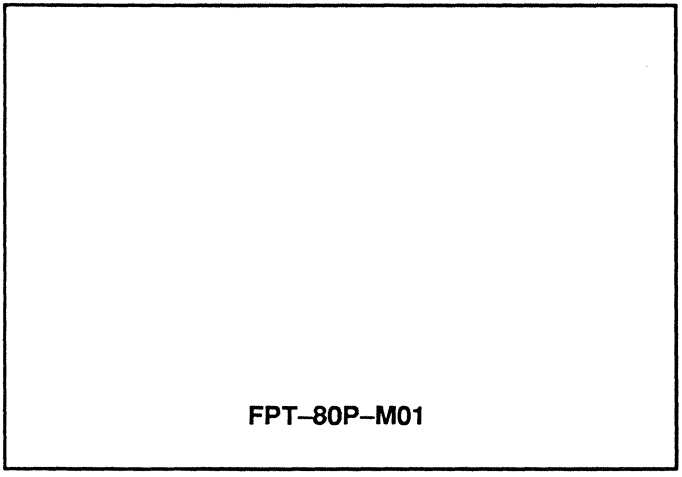
6.5.29 Synchronous Transfer Initiator (4) Input Operation

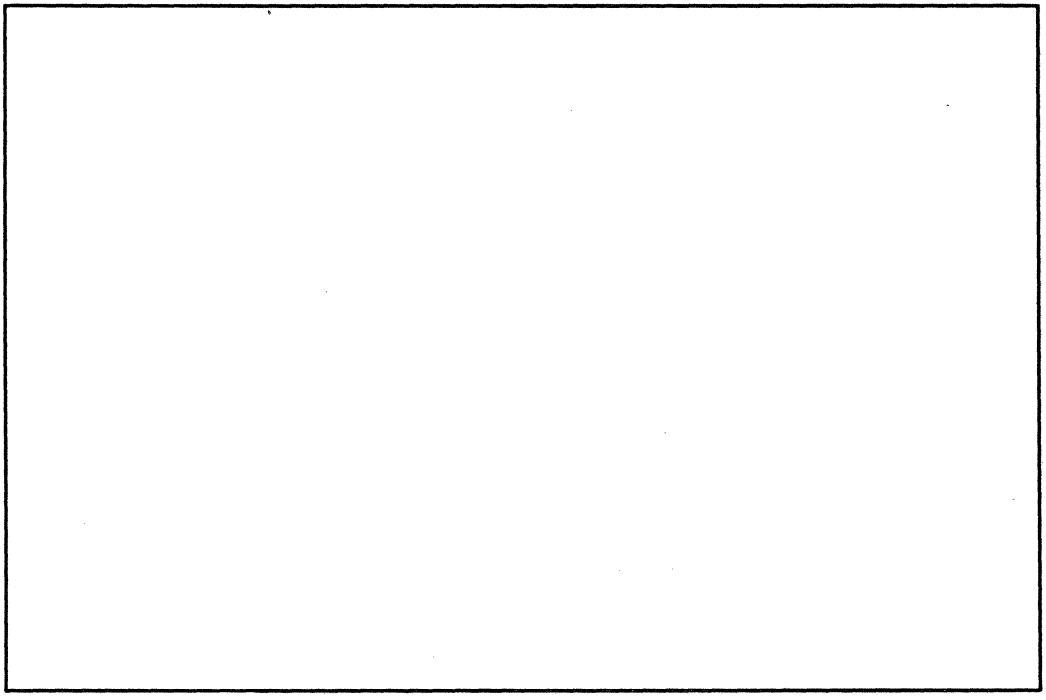


Item	Symbol	Min.	Typ.	Max.	Unit
I/O 'H' → data bus output stop	t_{IIDD}			110	ns
Phase designation → REQ 'L'	t_{PHRQ}	100			ns
Data bus confirmation → REQ 'L'	t_{DTSU}	10			ns
ACK 'L' → data bus hold	t_{DHLD}	40			ns
ACK 'H' → phase change	t_{AHPH}	10			ns

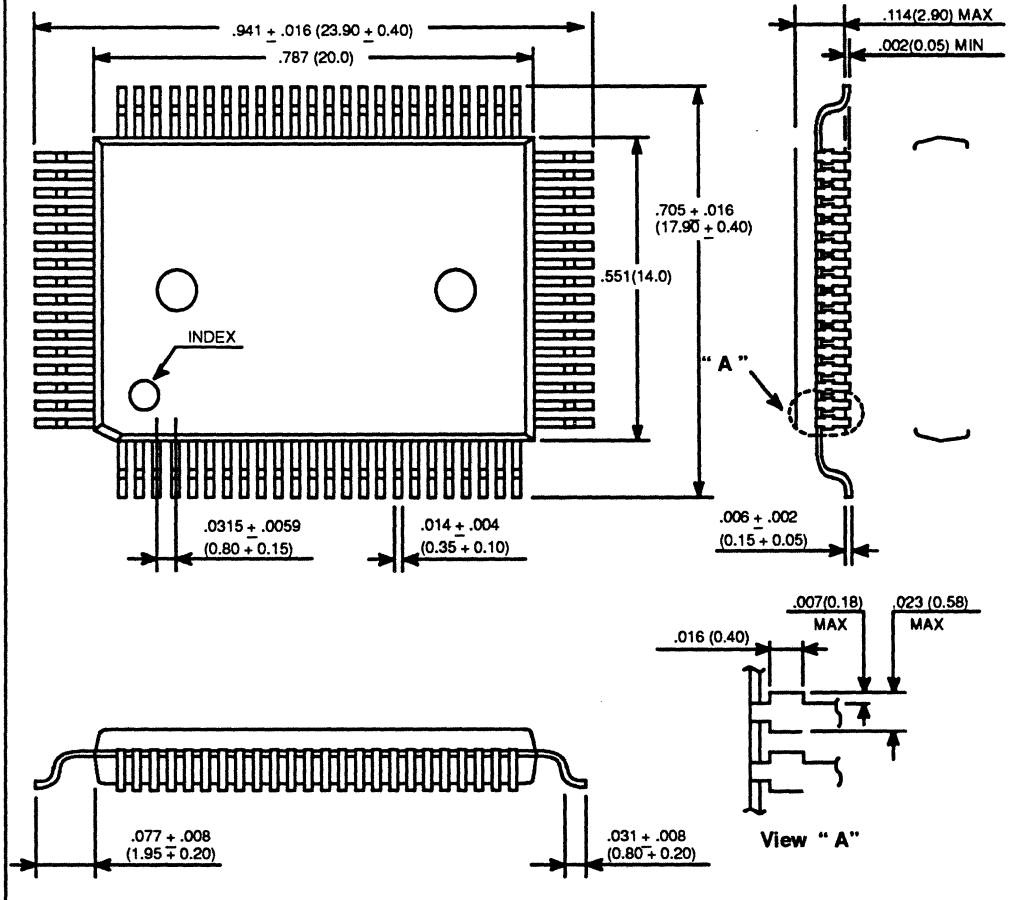
PACKAGE DIMENSIONS

80–Lead Plastic Flat Package





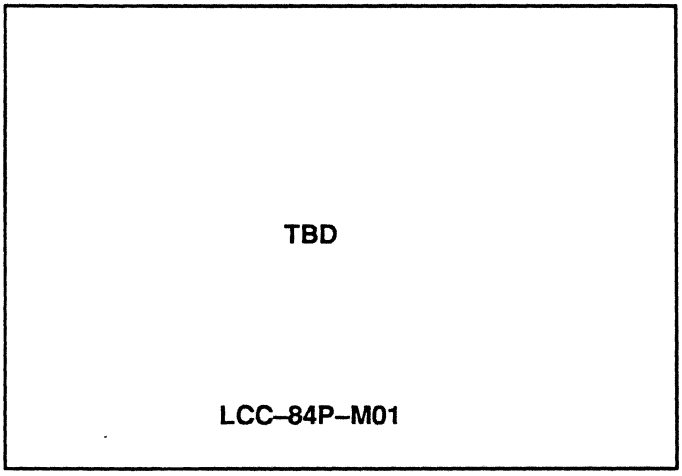
80–LEAD PLASTIC FLAT PACKAGE
(CASE No.: FPT–80P–M01)



Dimensions in
inches (millimeters)

PACKAGE DIMENSIONS (Continued)

84–Lead Plastic Leaded Chip Carrier



84–LEAD PLASTIC LEADED CHIP CARRIER
(CASE NO.: LCC–84P–M01)

TBD

MB89351
SCSI Protocol Controller

GENERAL DESCRIPTION

The Fujitsu MB89351 Small Computer System Interface (SCSI) is a System Protocol Controller (SPC) specifically designed to implement a SCSI-bus to CPU/DMAC interface. Except for SCSI synchronous mode transfers, the MB89351 can handle virtually all interface control procedures of the SCSI bus and the SPC is adaptable to either an 8-bit or a 16-bit CPU. To optimize efficiency and reduce CPU overhead, the MB89351 uses an 8-byte FIFO data buffer register and a 24-bit transfer byte counter. The SPC serves a wide range of applications acting as an INITIATOR or TARGET device for the SCSI. Thus, the device can be used as an I/O controller or as a host adapter.

The MB89351 SPC is fabricated in silicon-gate CMOS and housed in a 64-pin plastic shrink DIP or a 64-pin plastic flat package.

SCSI Compatibility

- Serves as either INITIATOR or TARGET
- DMA interface and parity check

Data Transfer Rate/Byte Counter

- Up to 2.5 Megabytes-per-second
- 8-byte FIFO Data Buffer
- 24-Bit transfer byte counter

Drive Options

- Single-ended
- Differential

Selectable Transfer Mode

- DMA interface
- Program transfer
- Manual transfer

Clock Requirements

- 8 MHz clock with 33%-to-66% duty cycle

Technology/Power Requirements

- Silicon-Gate CMOS
- Single +5V power supply

Available Packaging

- 64-pin plastic shrink DIP (suffix -PSH)
- 64-pin plastic flat package (suffix -PF)

ABSOLUTE MAXIMUM RATINGS[1]

Rating	Designator	Values		Unit
		Min.	Max.	
Supply Voltage	V_{CC}	V_{SS} – .05	V_{SS} + 7.0	V
Input Voltage[2]	V_{IN}	V_{SS} – .05	V_{SS} + 7.0	V
Output Voltage[2]	V_{OUT}	V_{SS} – .03	V_{SS} + 7.0	V
Operating Ambient Temperature	T_A	0	+70	°C
Storage Temperature	T_{STG}	–55	+150	°C

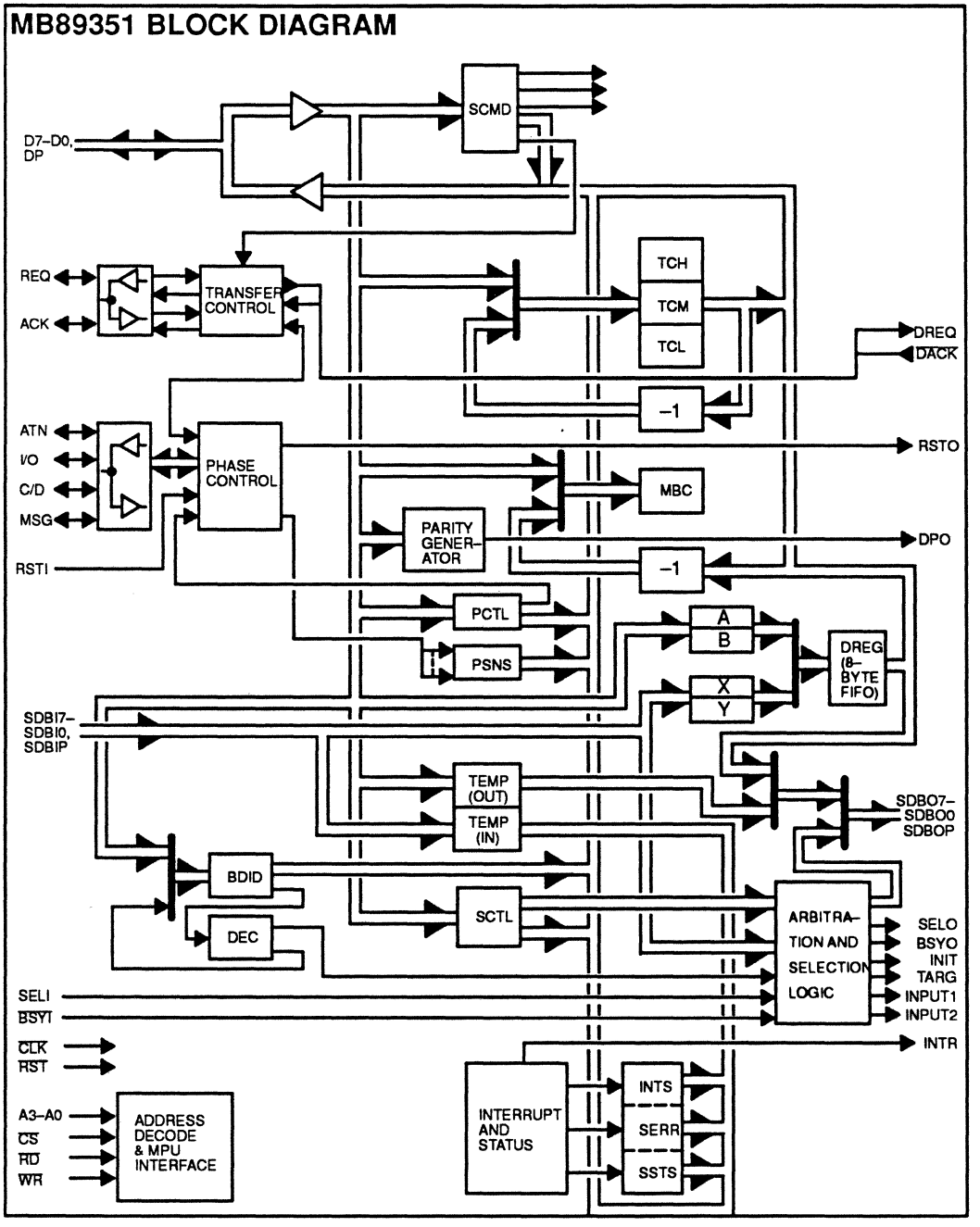
NOTES:

1. Permanent device damage may occur if the above absolute maximum ratings are exceeded. Functional operation should be restricted to the conditions as detailed in the operational sections of this data sheet. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.
2. Should not exceed V_{CC} + 0.5V.

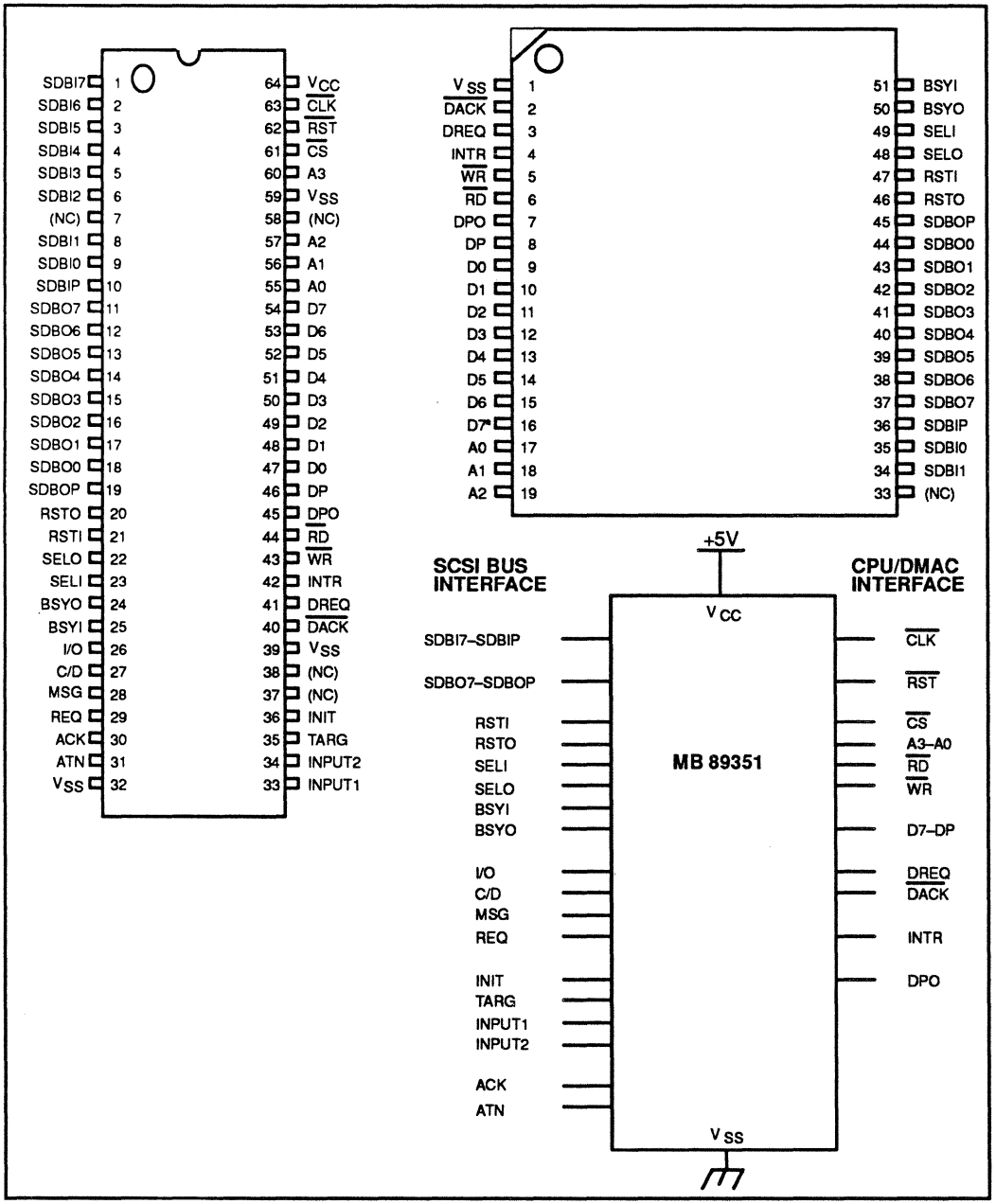
RECOMMENDED OPERATING CONDITIONS

Parameter	Designator	Values			Unit	Remarks
		Min.	Typ.	Max.		
Supply Voltage	V_{CC}	4.75	5.0	5.25	V	
	V_{SS}		0		V	
Operating Ambient Temperature	T_A	0		+70	°C	

MB89351 BLOCK DIAGRAM



PIN ASSIGNMENTS AND INTERFACE DIAGRAM



PIN DESCRIPTIONS

Designator	Pin No.		Function
	DIP	FPT	
V_{CC}	64	25	+5V power supply.
V_{SS}	32,39,59	1,20,59	Circuit ground.
$\overline{CLK}$	63	24	Clock input for controlling internal operation and data transfer speed of the SPC.
$\overline{RST}$	62	23	Asynchronous reset signal used to clear all internal circuits of the SPC.
$\overline{CS}$	61	22	Input selection enable signal for accessing an internal register. When active low, the following input/output signals are valid: RD, WR, A0–A3, DP0–DP7 and DP.
A3 A2 A1 A0	60 57 56 55	21 19 18 17	Address input signals for selecting an internal register in SPC. MSB is A3; LSB is A0. When $\overline{CS}$ is active low, read/ write is enabled for an internal register selected by these address inputs via data bus lines D0–D7 and DP.
$\overline{RD}$	44	6	The read strobe ($\overline{RD}$) input is used to readout the contents of an internal SPC register and is asserted only if $\overline{CS}$ is active low. The register to be read is specified by A0–A3; the input address data is input via D0–D7 and DP. In the program transfer mode, the falling edge of $\overline{RD}$ terminates the data read cycle.
$\overline{WR}$	43	5	The write strobe ($\overline{WR}$) input is used to write into an internal SPC register and is asserted only if $\overline{CS}$ is active low. On the falling edge of $\overline{WR}$, the data present on D0–D7 and DP is loaded into the internal register specified by A0–A3 (except when A0 = A1 = A2 = A3 = H). In the program transfer mode, the falling edge of $\overline{WR}$ indicates a data–ready condition to the MPU.

PIN DESCRIPTIONS (Continued)

Designator	Pin No.		Function
	DIP	FPT	
D7 – DP	54 – 45	16 – 8	Used to write/read data to/from an internal register in the SPC. The data bus is 3–state and bidirectional.The MSB is D7 and the LSB is D0; DP is an odd parity bit. When both $\overline{CS}$ and RDG inputs are active low (read operation), the contents of a selected internal register are output to the data bus. In operations other than read, the data bus is kept at a high–impedance level.
DPO	45	7	Outputs an odd parity for D0–D7. If parity bit is not generated for external memory, DPO can be used as an input parity bit for DP.
INTR	42	4	The INTR output signal is issued by the SPC and requests an interrupt to indicate completion of an internal operation or the occurrence of an error. Except for an interrupt caused by the RSTI input (reset condition in SCSI), interrupt masking is allowed. When an interrupt request is granted, the INTR signal remains active until the interrupt is cleared. In the program transfer mode, the INTR signal can be used as a data request signal instead of reading internal registers of the SPC; the data–request function is enabled by proper settings of the appropriate registers. The INTR signal is automatically disabled when interrupt conditions are not present.
DREQ	41	3	When the MB89351 is operating in the DMA mode and the data request (DREQ) output signal is active high, data is transferred between external memory and the SPC or vice–versa. During output operations, DREQ is active when the data buffers in the SPC are not full. During input operations, DREQ is active when valid data is present in the buffers. In either case, DMA transfers can occur and DREQ is asserted.
$\overline{DACK}$	40	2	An active low response signal to the DREQ which request data transfer between SPC and the external memory in the DMA mode. This signal, in DMA mode, functions similarly to the signal combination of $\overline{CS}$=low, A3=high, A2=low, A1=high, and A0=low (selection of DREG) in the program transfer mode. Since the DREG is selected by this $\overline{DACK}$ signal in the DMA mode instead of the address input from A3–A0, data transfer between DREG of SPC and external memory is possible.

PIN DESCRIPTIONS (Continued)

Designator	Pin No.		Function
	DIP	FPT	
SDBI7,SDBI6 SDBI5,SDBI4 SDBI3,SDBI2 SDBI1,SDBI0 SDBIP	1,2 3,4 5,6 8,9 10	27,28 29,30 31,32 34,35 36	Inputs from the SCSI data bus. The MSB is SDBI7; the LSB is SDBI0. SDBIP is an odd parity bit. Parity checking for the SCSI data bus is programmable.
SDBO7 SDBO6 SDBO5 SDBO4 SDBO3 SDBO2 SDBO1 SDBO0 SDBOP	11 12 13 14 15 16 17 18 19	37 38 39 40 41 42 43 44 45	Outputs to the SCSI data bus. The MSB is SDBO7; the LSB is SDBO0. SDBOP is an odd parity bit. An open-collector bus driver is used to connect the SCSI bus. **(Note: For the SDBI and SDBO pin groupings, only one pin is used for output and the other pins are used for input during arbitration. During selection, reselection, and normal data transfers, all pins are used for outputs. Typical system connections are shown in the Pin Assignments and Interface Diagram).**
RSTO,RSTI	20,21	46,47	RSTI is a reset input from other SCSI devices; RSTO is a reset output to other SCSI devices. Both signals are active high and can be masked.
SELO,SELI	22,23	48,49	SELO is an output that corresponds to the INITIATOR or TARGET device; SELI inputs the response to the SPC during the selection or reselection phase. Both signals are active high.
BSYO,BSYI	24,25	50,51	BSYO indicates the SCSI bus is in the output mode of operation, whereas, BSYI indicates the bus is operating in the input mode. One or the other of these signals is active high during arbitration and in the "connected status" mode of operation.
I/O	26	52	During the data transfer phase, the I/O signal indicates the transfer direction. When I/O is high, data is transferred from the TARGET to the INITIATOR. When I/O is low, data is transferred from the INITIATOR to the TARGET.
C/D	27	53	During the data transfer phase, the C/D signal is set high during the command–, status–, and message–phases of operation.
MSG	28	54	In the data transfer phase, the MSG signal is set high only during the message phase.

PIN DESCRIPTIONS (Continued)

<table>
<tr><th rowspan="2">Designator</th><th colspan="2">Pin No.</th><th rowspan="2">Function</th></tr>
<tr><th>DIP</th><th>FPT</th></tr>
<tr><td>REQ</td><td>29</td><td>55</td><td>In the data transfer phase, the REQ signal is used to notify the INITIATOR that the TARGET is ready to receive or send data. The REQ input is used as a timing control signal in the data transfer sequence.</td></tr>
<tr><td>INiT,TARG</td><td>36,35</td><td>63,62</td><td>These two output signals indicate the operational status of the SPC. The INIT and TARG outputs can also be used as SLCSI driver circuit control signals.
<table><tr><th>INIT</th><th>TARG</th><th>SPC Status</th></tr><tr><td>L</td><td>L</td><td>Logically disconnected from the SCSI.</td></tr><tr><td>L</td><td>H</td><td>Executing the reselection phase or operating as a TARGET.</td></tr><tr><td>H</td><td>L</td><td>Executing the selection phase or operating as an INITIATOR.</td></tr></table></td></tr>
<tr><td>INPUT1,INPUT2</td><td>33,34</td><td>60,61</td><td>These two outputs are directly related to the preceding INIT and TARG signals. INPUT1 corresponds to INIT and INPUT2 corresponds to TARG. However, unlike INIT and TARG, INPUT1 and INPUT2 are set High when the SPC is not logically connected to the SCSI.</td></tr>
<tr><td>ACK</td><td>30</td><td>56</td><td>In the data transfer phase, the acknowledge signal is in response to a transfer request (REQ) signal from the TARGET. In the same way as REQ, an ACK input is used as a timing signal in the data transfer sequence.</td></tr>
<tr><td>ATN</td><td>31</td><td>57</td><td>Except during arbitration and during the bus free phase, the ATN signal is used to notify the TARGET that the INITIATOR has a prepared message.</td></tr>
<tr><td>NC</td><td>7,37,
38,58</td><td>33,64</td><td>No connection.</td></tr>
<tr><td>Open</td><td>—</td><td>26,58</td><td>—</td></tr>
</table>

ADDRESSING OF INTERNAL REGISTERS

SPC has internal registers consisting of 17 bytes that are accessible from an external circuit. These internal registers are used for controlling SPC internal operation and indicating SPC processing status/result status. A unique address is assigned to each internal register, and a particular register is identified by address bits A3 to A0. The following table shows internal register addressing:

Internal Register Addressing

$\overline{CS}$	A3	A2	A1	A0	Operation	Register Name	Abbr.
0	0	0	0	0	Read/Write	Bus Device ID	BDID
0	0	0	0	1	Read/Write	SPC Control	SCTL
0	0	0	1	0	Read/Write	Command	SCMD
0	0	0	1	1	—	Open	—
0	0	1	0	0	Read	Interrupt Sense	INTS
					Write	Reset Interrupt	
0	0	1	0	1	Read	Phase Sense	PSNS
					Write	SPC Diagnostic Control	SDGC
0	0	1	1	0	Read	SPC Status	SSTS
					Write	—	
0	0	1	1	1	Read	SPC Error Status	SERR
					Write	—	
0	1	0	0	0	Read/Write	Phase Control	PCTL
0	1	0	0	1	Read	Modified Byte Counter	MBC
					Write	—	
0	1	0	1	0	Read/Write	Data Register	DREG
0	1	0	1	1	Read/Write	Temporary Register	TEMP
0	1	1	0	0	Read/Write	Transfer Counter (High)	TCH
0	1	1	0	1	Read/Write	Transfer Counter (Middle)	TCM
0	1	1	1	0	Read/Write	Transfer Counter (Low)	TCL

BIT ASSIGNMENTS

The following table shows the bit assignments to each internal register. When accessing an internal register (in read/write), remember the following:

1. The internal register block includes the read–only/write–only register and those having different meanings in read and write operations.
2. A write command to a read–only register is ignored.
3. If the write–only register is read out, the data and parity bit are undefined.
4. At bit positions indicating "_" for a write in Table 3.2.2, either 1 of 0, or may be written.

Bit Assignments for Internal Registers

HEX Address	Register and Mnemonic	R/W Operation	7 (MSB)	6	5	4	3	2	1	0 (LSB)	Parity
0	Bus Device ID (BDID)	R	#7	#6	#5	#4	#3	#2	#1	#0	0
		W	SCSI Bus Device ID					ID4	ID2	ID1	—
1	SPC Control (SCTL)	R/W	Reset & Disable	Control Reset	Diag. Mode	ARBIT Enable	Parity Enable	Select Enable	Reselect Enable	INT Enable	P
2	Command (SCMD)	R/W	Command Code			RST Out	Intercept Xfer	Transfer PRG Xfer	Modifier 0	Term Mode	P
3		R									
		W									
4	Interrupt Sense (INTS)	R	Selected	Reselected	Disconnect	Command Complete	Sevice Required	Time Out	SPC Hard Error	Reset Condition	P
		W	Reset Interrupt								—
5	Phase Sense (PSNS)	R	REQ	ACK	ATN	SEL	BSY	MSG	C/D	I/O	P
	SPC Diag Control (SDGC)	W	Diag. REQ	Diag. ACK	Xfer Enable	—	Diag. BSY	Diag. MSG	Diag. C/D	Diag. I/O	—
6	SPC Status (SSTS)	R	Connected INIT	TARG	SPC BSY	XFER In Progress	SCSI RST	TC=0	DREG Full	Status Empty	P

Bit Assignments For Internal Registers (Continued)

HEX Address	Register and Mnemonic	R/W Operation	7 (MSB)	6	5	4	3	2	1	0 (LSB)	Parity
7	SPC Error Status (SERR)	R	Data Error SCSI	Data Error SPC	Xfer Out	0	TC Parity Error	0	Short Period	0	P
8	Phase Control (PCTL)	R/W	Bus Free Interrupt Enable	0				Transfer Phase MSG Out	C/D Out	I/O Out	P
9	Modified Byte Counter (MBC)	R	0				MBC Bit3	Bit2	Bit1	Bit0	P
A	Data Register (DREG)	R/W	Internal Data Register (8 Byte FIFO) Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	P
B	Temporary Register (TEMP)	R	Temporary Data (Input: From SCSI) Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	P
		W	Temporary Data (Output: From SCSI) Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	P
C	Transfer Counter High (TCH)	R/W	Transfer Counter High (MSB) Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16	P
D	Transfer Counter Mid. (TCM)	R/W	Transfer Counter High (2nd Byte) Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	P
E	Transfer Counter Low (TCL)	R/W	Transfer Counter High (LSB) Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	1Bit	Bit0	P
F	External Buffer (EXBF)	R									
		W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	P

DC CHARACTERISTICS

(Recommended operating conditions unless otherwise specified)

Parameter	Designator	Conditions	Values			Unit
			Min.	Typ.	Max.	
Input High Voltage	V_{IH}		2.2		V_{CC} + 0.5	V
Input Low Voltage	V_{IL}		V_{SS} – 0.5		0.8	V
Output High Voltage	V_{OH}	I_{OH} = –0.4mA	4.0		V_{CC}	V
Output Low Voltage	V_{OL}	I_{OL} = +3.2mA	V_{SS}		0.4	V
Input Leakage Current	I_{IL}	V_{IN} = 0V to 5.25V	–10		20	mA
Output Leakage Current	I_{IZ}	V_{IN} = 0V to 5.25V	–40		40	mA
Active Supply Current	I_{CC}	fc = 8MHz, All outputs open			10	mA
Standby Supply Current	I_{CS}	fc = 8 MHz, All outputs open, inputs fixed, $\overline{RST}$ active			40	mA

AC CHARACTERISTICS

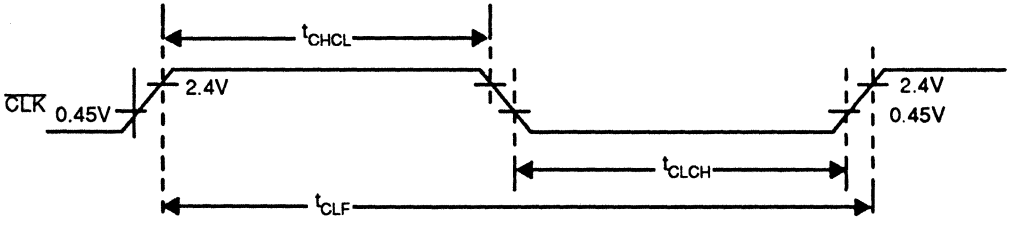
(Recommended operating conditions unless otherwise noted)

CPU/DMAC Interface

$\overline{CLK}$ Input

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
$\overline{CLK}$ Cycle Time	t_{CLF}	125		200	ns
$\overline{CLK}$ High Time	t_{CHCL}	44			ns
$\overline{CLK}$ Low Time	t_{CLCH}	44			ns
$\overline{CLK}$ Rise Time	t_r			10	ns
$\overline{CLK}$ Fall Time	t_f			10	ns

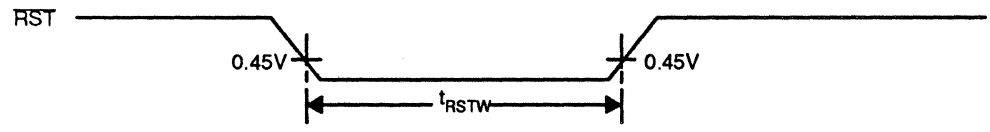
$\overline{CLK}$ Input Timing



AC CHARACTERISTICS (Continued)

RST Input					
Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
RST Pulse Width	t_{RSTW}	100			ns

RST Input Timing

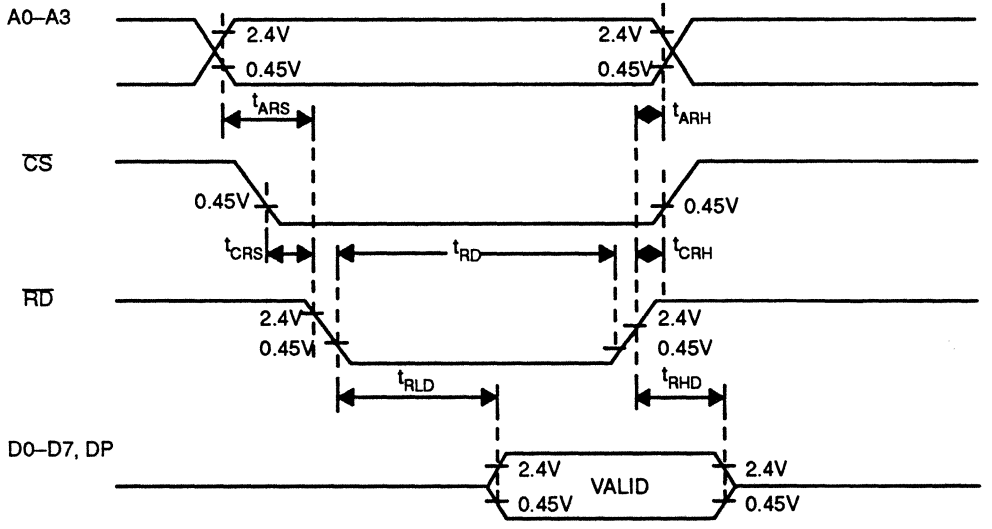


AC CHARACTERISTICS (Continued)

Register Read

Parameter	Designator	Test Conditions	Values			Unit
			Min.	Typ.	Max.	
Address Setup Time	t_{ARS}		40			ns
Address Hold Time	t_{ARH}		10			ns
$\overline{CS}$ Setup Time	t_{CRS}		25			ns
$\overline{CS}$ Hold Time	t_{CRH}		10			ns
Data Valid Time (from $\overline{RD}$ Low)	t_{RLD}	CL = 80 pF			90	ns
Data Hold Time (from $\overline{RD}$ High)	t_{RHD}	CL = 20 pF	10		60	ns
$\overline{RD}$ Pulse Width	t_{RD}		120			ns

Register Read Timing

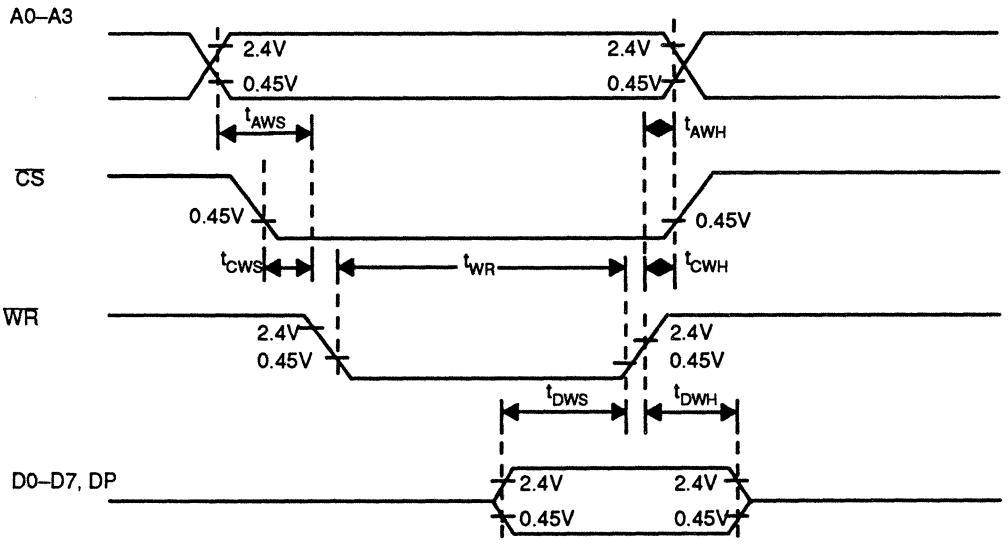


AC CHARACTERISTICS (Continued)

Register Write

Parameter	Designator	Test Conditions	Values			Unit
			Min.	Typ.	Max.	
Address Setup Time	t_{AWS}		40			ns
Address Hold Time	t_{AWH}		10			ns
$\overline{CS}$ Setup Time	t_{CWS}		25			ns
$\overline{CS}$ Hold Time	t_{CWH}		10			ns
Data Valid Time (from $\overline{WR}$ Low)	t_{DWS}		30			ns
Data Hold Time (from $\overline{WR}$ High)	t_{DWH}		20			ns
$\overline{WR}$ Pulse Width	t_{WR}		100			ns

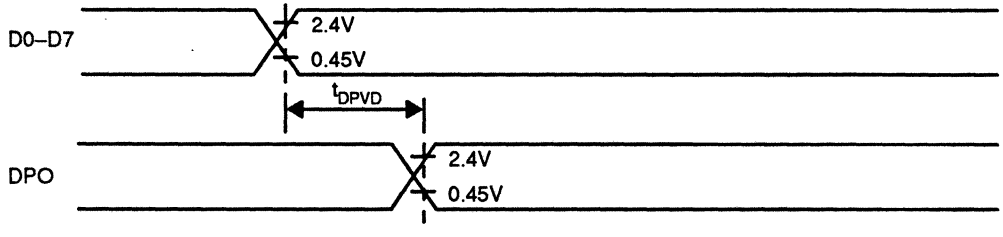
Register Write Timing



AC CHARACTERISTICS (Continued)

DPO (Data Parity Output)						
Parameter	**Designator**	**Test Conditions**	**Values**			**Unit**
			Min.	**Typ.**	**Max.**	
DPO Valid Time (from D7–D0 to DPO Valid)	t_{DPVD}	CL = 30 pF			60	ns

DPO Output Timing

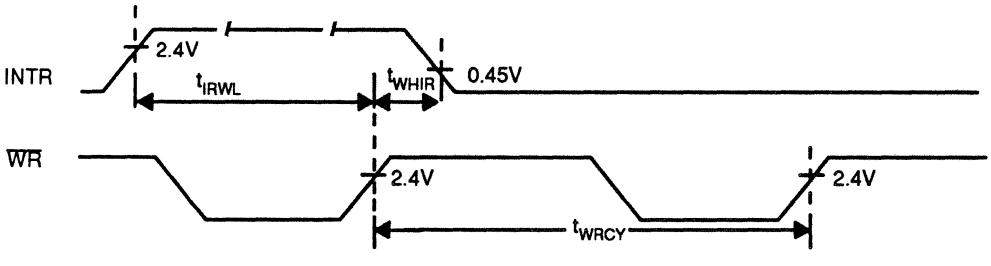


AC CHARACTERISTICS (Continued)

INTR (Interrupt Request) Output						
Parameter	Designator	Test Conditions	Values			Unit
			Min.	Typ.	Max.	
$\overline{WR}$ Service Time (from INTR High to $\overline{WR}$ High)	t_{IRWL}		0			ns
INTR Release Time (from $\overline{WR}$ High to INTER Low)	t_{WHIR}	CL = 10pf	t_{CLF}		$2t_{CLF}$+ 100	ns
INTR Reset Cycle Time	t_{WRCY}		$4t_{CLF}$			ns

NOTE: Applicable only when interrupt reset is executed.

INTR Output Timing



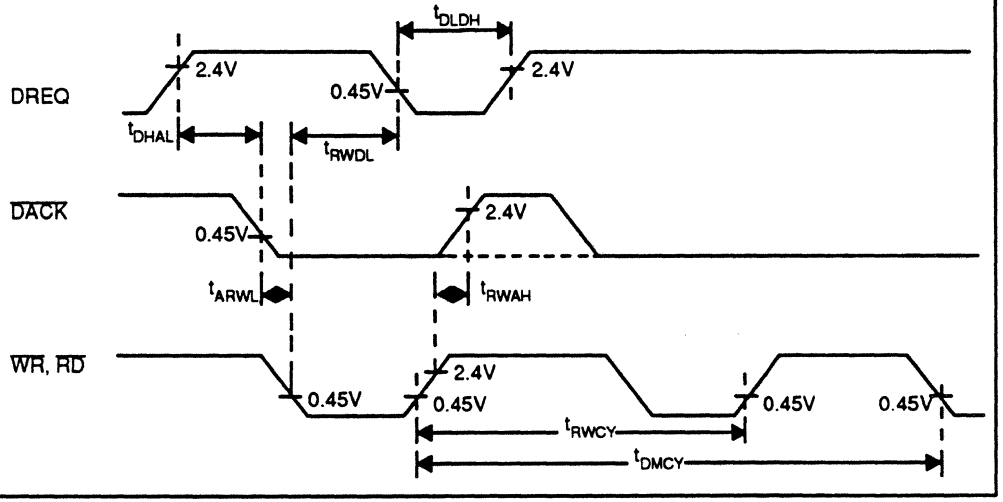
AC CHARACTERISTICS (Continued)

DMA Access

Parameter	Designator	Test Conditions	Values			Unit
			Min.	Typ.	Max.	
DACK Service Time (from DREQ High to $\overline{DACK}$ Low)	t_{DHAL}		0			ns
$\overline{WR}$ and $\overline{RD}$ Service Time (from $\overline{DACK}$ Low to $\overline{WR}$ or $\overline{RD}$ Low)	t_{ARWL}		40			ns
DREQ Release Time (from $\overline{WR}$ or $\overline{RD}$ Low to DREQ Low)	t_{RWDL}	CL = 30 pF	35		150	ns
$\overline{DACK}$ Hold Time (from $\overline{WR}$ or $\overline{RD}$ High to $\overline{DACK}$ High)	t_{RWAH}		10			ns
DREQ Interval (from DREQ Low to DREQ High)	t_{DLDH}		0			ns
DREG Access Cycle Time (1)	t_{RWCY}		$2t_{CLF}$			ns
DREG Access Cycle Time (2)	t_{DMCY}		$3t_{CLF}$			ns

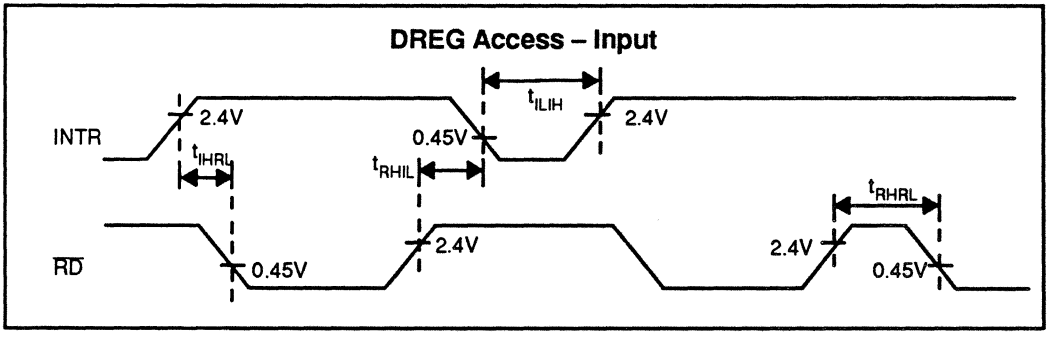
NOTE: The $\overline{WR}$ parameter is applicable when data buffer register will be full; the $\overline{RD}$ parameter is applicable when the data buffer register will be empty.

DMA Access Timing



AC CHARACTERISTICS (Continued)

DREG Access – Program Transfer with INTR (Input Operation)						
Parameter	Designator	Test Conditions	Values			Unit
			Min.	Typ.	Max.	
$\overline{RD}$ Service Time (from INTR High to $\overline{RD}$ Low)	t_{IHRL}		0			ns
INTR Release Time (from $\overline{RD}$ High to INTR Low)	t_{RHIL}	CL = 20pF	35		150	ns
INTR Recovery Time (from INTR Low to INTR High)	t_{ILIH}		0			ns
$\overline{RD}$ Recovery Time (from $\overline{RD}$ High to $\overline{RD}$ Low)	t_{RHRL}		50			ns

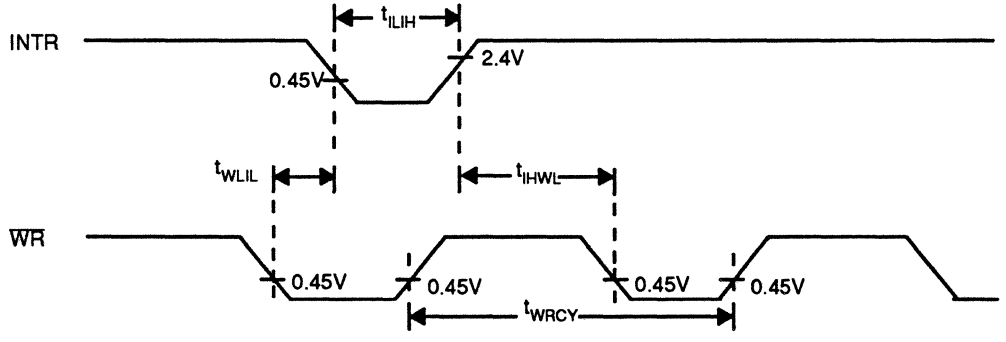


AC CHARACTERISTICS (Continued)

DREG Access – Program Transfer with INTR (Output Operation)

Parameter	Designator	Test Conditions	Values			Unit
			Min.	Typ.	Max.	
$\overline{WR}$ Service Time (from INTR High to $\overline{WR}$ Low)	t_{IHWL}		0			ns
INTR Release Time (from $\overline{WR}$ Low to INTR Low)	t_{WLIL}	CL = 20pF	35		150	ns
INTR Recovery Time (from INTR Low to INTR High)	t_{ILIH}		0			ns
$\overline{WR}$ Cycle Time	t_{WRCY}		$2t_{CLF}$			ns

DREG Access – Output



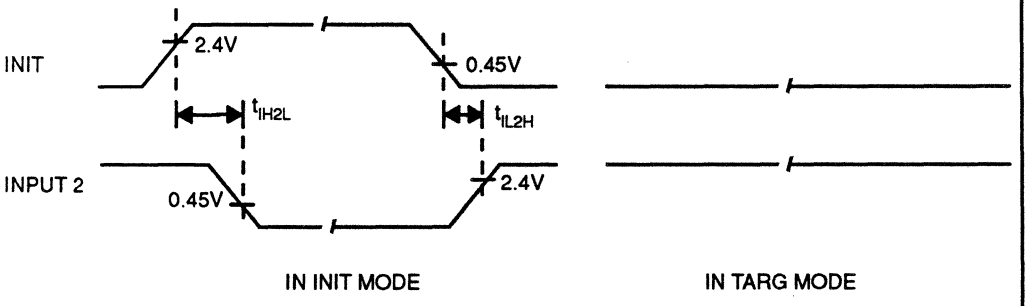
AC CHARACTERISTICS (Continued)

INIT/TARG and INPUT1/INPUT2 Output

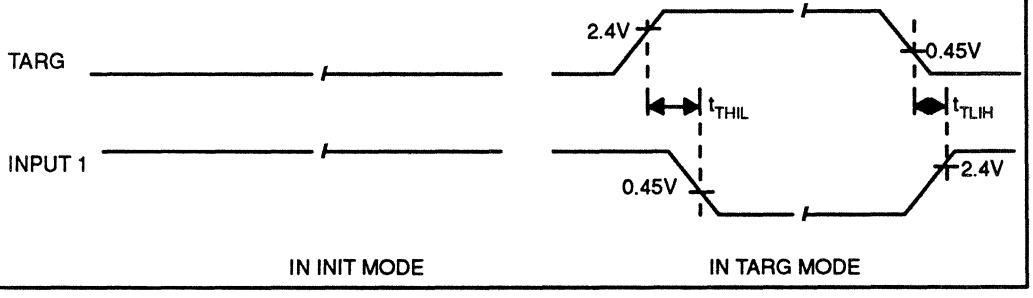
Parameter	Designator	Test Conditions	Values			Unit
			Min.	Typ.	Max.	
INPUT2 Valid Time (from INIT High to INPUT2 Low)	t_{IH2L}	CL = 30pF	0		20	ns
INPUT2 Invalid Time (from INIT Low to INPUT2 High)	t_{IL2H}	CL = 30pF	–20		20	ns
INPUT1 Valid Time (from TARG High to INPUT1 Low)	t_{TH1L}	CL = 40pF	0		20	ns
INPUT1 Invalid Time (from TARG Low to INPUT1 High)	t_{TL1H}	CL = 40pF	–20		20	ns

INIT/TARG and INPUT1/INPUT2 Output Timing

INIT and INPUT2 Output Timing:



TARG and INPUT1 Output Timing:



AC CHARACTERISTICS (Continued)

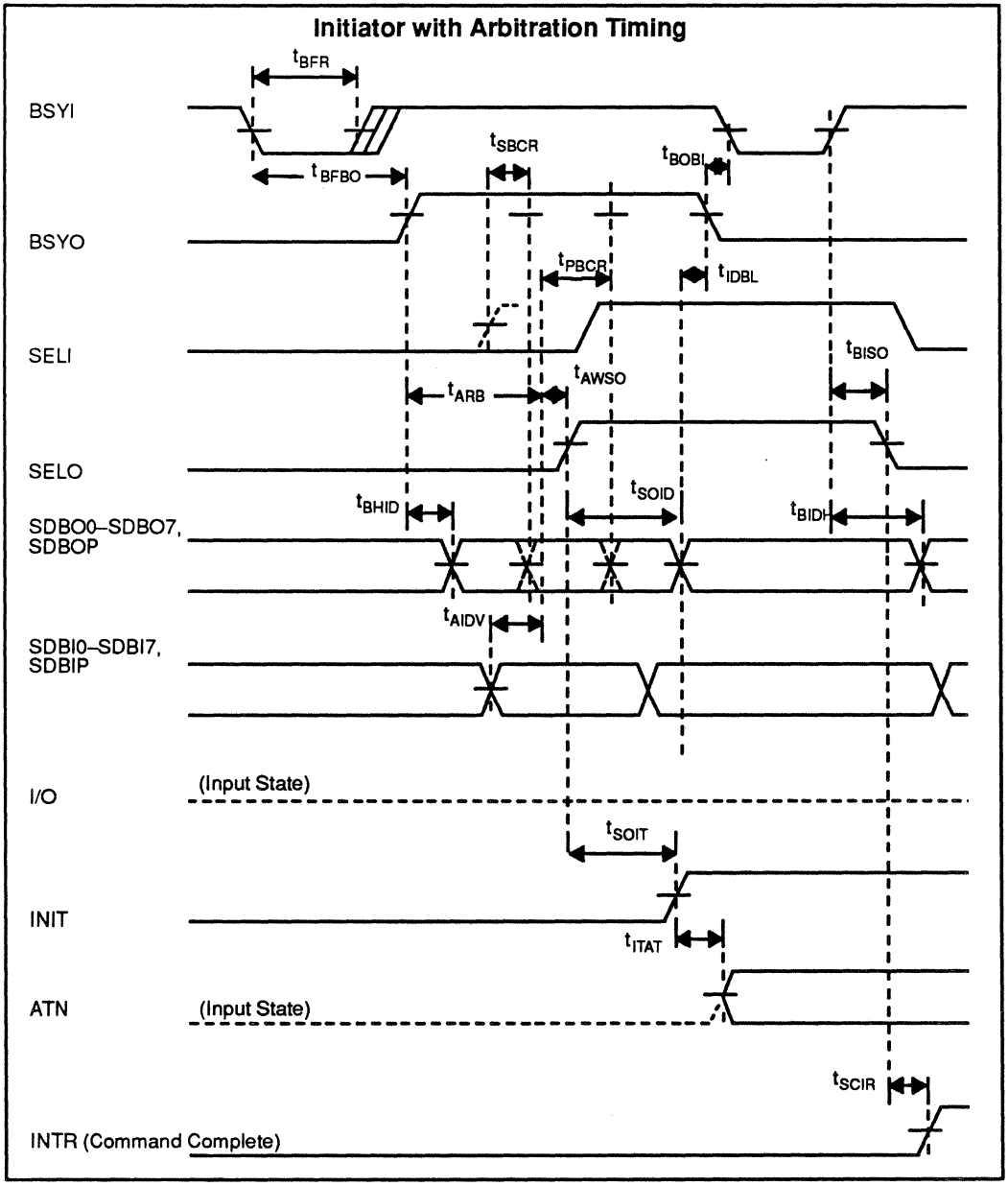
SCSI Bus Interface – Selection Phase Timing

Initiator with Arbitration

Parameter	Designator	Test Conditions	Values [1]			Unit
			Min.	Typ.	Max.	
Bus Free Time [2]	t_{BFR}		$4t_{CLF} + 50$			ns
From BSYI Low to BSYO High	t_{BFBO}	CL = 10pF	$(6 + n) \times t_{CLF}$		$(7 + n) \times t_{CLF} + 60$	ns
From BSYO High to Device ID Out	t_{BHID}	CL = 30pF	0		60	ns
From BSYO High to Prioritize	t_{ARB}		$32t_{CLF} - 60$			ns
From Data Bus Valid to Prioritize	t_{AIDV}		100			ns
From Bus Usage Permission Granted to SELO High	t_{AWSO}	CL = 10pF	0		50	ns
From SELO High to SELECT ID Output	t_{SOID}	CL = 30pF	$11t_{CLF} - 30$			ns
From SELO High to INIT High	t_{SOIT}	CL = 10pF	$11t_{CLF}$			ns
From INIT High to ANT High	t_{ITAT}	CL = 10pF	0		60	ns
From SELECT ID Output to BSYO Low	t_{IDBL}	CL = 10pF	$2t_{CLF} - 80$			ns
From BSYO Low to BSYI Low	t_{BOBI}	CL = 10pF	0		t_{CLF}	ns
From BSYI High to BSYO Low	t_{BISO}	CL = 10pF	$2t_{CLF}$		$3t_{CLF} + 60$	ns
From BSYI High to $\overline{SELO}$ Low	t_{BIDH}	CL = 10pF	$2t_{CLF}$			ns
From SELO Low to INTR High	t_{SCIR}	CL = 30pF			60	ns
From SELI High to BSYO Low, ID Bit Low	t_{SBCR}	CL = 30pF (BSYO) CL = 30pF (SDBO0–SDBO7, SDBOP)			$3t_{CLF} + 100$	ns
From Priority Judge to BSYO and ID Bit Low	t_{PBCR}	CL = 30pF (BSYO) CL = 30pF (SDBO0–SDBO7, SDBOP)			80	ns

NOTES: 1. n = value of TCL register.
2. The bus free time is the minimum time interval until the booked select command is executed.

AC CHARACTERISTICS (Continued)

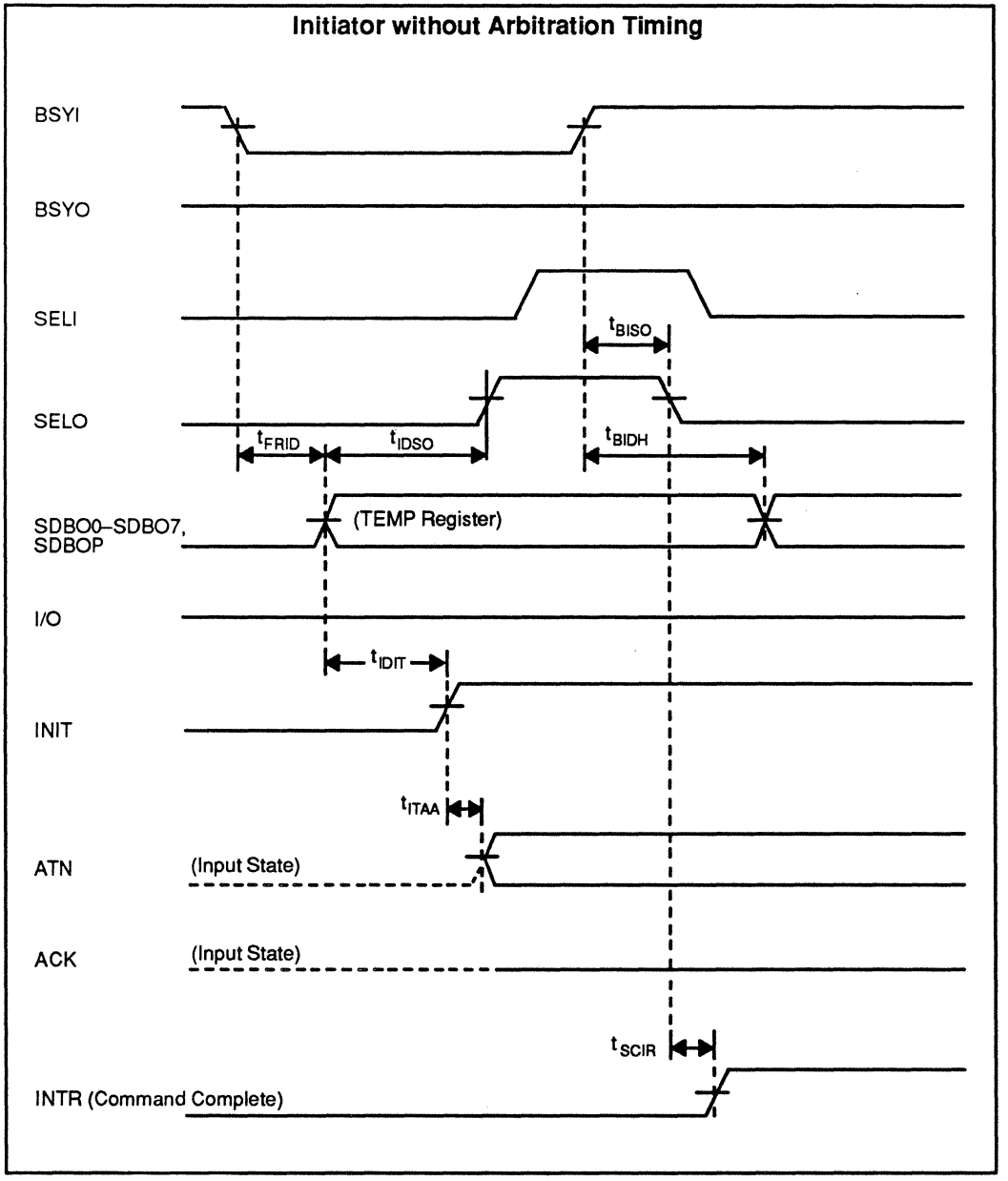


AC CHARACTERISTICS (Continued)

Initiator without Arbitration						
Parameter	Designator	Test Conditions	Values (Note)			Unit
			Min.	Typ.	Max.	
From BSYI Low to SELECT ID Output	t_{FRID}	CL = 30pF	(6 + n) x t_{CLF}		(7 + n) x t_{CLF}+60	ns
From ID Output to SELO High	t_{IDSO}	CL = 10pF	11t_{CLF} – 80			ns
From ID Output to INIT High	t_{IDIT}	CL = 10pF	11t_{CLF} – 80			ns
From INIT High to ATN High	t_{ITAA}	CL = 10pF	0		60	ns
From BSYI High to SELO Low	t_{BISO}	CL = 10pF	2t_{CLF}			ns
From BSYI High to SELECT ID Hold	t_{BIDH}	CL = 30pF	2t_{CLF}			ns
From SELO Low to INTR High	t_{SCIR}				60	ns

NOTE: n = value of TCL register.

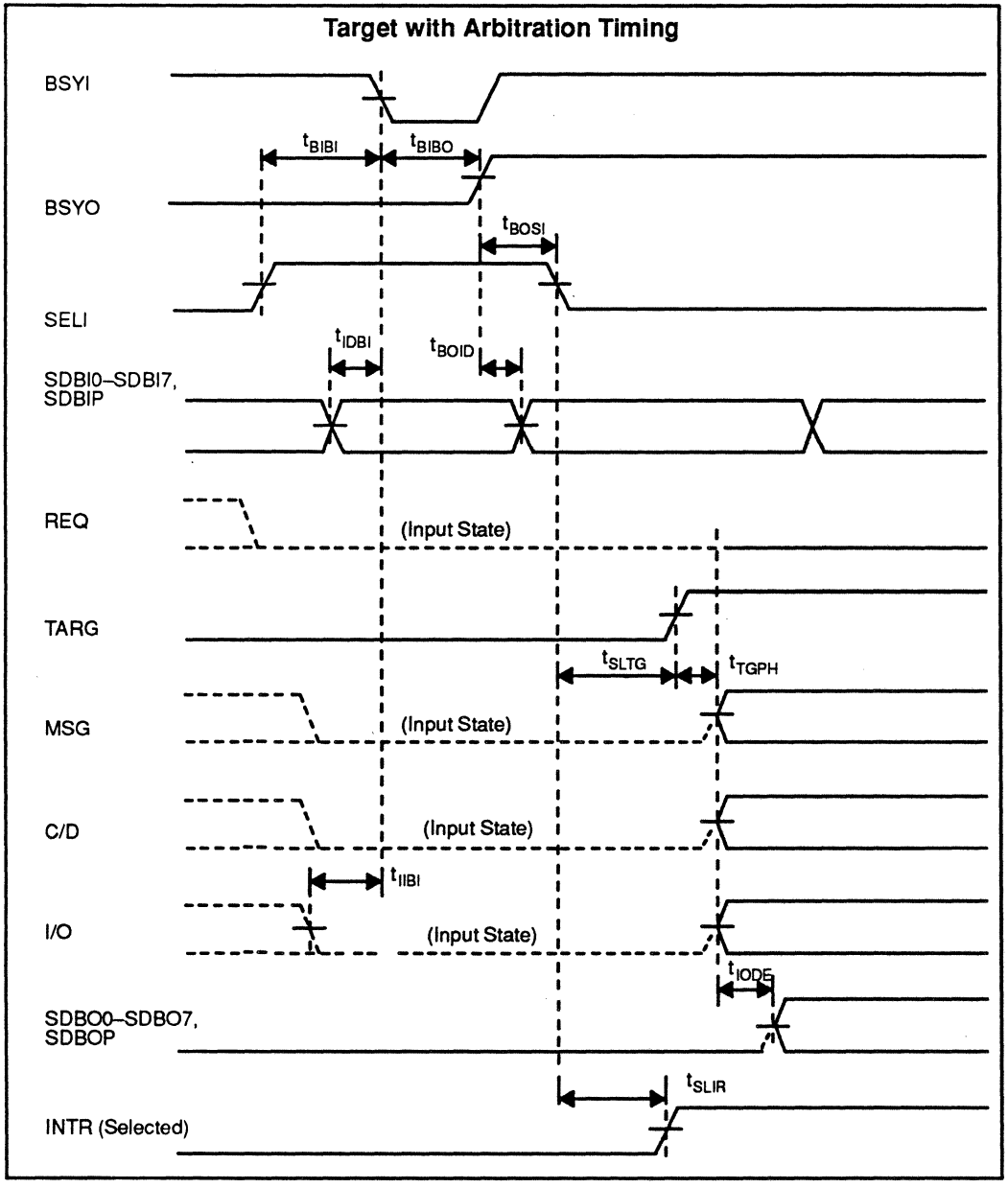
AC CHARACTERISTICS (Continued)



AC CHARACTERISTICS (Continued)

Target with Arbitration						
Parameter	Designator	Test Conditions	Values			Unit
			Min	Typ	Max	
From SELI High to BSYI Low	t_{SIBI}		0			ns
From Data Bus (ID) Valid to BSYI Low	t_{IDBI}		0			ns
From I/O Low to BSYI Low	t_{IIBI}		0			ns
From BSYI to BSYO High	t_{BIBO}	CL = 30pF	$4t_{CLF}$		$5t_{CLF} + 60$	ns
From BSYO High to ID Hold	t_{BOID}		60			ns
From BSYO High to SELI Low	t_{BOSI}		0			ns
From SELI Low to TARG High	t_{SLTG}		$3t_{CLF}$		$4t_{CLF} + 80$	ns
From TARG High to Phase Signal Output	t_{TGPH}		0		50	ns
From I/O High to Data Bus Enable	t_{IODE}		$7t_{CLF}$			ns
From SELI Low to INTR High	t_{SLIR}				$3t_{CLF} + 80$	ns

AC CHARACTERISTICS (Continued)

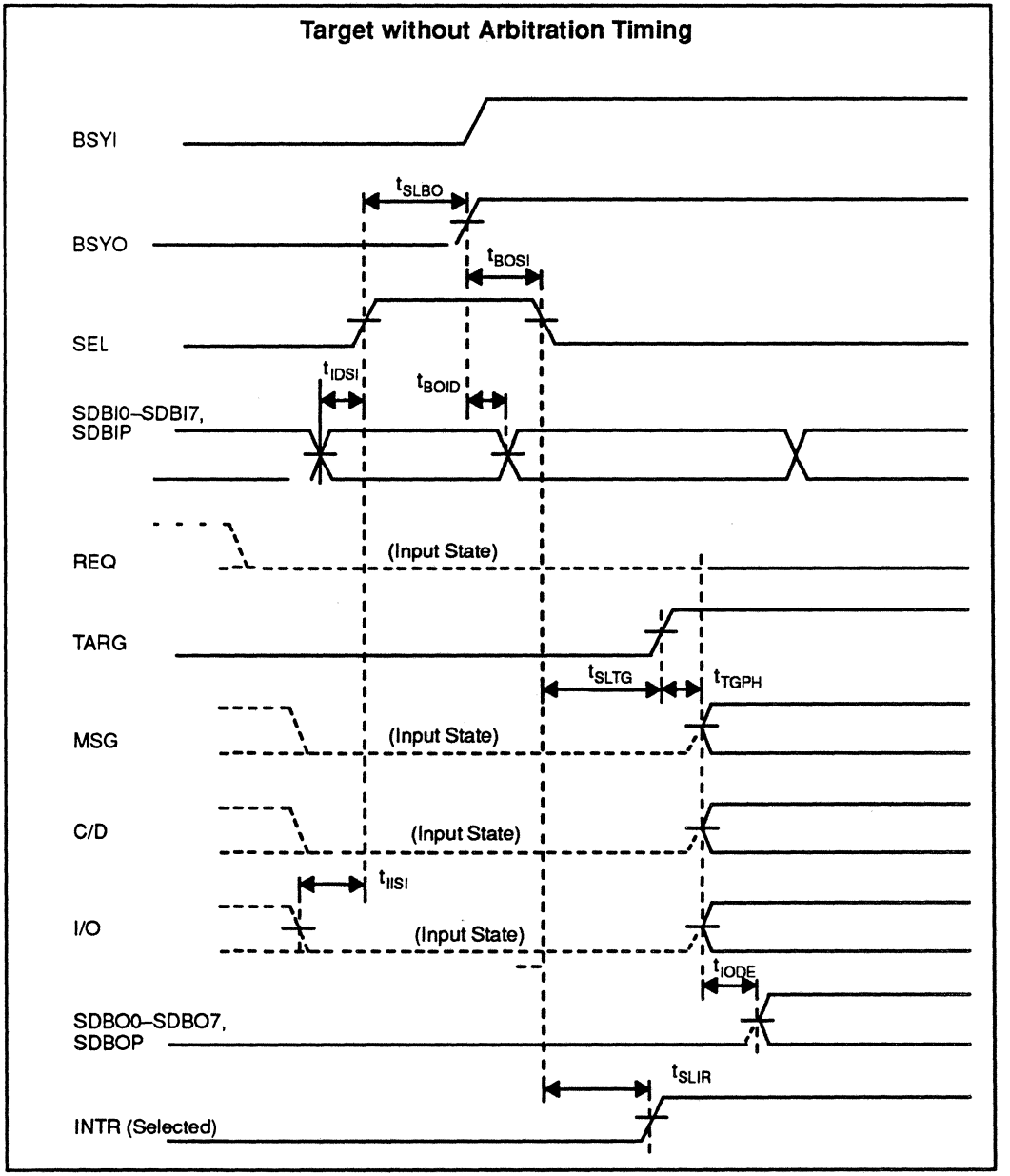


AC CHARACTERISTICS (Continued)

Target without Arbitration						
Parameter	**Designator**	**Test Conditions**	**Values**			**Unit**
			Min.	**Typ.**	**Max.**	
From Data Bus (ID) Valid to SELI High	t_{IDSI}		0			ns
From I/O Low to SELI High	t_{IISI}		0			ns
From SELI High to BSYO High	t_{SLBO}	CL = 30pF	$2t_{CLF}$		$3t_{CLF} + 50$	ns
From BSYO High to ID Hold	t_{BOID}		60			ns
From BSYO High to SELI Low	t_{BOSI}		0			ns
From SELI Low to TARG High	t_{SLTG}	CL = 30pF	$3t_{CLF}$		$4t_{CLF} + 80$	ns
From TARG High to Phase Signal Output	t_{TGPH}	CL = 30pF	0		50	ns
From I/O High to Data Bus Enable	t_{IODE}	CL = 30pF	$7t_{CLF}$			ns
From SELI Low to INTR High	t_{SLIR}	CL = 30pF			$3t_{CLF} + 80$	ns

AC CHARACTERISTICS (Continued)

Target without Arbitration Timing



AC CHARACTERISTICS (Continued)

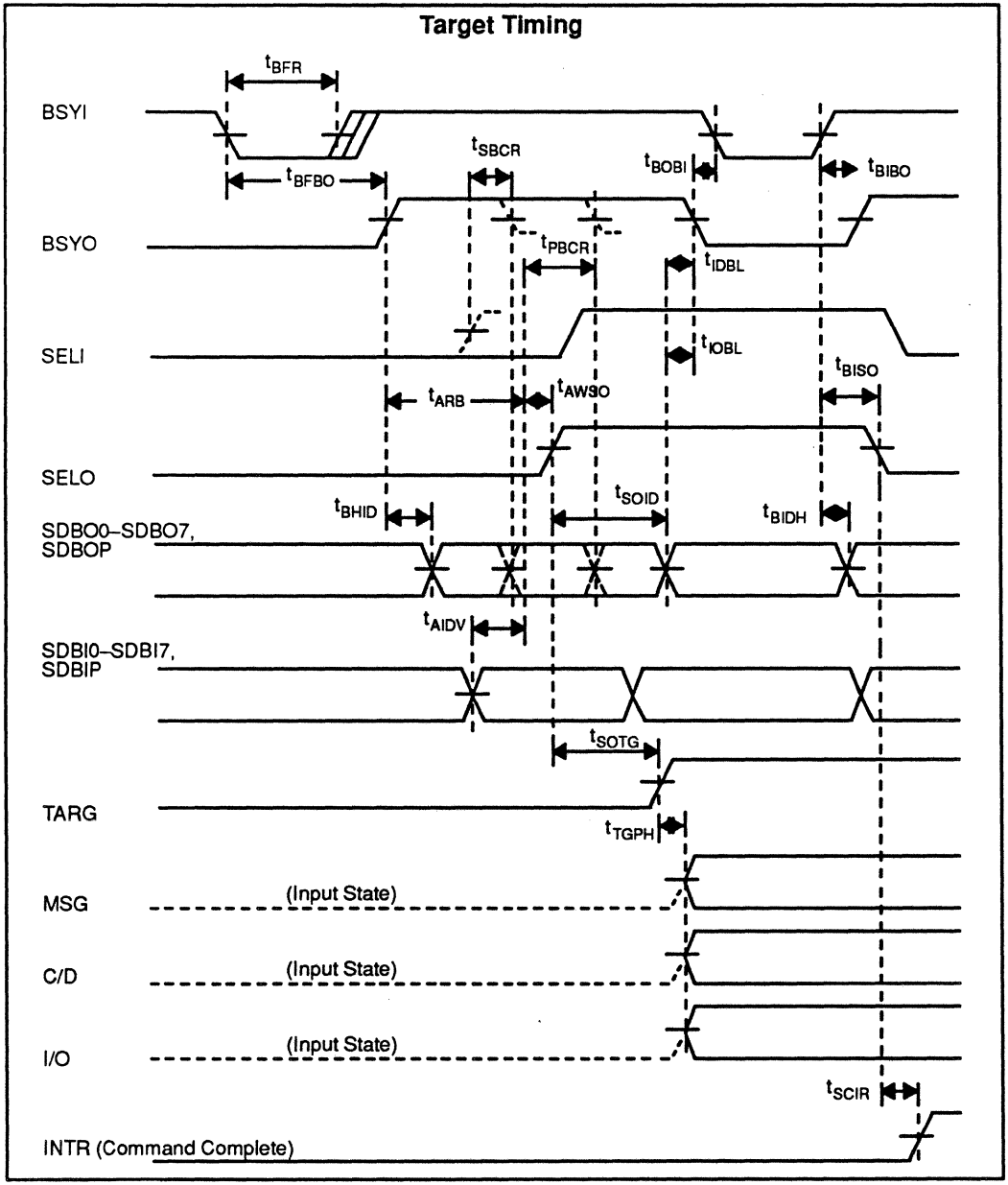
SCSI Bus Interface – Reselection Phase Timing

Target

Parameter	Designator	Test Conditions	Values[1]			Unit
			Min.	Typ.	Max.	
Bus Free Time [2]	t_{BFR}		$4t_{CLF} + 50$			ns
From BSYI Low to BSYO High	t_{BFBO}	CL = 10pF	$(6 + n) \times t_{CLF}$		$(7 + n) \times t_{CLF} + 60$	ns
From BSYO High to Device ID Out	t_{BHID}	CL = 30pF	0		60	ns
From BSYO High to Prioritize	t_{ARB}		$32t_{CLF} - 60$			ns
From Data Bus Valid to Prioritize	t_{AIDV}		100			ns
From Bus Usage Permission Granted to SELO High	t_{AWSO}	CL = 10pF	0		50	ns
From SELO High to RESELECT ID Output	t_{SOID}		$11t_{CLF} - 30$			ns
From SELO High to TARG High	t_{SOTG}		$11t_{CLF} - 50$			ns
From TARG High to Phase Signal Output	t_{TGPH}	CL = 30pF	0		50	ns
From I/O High to BSYO Low	t_{IOBL}	CL = 10pF	$2t_{CLF} - 80$			ns
From RESELECT ID Output to BSYO Low	t_{IDBL}	CL = 10pF	$2t_{CLF} - 80$			ns
From BSYO Low to BSYI Low	t_{BOBI}	CL = 10pF	0		t_{CLF}	ns
From BSYI High to SELO Low	t_{BISO}	CL = 10pF	$3t_{CLF}$			ns
From BSYI High to RESELECT ID Hold	t_{BIDH}	CL = 10pF	$2t_{CLF}$			ns
From SELO Low to INTR High	t_{SCIR}	CL = 30pF			80	ns
From SELI High to BSYO and ID Bit Low	t_{SBCR}	CL = 30pF (BSYO) CL = 30pF (SDBO0–SDBO7, SDBOP)			$3t_{CLF} + 80$	ns
From Prioritze to BSYO and ID Bit Low	t_{PBCR}	CL = 30pF (BSYO) CL = 30pF (SDBO0–SDBO7, SDBOP)			60	ns
From BSYI High to BSYI High	t_{BIBO}	CL = 10pF	$2t_{CLF} + 20$		$3t_{CLF} + 60$	ns

NOTES:
1. n = value of TCL register.
2. The bus free time is the minimum time interval until the booked select command is executed.

AC CHARACTERISTICS (Continued)

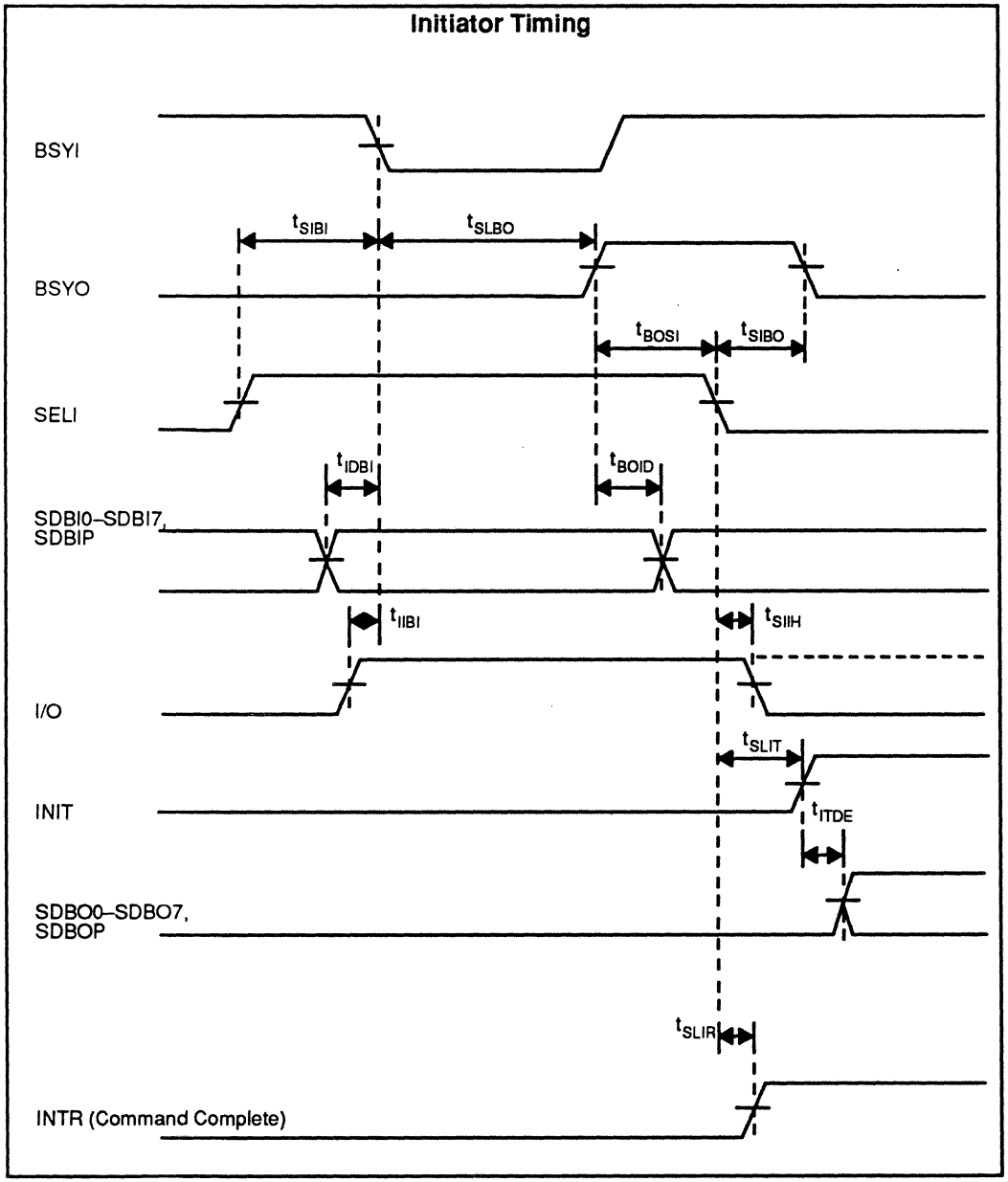


AC CHARACTERISTICS (Continued)

Initiator

Parameter	Designator	Test Conditions	Values			Unit
			Min	Typ	Max	
From SELI High to BSYI Low	t_{SIBI}		0			ns
From Data Bus (ID) Valid to BSYI Low	t_{IDBI}		0			ns
From I/O High to BSYI Low	t_{IIBI}		0			ns
From BSYI Low to BSYO High	t_{SLBO}	CL = 30pF	$4t_{CLF}$		$5t_{CLF} + 60$	ns
From BSYO High to ID Hold	t_{BOID}		60			ns
From BSYO High to SELI Low	t_{BOSI}		0			ns
From SELI Low to BSYO Low	t_{SIBO}	CL = 30pF	$2t_{CLF}$		$3t_{CLF} + 60$	ns
From SELI Low to I/O Hold	t_{SIIH}		100			ns
From SELI Low to INTR High	t_{SLIR}	CL = 30pF			$3t_{CLF} + 80$	ns
From SELI Low to INIT High	t_{SLIT}	CL = 30pF	$3t_{CLF} + 30$		$4t_{CLF} + 80$	ns
From INIT High to Data Bus Enable when I/O is Low	t_{ITDE}	CL = 30pF			50	ns

AC CHARACTERISTICS (Continued)



AC CHARACTERISTICS (Continued)

SCSI Bus Interface – Transfer Phase Timing

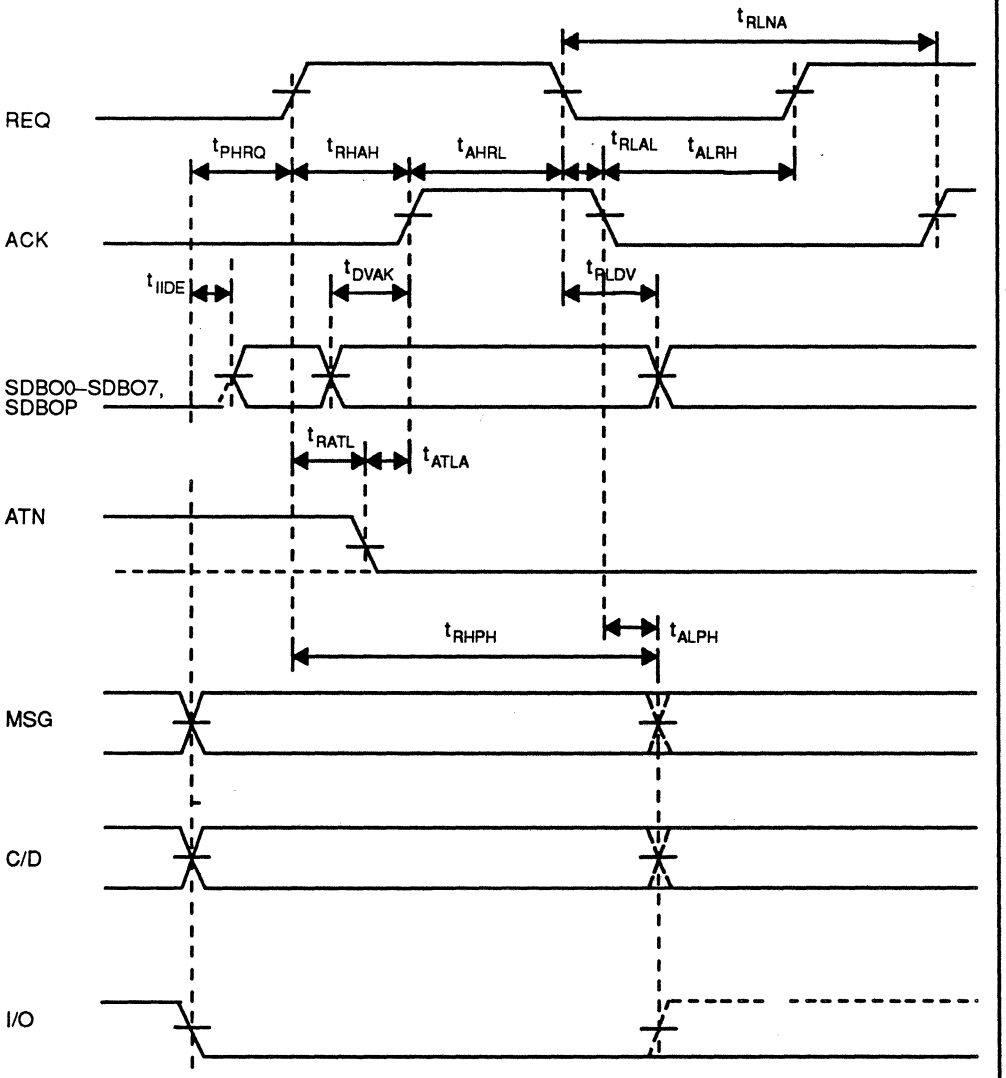
Asynchronous Transfer Output (Initiator)

Parameter	Designator	Test Conditions	Values			Unit
			Min.	Typ.	Max.	
From I/O Low to Data Bus Enable	t_{IIDE}	CL = 10pF	10			ns
From Phase Specify to REQ High	t_{PHRQ}		100			ns
From ACK Low to Phase Change	t_{ALPH} [1]	CL = 10pF	10			ns
From REQ High to ATN Low	t_{RATL} [2]		$2t_{CLF}$			ns
From ATN Low to ACK High	t_{ATLA} [2]		$t_{CLF}-20$			ns
From Data Bus Valid to ACK High	t_{DVAK}	CL = 10pF	$2t_{CLF}-80$			ns
From REQ Low to Data Bus Hold	t_{RLDV}	CL = 10pF	15			ns
From REQ High to ACK High	t_{RHAH}	CL = 10pF	20			ns
From ACK High to REQ Low	t_{AHRL}		0			ns
From REQ Low to ACK Low	t_{RLAL}	CL = 10pF	10			ns
From ACK Low to REQ High	t_{ALRH}		10			ns
From REQ Low to ACK High	t_{RLNA}	CL = 10pF	$2t_{CLF}$			ns
From REQ High to Phase Change	t_{RHPH} [1]		$3t_{CLF}$			ns

NOTES:
1. Phase change must satisfy both t_{ALPH} and t_{RHPH} specifications.
2. This specification is applicable only when the last byte of the message transfer phase is transferred using the hardware transfer mode.

AC CHARACTERISTICS (Continued)

Asynchronous Transfer Output (Initiator) Timing

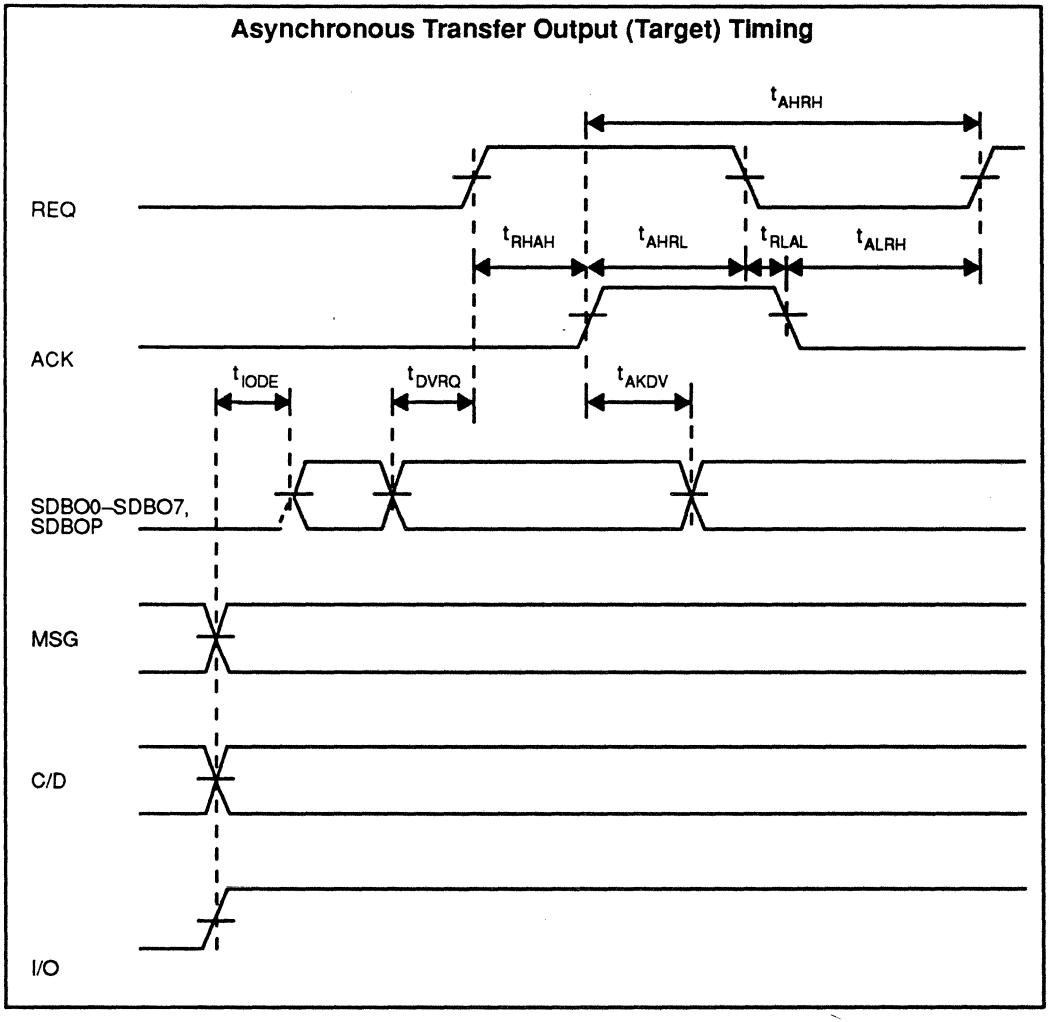


AC CHARACTERISTICS (Continued)

Asynchronous Transfer Output (Target)						
Parameter	Designator	Test Conditions	Values			Unit
			Min	Typ	Max	
From I/O High to Data Bus Enable	t_{IODE}	CL = 10pF	$7t_{CLF}$			ns
From Data Bus Valid to REQ High	t_{DVRQ}	CL = 10pF	$2t_{CLF}-80$			ns
From ACK High to Data Bus Hold	t_{AKDV}	CL = 10pF	15			ns
From REQ High to ACK High	t_{RHAH}		20			ns
From ACK High to REQ Low	t_{AHRL}	CL = 30pF	10		100	ns
From REQ Low to ACK Low	t_{RLAL}		0			ns
From ACK Low to REQ High	t_{ALRH}	CL = 10pF	10			ns
From ACK High to REQ High	t_{AHRH}	CL = 10pF	$2t_{CLF}$			ns

AC CHARACTERISTICS (Continued)

Asynchronous Transfer Output (Target) Timing



AC CHARACTERISTICS (Continued)

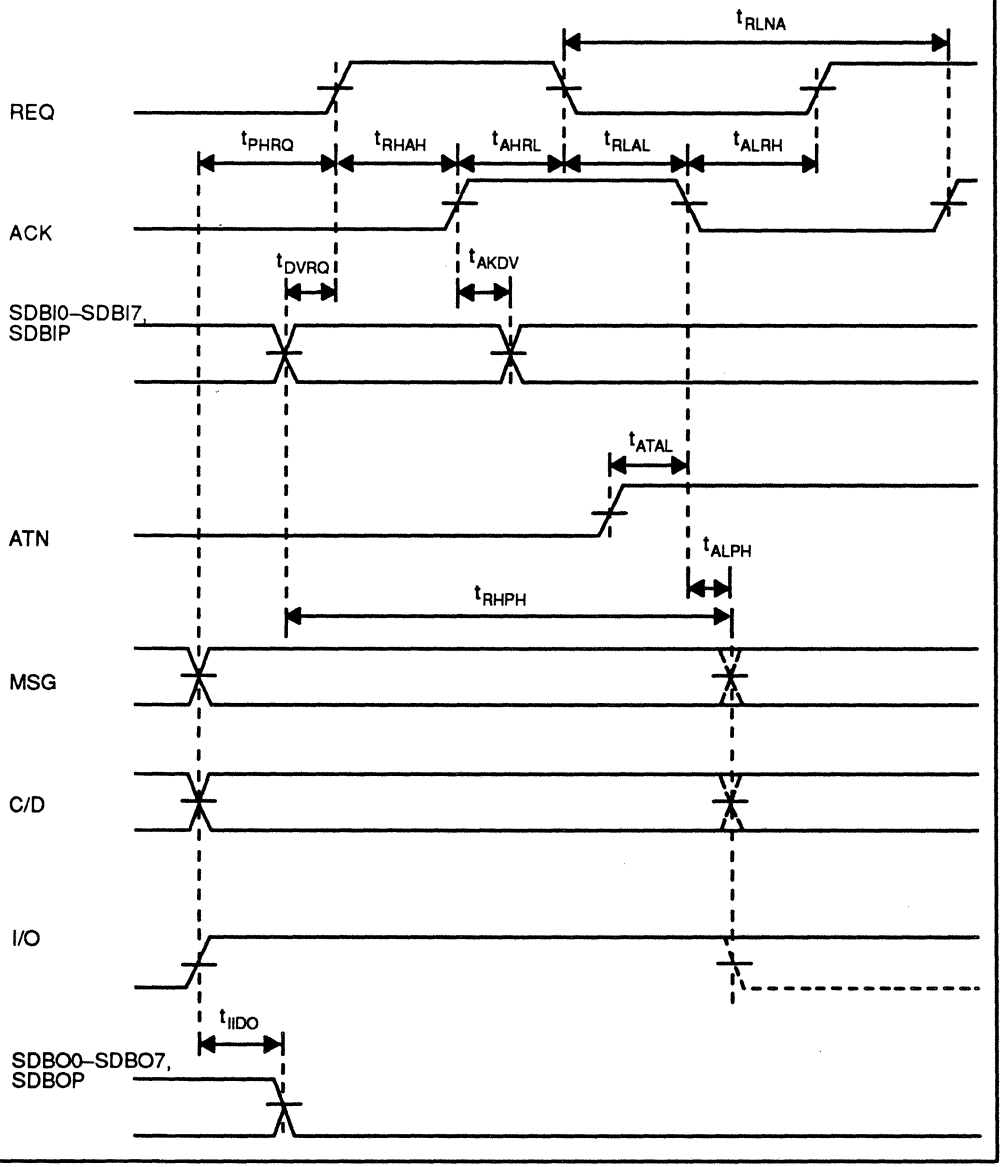
Asynchronous Transfer Input (Initiator)

Parameter	Designator	Test Conditions	Values			Unit
			Min.	Typ.	Max.	
From I/O High to Data Bus Disable	t_{IIDD}	CL = 30pF			60	ns
From Phase Specify to REQ High	t_{PHRQ}		100			ns
From ACK Low to Phase Change	t_{ALPH} [1]		10			ns
From Data Bus Valid to REQ High	t_{DVRQ}		10			ns
From ACK High to Data Bus Hold	t_{AKDV}		15			ns
From REQ High to ACK High	t_{RHAH}	CL = 10pF	20			ns
From ACK High to REQ Low	t_{AHRL}		0			ns
From REQ Low to ACK Low	t_{RLAL}	CL = 10pF	20			ns
From ACK Low to REQ High	t_{ALRH}		10			ns
From REQ Low to ACK High	t_{RLNA}	CL = 10pF	t_{CLF}			ns
From ATN High to ACK Low	t_{ATAL} [2]	CL = 10pF	t_{CLF} – 20			ns
From REQ High to Phase Change	t_{RHPH} [1]		$3t_{CLF}$			ns

NOTES:
1. Phase change must satisfy both t_{ALPH} and t_{RHPH} specifications.
2. Based on this timing parameter, the ATN signal is transferred only when parity check function is enabled and a parity error is detected on the input data.

AC CHARACTERISTICS (Continued)

Asynchronous Transfer Input (Initiator) Timing

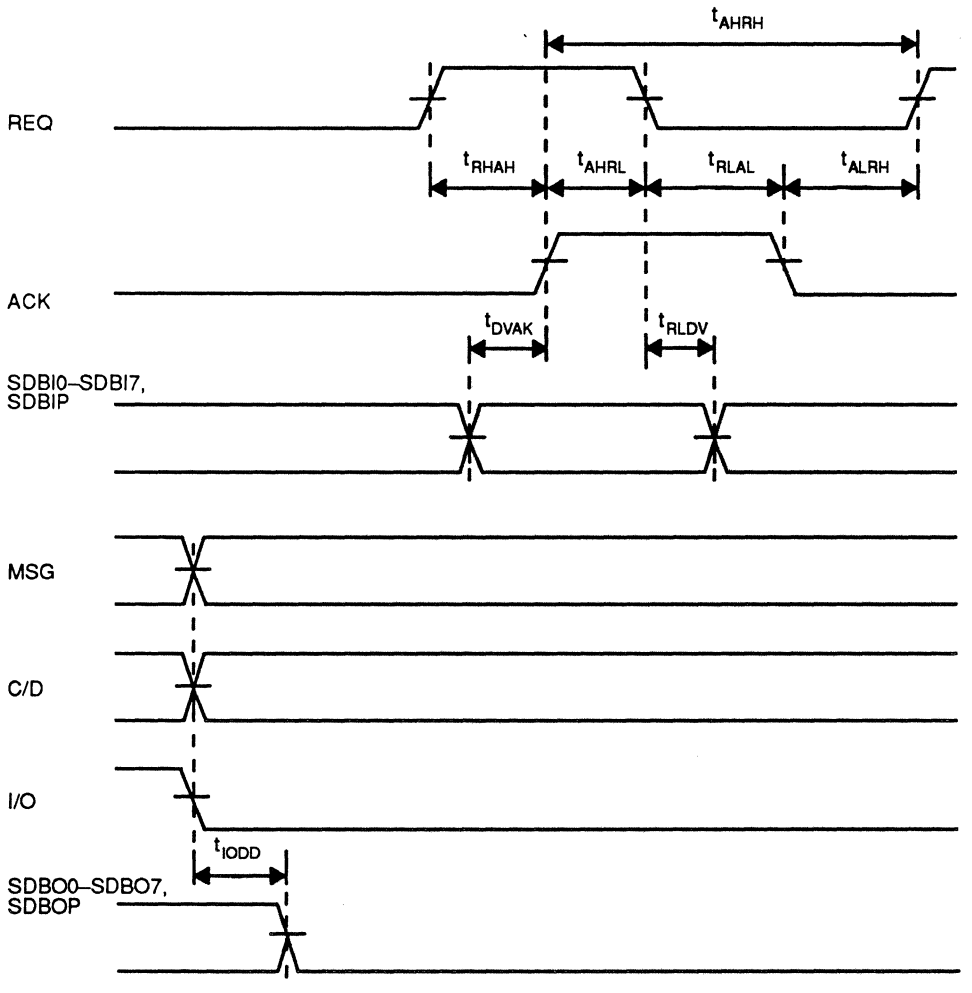


AC CHARACTERISTICS (Continued)

Asynchronous Transfer Input (Target)						
Parameter	Designator	Test Conditions	Values			Unit
			Min.	Typ.	Max.	
From I/O Low to Data Bus Disable	t_{IODD}	CL = 30pF			30	ns
From Data Bus Valid to ACK High	t_{DVAK}		10			ns
From REQ Low to Data Bus Hold	t_{RLDV}	CL = 10pF	15			ns
From REQ High to ACK High	t_{RHAH}		20			ns
From ACK High to REQ Low	t_{AHRL}	CL = 30pF	10		100	ns
From REQ Low to ACK Low	t_{RLAL}		0			ns
From ACK Low to REQ High	t_{ALRH}	CL = 10pF	10			ns
From ACK High to REQ High	t_{AHRH}	CL = 10pF	$2t_{CLF}$			ns

AC CHARACTERISTICS (Continued)

Asynchronous Transfer Input (Target) Timing

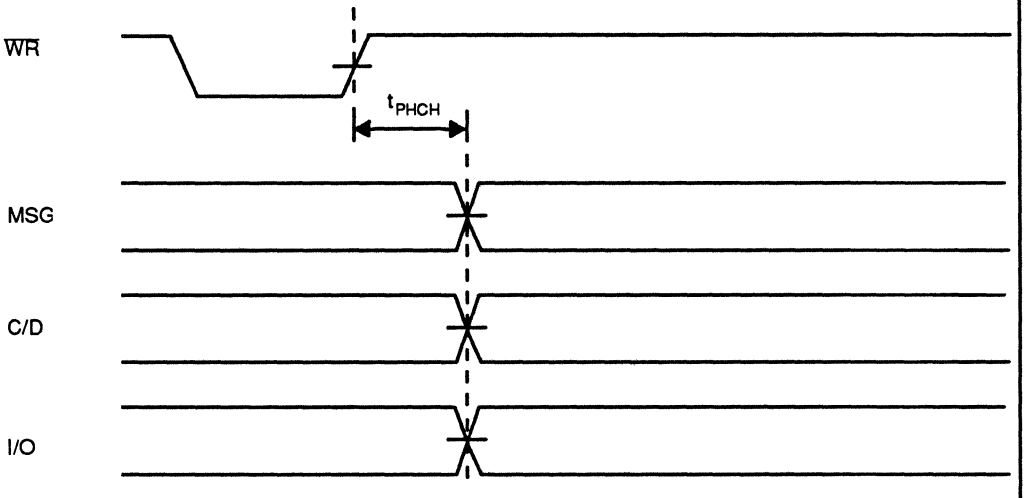


AC CHARACTERISTICS (Continued)

Transfer Phase Change (Target)

Parameter	Designator	Test Conditions	Values			Unit
			Min.	Typ.	Max.	
From WR High to MSG, C/D, I/O	t_{PHCH}	CL = 30pF	10		100	ns

Transfer Phase Change (Target) Timing



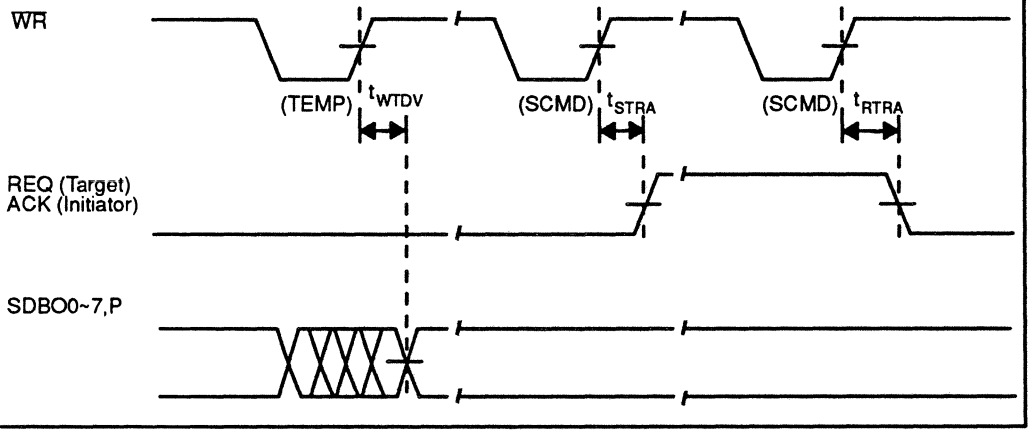
AC CHARACTERISTICS (Continued)

Manual Transfer (Note)

Parameter	Designator	Test Conditions	Values			Unit
			Min.	Typ.	Max.	
From $\overline{WR}$ High to Data Bus Valid for TEMP Register	t_{WTDV}	CL = 30pF			100	ns
From $\overline{WR}$ High to REQ High, ACK High for SET ACK/REQ Command	t_{STRA}	CL = 30pF	$2t_{CLF}$		$3t_{CLF} + 60$	ns
From $\overline{WR}$ High to REQ Low, ACK Low for RESET ACK/REQ Command	t_{RTRA}	CL = 30pF	$2t_{CLF}$		$3t_{CLF} + 60$	ns

NOTE: Timing sequences not shown are the same as those for asynchronous transfers.

Manual Transfer Timing

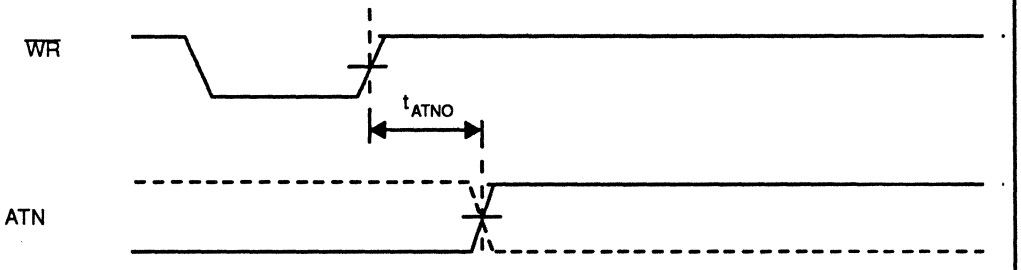


AC CHARACTERISTICS (Continued)

SCSI Bus Interface – Attention Condition

Initiator						
Parameter	Designator	Test Conditions	Values			Unit
			Min.	Typ.	Max.	
From $\overline{WR}$ high to ATN High, ATN Low for SET/RESET ATN Command	t_{ATNO}	CL = 30pF	$2t_{CLF}$		$3t_{CLF} + 60$	ns

Attention Condition (Initiator) Timing



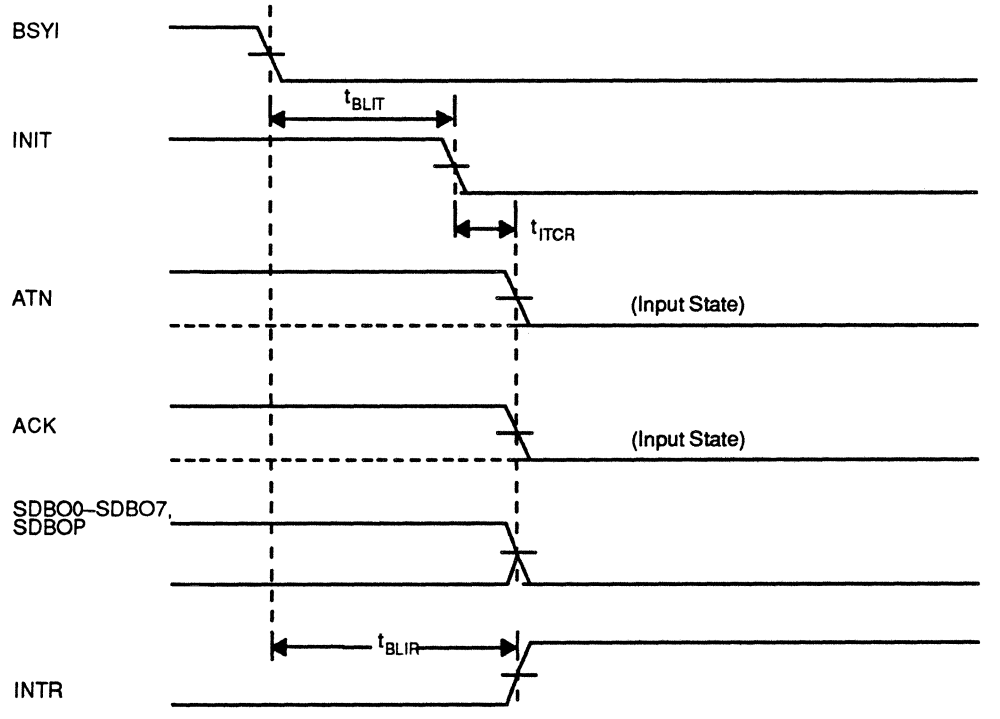
AC CHARACTERISTICS (Continued)

SCSI Bus Interface – Bus Free

Initiator (Disconnected)

Parameter	Designator	Test Conditions	Values			Unit
			Min	Typ	Max	
From BSYI Low to INIT Low	t_{BLIT}	CL = 30pF			$5t_{CLF} + 60$	ns
From INIT Low to Bus Clear	t_{ITCR}	CL = 30pF			80	ns
From BSYI Low to INTR High	t_{BLIR}	CL = 30pF			$6t_{CLF} + 80$	ns

Initiator (Disconnected) Timing

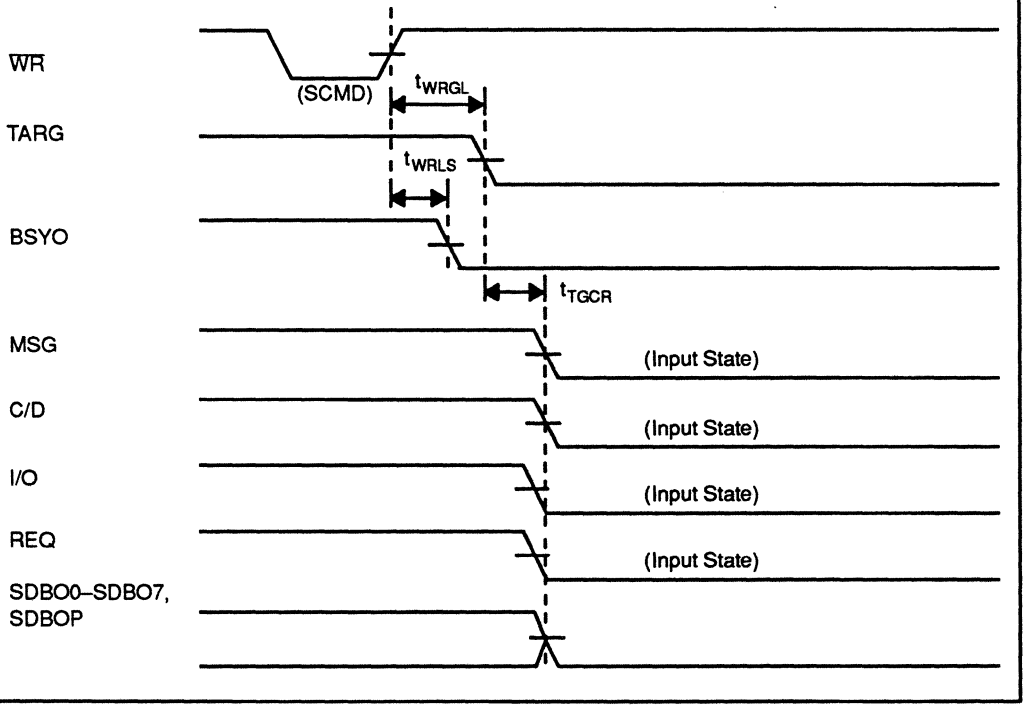


AC CHARACTERISTICS (Continued)

Target (Bus Release Command)

Parameter	Designator	Test Conditions	Values			Unit
			Min.	Typ.	Max.	
From $\overline{WR}$ High (SCMD Register) to BSYO Low	t_{WRLS}	CL = 30pF			3t $_{CLF}$ + 60	ns
From $\overline{WR}$ High (SCMD Register) to TARG Low	t_{WRGL}	CL = 40pF			3t $_{CLF}$ + 60	ns
From TARG Low to Bus Clear	t_{TGCR}	CL = 30pF			80	ns

Target (Bus Release Command) Timing

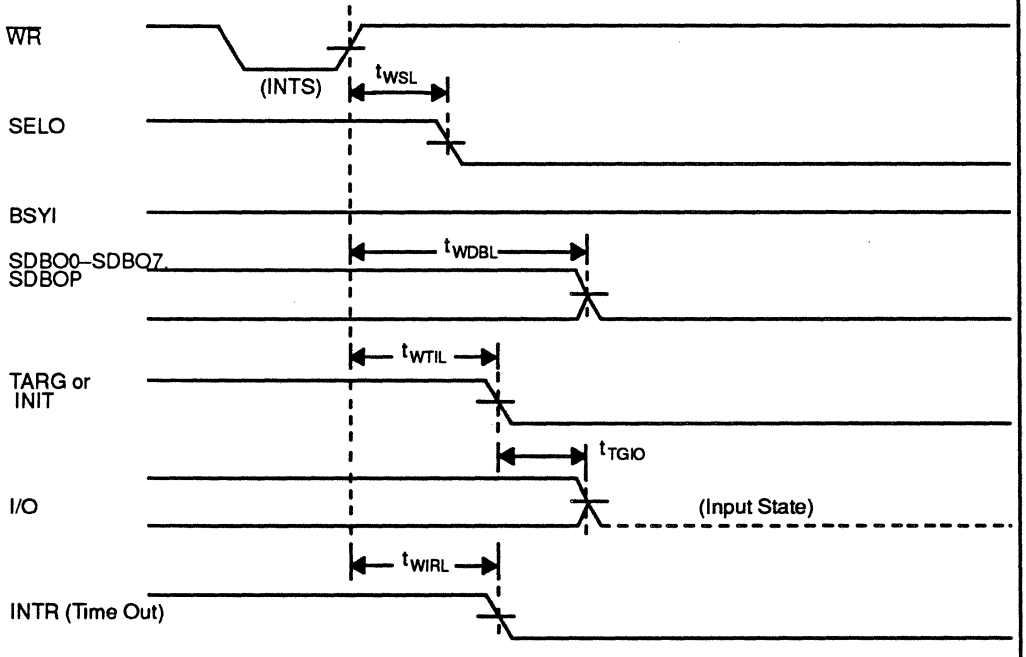


AC CHARACTERISTICS (Continued)

Termination (Time Out) – Selection and Reselection Phases

Parameter	Designator	Test Conditions	Values			Unit
			Min.	Typ.	Max.	
From $\overline{WR}$ High (INTS Register) to SELO low	t_{WSL}	CL = 30pF			3t CLF + 60	ns
From $\overline{WR}$ High (INTS Register) to Data Bus Disable	t_{WDBL}	CL = 30pF			3t CLF + 100	ns
From $\overline{WR}$ High (INTS Register) to TARG Low or INIT Low	t_{WTIL}	CL = 40pF			3t CLF + 60	ns
From TARG Low to I/O High–Z	t_{TGIO}	CL = 30pF			50	ns
From $\overline{WR}$ High (INTS Register) to Data Bus Disable	t_{WIRL}	CL = 30pF			3t CLF + 60	ns

Termination (Time Out) Selection and Reselection Phases Timing

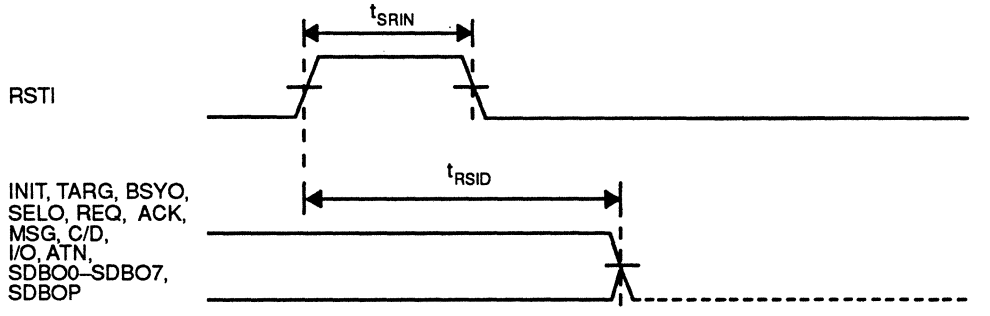


AC CHARACTERISTICS (Continued)

SCSI Bus Interface – Reset Condition

RST INPUT						
Parameter	**Designator**	**Test Conditions**	**Values**			**Unit**
			Min.	**Typ.**	**Max.**	
RSTI Pulse Width	t_{SRIN}		$3t_{CLF}$			ns
Reset Delay	t_{RSID}	CL = 30pF			$4t_{CLF}$ + 110	ns

RSTI Input Timing

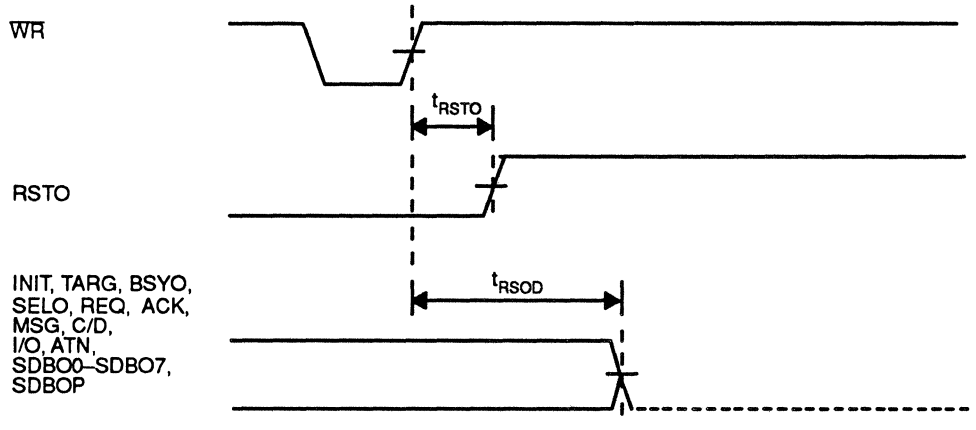


AC CHARACTERISTICS (Continued)

RST Output

Parameter	Designator	Test Conditions	Values			Unit
			Min.	Typ.	Max.	
From WR High (SCMD Register's Bit 4) to RSTO High	t_{RSTO}	CL = 30pF	10		80	ns
Reset Delay	t_{RSOD}	CL = 30pF			110	ns

RSTO Output Timing



AC CHARACTERISTICS (Continued)

AC Test Conditions (Input)

Timing Reference Levels for CPU/DMAC Interface

Logical 1 = 2.4 Vdc
Logical 0 = 0.45 Vdc

AC Test Conditions (Output)

Timing Reference Levels for SCSI Bus Interface

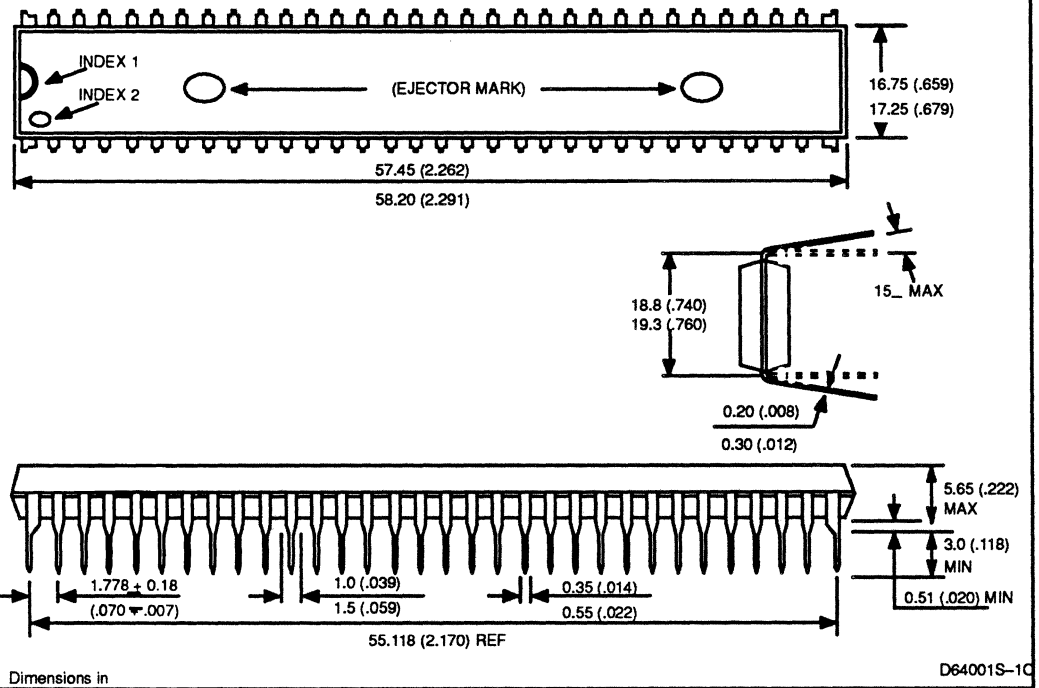
Logical 1 = 2.4 Vdc
Logical 0 = 0.45 Vdc

Capacitive Output Loading

Input/Output Pins	Values			Unit
	Min.	Typ.	Max.	
D0 – D7, DP		—	80	pF
DPO		10	30	pF
INTR		10	30	pF
DREQ		10	30	pF
TARG, INPUT1		20	40	pF
INIT, INPUT2		10	30	pF
SDBO0 – SDBO7, SDBOP		10	30	pF
RSTO, SELO, BSYO		10	30	pF
MSG, C/D, I/O		10	30	pF
REQ, ACK, ANT		10	30	pF

DIP–64P–M01

64–LEAD PLASTIC DUAL IN–LINE PACKAGE
(CASE NO.: DIP–64P–M01)

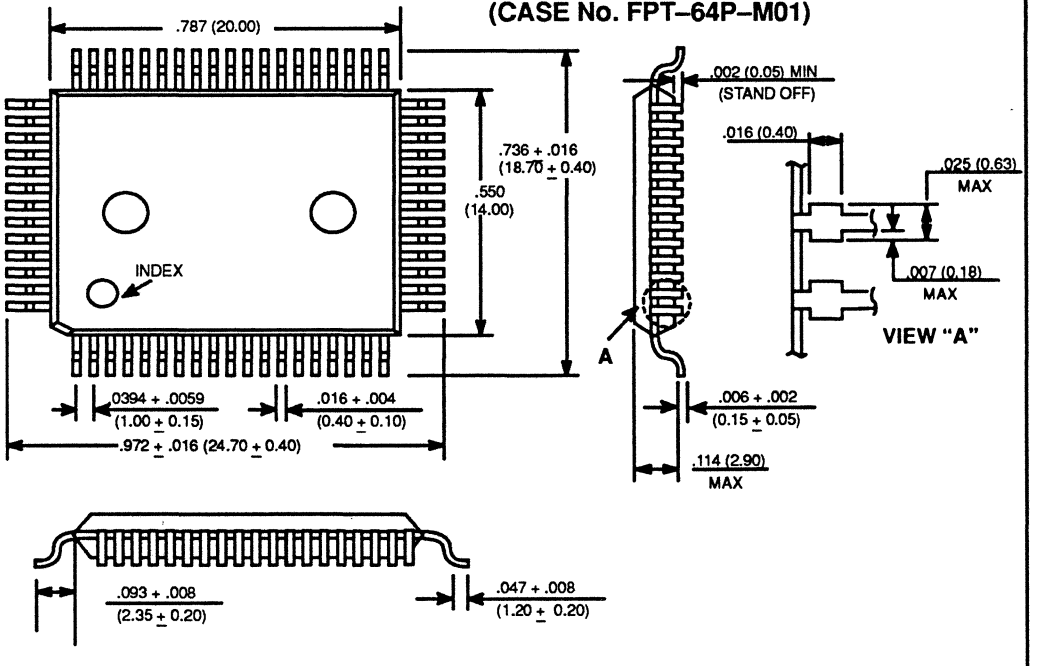


Dimensions in inches (millimeters)

TBD

FPT–64P–M01

64–LEAD PLASTIC FLAT PACKAGE
(CASE No. FPT–64P–M01)



Dimensions in
inches (millimeters)

F64005S–5C

MB89352
SCSI Protocol Controller (SPC) with On-Chip Drivers/Receivers

GENERAL DESCRIPTION

The MB89352 CMOS LSI SPC (SCSI Protocol Controller) is a circuit designed to make control of the small computer system interface (SCSI) easier.

The MB89352 can be used as a peripheral LSI circuit for an 8- or 16-bit MPU to realize high-level SCSI control. The SPC can control all the SCSI interface signals, handle almost all the interface control procedures and the on-chip driver/receivers allow for direct connection to the SCSI BUS.

This LSI circuit has an 8-byte FIFO data buffer register and a transfer byte counter that is 24 bits long. Furthermore, the MB89352 can serve as either an INITIATOR or a TARGET device for the SCSI, and can therefore be used for either an I/O controller or a host adapter.

SCSI Compatibility

- Full support for SCSI control (ANSI X3.131S •1986 Specification) except for synchronized transfer mode
- Serves as either INITIATOR or TARGET

Data Transfer Rate/Byte Counter

- 8-byte FIFO data timing control
- 24-bit transfer byte counter

Drive Options (on-chip driver/receiver)

- Single -ended

Selectable Transfer Modes

- DMA Transfer
- Program Transfer
- Manual Transfer

Clock Requirements

- 8 MHz clock

Technology/Power Requirements

- Silicon-gate CMOS
- Single +5V power supply

Available Packaging

- 48-pin DIP or FLAT plastic packages

ABSOLUTE MAXIMUM RATINGS[1]

Rating	Designator	Values		Unit
		Min.	Max.	
Supply Voltage	V_{CC}	V_{SS} –.03	7.0	V
Input Voltage	V_I	V_{SS} –.03	7.3	V
Output Voltage[2]	V_O	V_{SS} –.03	V_{SS} + 7.3	V
Operating Ambient Temperature	T_A	0	70	°C
Storage Temperature	T_{STG}	–55	150	°C

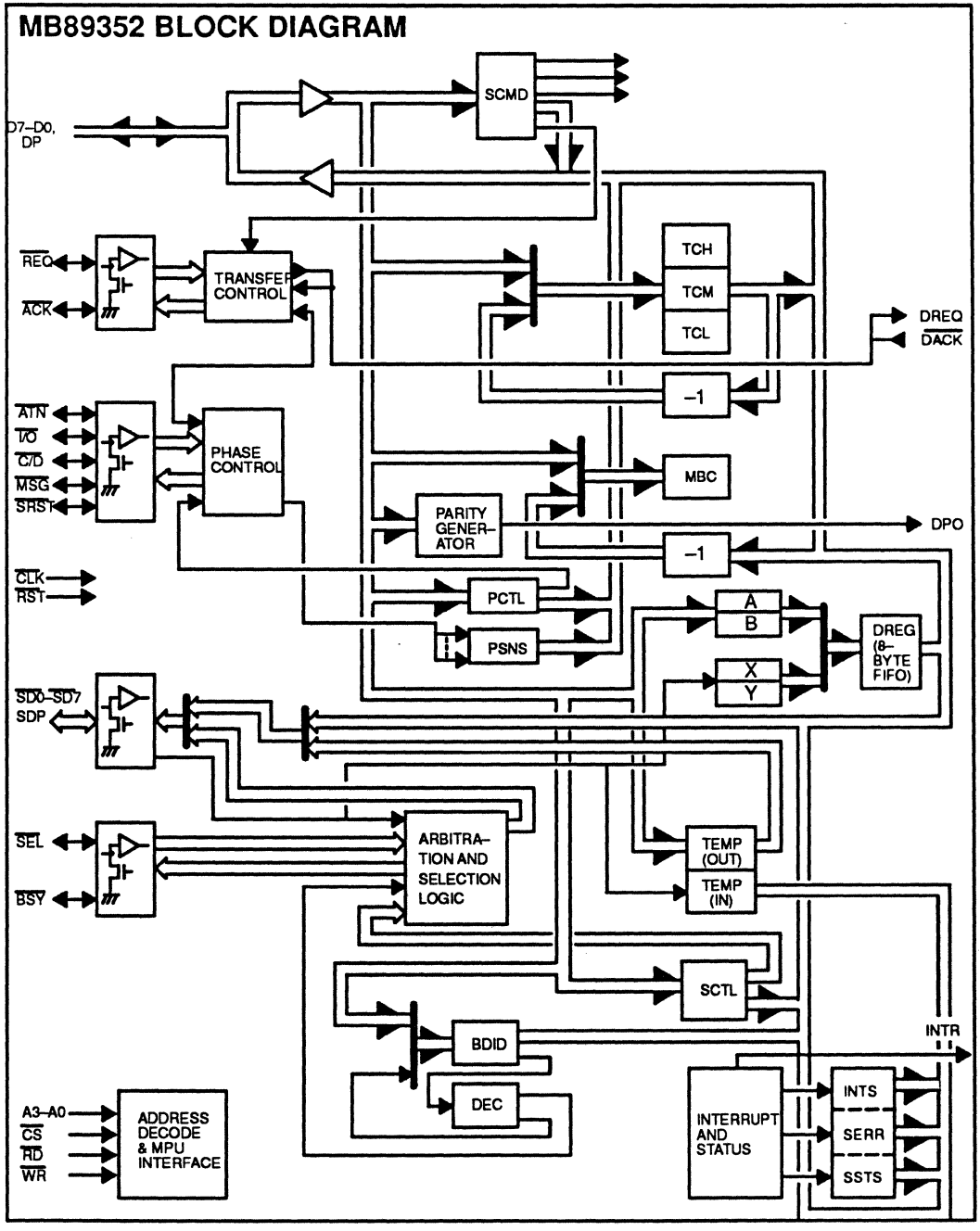
NOTES:

1. Permanent device damage may occur if the above absolute maximum ratings are exceeded. Functional operation should be restricted to the conditions as detailed in the operational sections of this data sheet. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.
2. Should not exceed V_{CC} + 0.5V.

RECOMMENDED OPERATING CONDITIONS

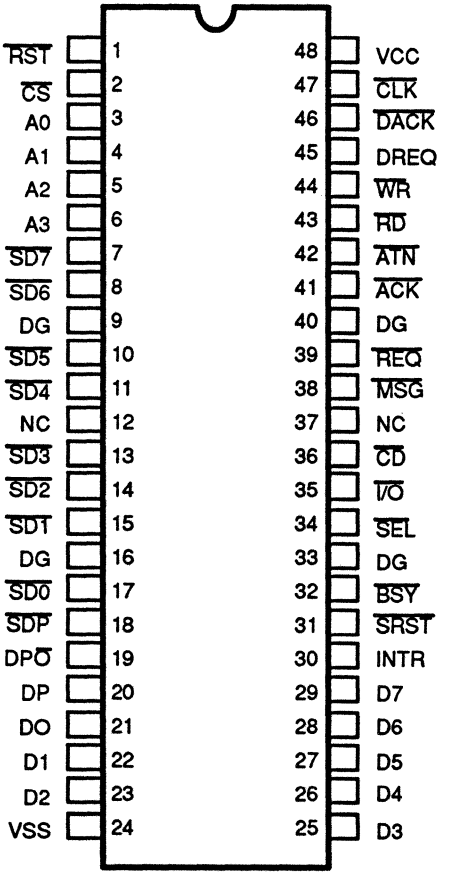
Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
Supply Voltage	V_{CC}	4.75	5.0	5.25	V
Operating Ambient Temperature	T_A	0		+70	°C

MB89352 BLOCK DIAGRAM



PIN ASSIGNMENTS

48-Pin DIP

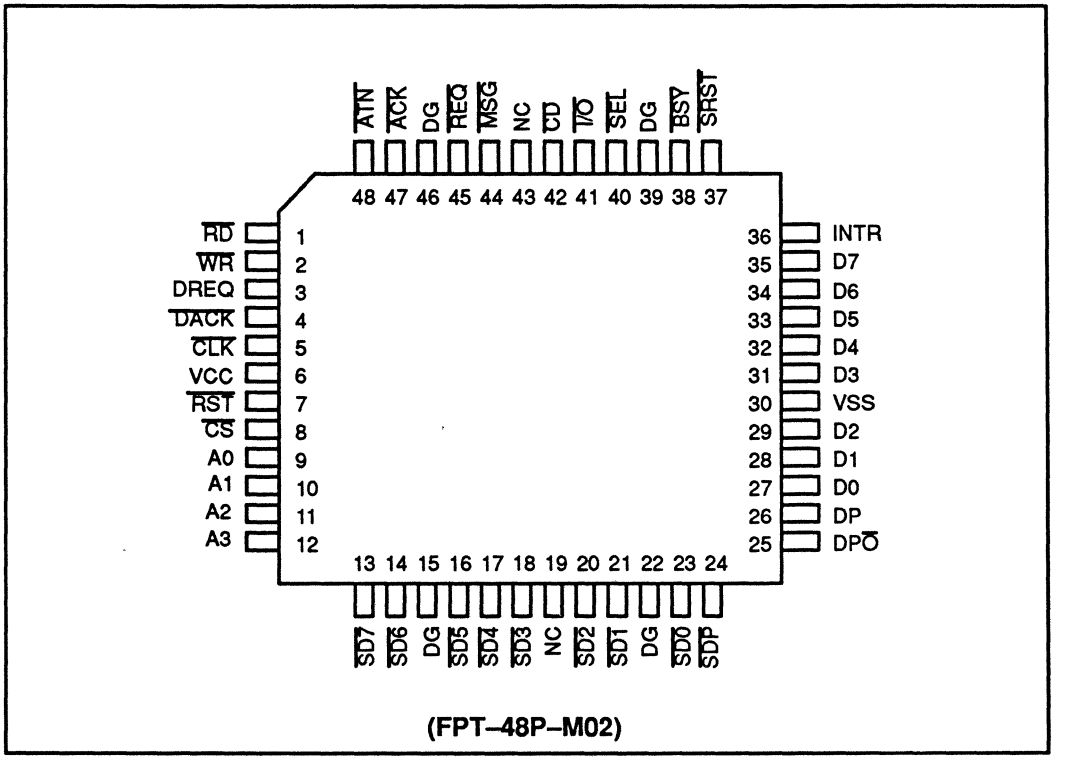


(DIP–48P–M02)

PIN ASSIGNMENTS (Continued)

48-Pin DIP

48-Pin Plastic Flat Package



(FPT–48P–M02)

PIN DESCRIPTIONS

Designator	Pin No.		I/O	Function
	DIP	FPT		
V_{CC}	48	6	—	+5V power supply.
V_{SS}	24	30	—	Circuit ground.
DG	9 16 33 40	15 22 39 46	—	Ground (OV) for internal drivers. The SCSI bus drivers can sink up to 48-mA each. Up to 16 drivers can be active at once. We recommend a good solid ground plane.
$\overline{CLK}$	47	5	I	Clock input for controlling internal operation and data transfer speed of the SPC.
$\overline{RST}$	1	7	I	Asynchronous reset signal used to clear all internal circuits of the SPC.
$\overline{CS}$	2	8	I	Input selection enable signal for accessing an internal register. When active low, the following input/output signals are valid: RD, WR, A3–A0, DP7–DP0 and DP.
A0 A1 A2 A3	3 4 5 6	9 10 11 12	I	Address input signals for selecting an internal register in SPC. MSB is A3; LSB is A0. When $\overline{CS}$ is active, read/ write is enabled for an internal register selected by these address inputs via data bus lines D0–D7 and DP.
$\overline{RD}$	43	1	I	This strobe input is used for reading out the contents of the SPC internal register, and is effective only when $\overline{CS}$ input is active. While $\overline{RD}$ is active, the contents of an internal register selected by A0 to A3 inputs are placed on data bus lines D7 to D0, DP. For a data transfer cycle in the program transfer mode, the rising edge of $\overline{RD}$ is used as a timing signal indicating the end of data read.
$\overline{WR}$	44	2	I	The strobe input is used for writing data into an SPC internal register, and is effective only when $\overline{CS}$ input is active. On the rising edge of this signal, data placed on data bus lines D0 to D7, DP are loaded into an internal register selected by A0 to A3 inputs. For a data transfer cycle in the program transfer mode, the rising edge of this signal is used as a timing signal indicating data ready status.
D0 D1 D2 D3 D4 D5 D6 D7	21 22 23 25 26 27 28 29	27 28 29 31 32 33 34 35	I/O	Used to write/read data to/from an internal register in the SPC. The data bus is 3-state and bidirectional. The MSB is D7 and the LSB is D0; DP is an odd parity bit. When both $\overline{CS}$ and RD inputs are active, the contents of a selected internal register are output to the data bus. In operations other than read/write, the data bus is kept at a high–Z level.

PIN DESCRIPTIONS (Continued)

Designator	Pin No.		I/O	Function
	DIP	FPT		
DPO	19	25	O	Outputs an odd parity of D0–D7. If parity bit is not generated for external memory, DPO can be used as an input parity bit for DP.
INTR	30	36	O	The INTR output signal is issued by the SPC and requests an interrupt to indicate completion of an internal operation or the occurrence of an error. Except for an interrupt caused by the RSTI input (reset condition in SCSI). When an interrupt request is granted, the INTR signal remains active until the interrupt is cleared.
DREQ	45	3	O	For a data transfer cycle in DMA mode, this signal is used to indicate a request for data transfer between the SPC and the external buffer memory. In an output operation, this signal becomes active to request a data transfer from the external buffer memory when the SPC internal data buffer register has free space available. In an input operation, it becomes active to request data transfer to the external buffer memory when the SPC internal data buffer register contains valid data.
$\overline{DACK}$	46	4	I	An active low response signal to the DREQ which request data transfer in between SPC and the external memory in the DMA mode. This signal in DMA mode functions similarly to the signal combination of $\overline{CS}$=low, A3=high, A2=low, A1–high, and A0=low (selection of DREG) in the program transfer mode. Since the DREG is selected by this $\overline{DACK}$ signal in the DMA mode instead of the address input from A3–A0, data transfer in between DREG of SPC and external memory is possible.
$\overline{SD0}$ $\overline{SD1}$ $\overline{SD2}$ $\overline{SD3}$ $\overline{SD4}$ $\overline{SD5}$ $\overline{SD6}$ $\overline{SD7}$ $\overline{SDP}$	17 15 14 13 11 10 8 7 18	23 21 20 18 17 16 14 13 24	I/O	Active low bi–directional SCSI data bus. MSB : $\overline{SD7}$, LSB : $\overline{SD0}$ Odd parity bit : $\overline{SDP}$ Parity check for the SCSI data bus is programmable.
$\overline{SEL}$	34	40	I/O	A signal to issue or detect selection or reselection phase. In selection phase, an initiater asserts this signal, and in reselection phase the signal is asserted by the target.
$\overline{BSY}$	32	38	I/O	This signal indicates the SCSI bus use condition.This signal goes "L" when SPC is in arbitration phase or working as a target. Also, this signal is used to detect bus free phase with $\overline{SEL}$ signal.

PIN DESCRIPTIONS (Continued)

<table>
<tr><th rowspan="2">Designator</th><th colspan="2">Pin No.</th><th rowspan="2">I/O</th><th rowspan="2">Function</th></tr>
<tr><th>DIP</th><th>FPT</th></tr>
<tr><td>$\overline{I/O}$
$\overline{C/D}$
$\overline{MSG}$</td><td>38</td><td>42</td><td>I/O</td><td>Signals to indicate actual phase of information transfer phase as follows:
<table>
<tr><th>MSG</th><th>C/D</th><th>I/O</th><th>Phase Name</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Data Out Phase</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Data In Phase</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Command Phase</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Status Phase</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Reserved</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Reserved</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Message Out Phase</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Message In Phase</td></tr>
</table>
These signals are output from the target and initiator receives them always.</td></tr>
<tr><td>$\overline{REQ}$</td><td>39</td><td>45</td><td>I/O</td><td>In the data transfer phase, the REQ signal is used to notify the INITIATOR that the TARGET is ready to receive or send data. The REQ input is used as a timing control signal in the data transfer sequence.</td></tr>
<tr><td>$\overline{ACK}$</td><td>41</td><td>47</td><td>I/O</td><td>In the data transfer phase, the acknowledge signal is in response to a transfer request (REQ) signal from the TARGET. In the same way as REQ, an ACK input is used as a timing signal in the data transfer sequence.</td></tr>
<tr><td>$\overline{ATN}$</td><td>42</td><td>48</td><td>I/O</td><td>A signal to indicate attention condition. This signal is only output from an initiator.</td></tr>
<tr><td>$\overline{SRST}$</td><td>31</td><td>37</td><td>I/O</td><td>SCSI reset signal to be enabled by register setting. SCSI input from other SCSI devices is non-maskable.</td></tr>
<tr><td>NC
(RNC)</td><td>12, 37</td><td>19, 43</td><td>—</td><td>No connectors</td></tr>
</table>

ADDRESSING OF INTERNAL REGISTERS

SPC has internal registers consisting of 15 bytes that are accessible from an external circuit. These internal registers are used for controlling SPC internal operation and indicating SPC processing status/result status. A unique address is assigned to each internal register, and a particular register is identified by address bits A3 to A0. The following table shows internal register addressing:

Internal Register Addressing

$\overline{CS}$	A3	A2	A1	A0	Operation	Register Name	Abbr
0	0	0	0	0	Read/Write	Bus Device ID	BDID
0	0	0	0	1	Read/Write	SPC Control	SCTL
0	0	0	1	0	Read/Write	Command	SCMD
0	0	0	1	1	—	Open	—
0	0	1	0	0	Read	Interrupt Sense	INTS
					Write	Reset Interrupt	
0	0	1	0	1	Read	Phase Sense	PSNS
					Write	SPC Diagnostic Control	SDGC
0	0	1	1	0	Read	SPC Status	SSTS
					Write	—	
0	0	1	1	1	Read	SPC Error Status	SERR
					Write	—	
0	1	0	0	0	Read/Write	Phase Control	PCTL
0	1	0	0	1	Read	Modified Byte Count	MBC
					Write	—	
0	1	0	1	0	Read/Write	Data Register	DREG
0	1	0	1	1	Read/Write	Temporary Register	TEMP
0	1	1	0	0	Read/Write	Transfer Counter (High)	TCH
0	1	1	0	1	Read/Write	Transfer Counter (Middle)	TCM
0	1	1	1	0	Read/Write	Transfer Counter (Low)	TCL

BIT ASSIGNMENTS

The following table shows the bit assignments to each internal register. When accessing an internal register (in read/write), remember the following:

1. The internal register block includes the read–only/write–only register and those having different meanings in read and write operations.
2. A write command to a read–only register is ignored.
3. If the write–only register is read out, the data and parity bit are undefined.
4. At bit positions indicating "_" for a write in either 1 or 0, may be written.

Bit Assignments for Internal Registers

HEX Address	Register and Mnemonic	R/W Operation	7 (MSB)	6	5	4	3	2	1	0 (LSB)	Parity
0	Bus Device ID (BDID)	R	#7	#6	#5	#4	#3	#2	#1	#0	0
		W	— —					ID4	ID2	ID1	—
1	SPC Control (SCTL)	R/W	Reset & Disable	Control Reset	Diag Mode	ARBIT Enable	Parity Enable	Select Enable	Reselect Enable	INT Enable	P
2	Command (SCMD)	R	Command Code			RST Out	Intercept Xfer	Transfer PRG Xfer	0	Modifer Term Mode	P
		W									
3		R									
		W									
4	Interrupt Sense (SERR)	R	Selected	Reselected	Disconnect	Command Complete	Service Required	Time Out	SPC Hard Error	Reset Condition	P
		W	Reset Interrupt								—
5	Phase Sense (PSNS)	R	REQ	ACK	ATN	SEL	BSY	MSG	C/D	I/O	P
	SPC Diag. Control (SDGC)	W	Diag. REQ	Diag. ACK	Xfer Enable		Diag. BSY	Diag. MSG	Diag. C/D	Diag. I/O	—
6	SPC Status (SSTS)	R	Connected INIT	Connected TARG	SPC BSY	XFER In Progress	SCSI RST	TC=0	DREG Status Full	DREG Status Empty	P
		W	—								—

Bit Assignments For Internal Registers (Continued)

HEX Address	Register and Mnemonic	R/W Operation	7 (MSB)	6	5	4	3	2	1	0 (LSB)	Parity
7	SPC Error Status (SERR)	R	Data Error SCSI	Data Error SPC	Xfer Out	0	TC Parity error	0	Short Period	0	P
8	Phase Control (PCTL)	R/W	Bus Free Interrupt Enable	0				Transfer Phase MSG Out	Transfer Phase C/D Out	Transfer Phase I/O Out	P
9	Modified Byte Counter (MBC)	R	0				MBC Bit3	MBC Bit2	MBC Bit1	MBC Bit0	P
A	Data Register (DREG)	R/W	Internal Data Register (8 Byte FIFO) Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	P
B	Temporary Register (TEMP)	R	Temporary Data (Input: From SCSI) Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	P
		W	Temporary Data (Output: to SCSI) Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	P
C	Transfer Counter High (TCH)	R/W	Transfer Counter High (MSB) Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16	P
D	Transfer Counter Mid. (TCM)	R/W	Transfer Counter High (2nd Byte) Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	p
E	Transfer Counter Low (TCL)	R/W	Transfer Counter High (LSB) Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	P
F	External Buffer (EXBF)	R	External Buffer								
		W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	P

These bit assignments for internal registers are identical to those in the MB87030, MB87031, and MB89351. Therfore, SPC replacement from them to this MB89352 is very easy and it does not require any new software design.

DC CHARACTERISTICS (T_A=0–70°C, V_{CC}=5V (±5%))

(Recommended operating conditions unless otherwise specified)

SCSI Bus Signal Pins

Parameter	Designator	Conditions	Values			Unit
			Min.	Typ.	Max.	
Input High Voltage	V_{IH}		2.0	—	5.25	V
Input Low Voltage	V_{IL}		0	—	0.8	V
Input High Current	I_{IH}	V_{IH} = 5.25V	—	100	400	mA
Input Low Current	I_{IL}	V_{IL} = OV	—	–100	–400	mA
Output Voltage	V_{OL}	V_{CC} = 4.75V I_{OL} = 48 mA	—	—	0.5	V
Input Hysteresis Width	V_{HM}	—	0.2	0.4	—	V

DC CHARACTERISTICS (Continued)

MPU Bus Signal Pins

Parameter	Designator	Conditions	Values			Unit
			Min.	Typ.	Max.	
Input High Voltage	V_{IH}		2.2	—	V_{CC} + 0.3	V
Input Low Voltage	V_{IL}		V_{ss} – 0.3	—	0.8	V
Output High Voltage	V_{OH}	I_{OH} = – 0.4 mA	4.0		V_{CC}	V
Output Low Voltage	V_{OL}	I_{OL} = + 3.2 mA	V_{ss}		0.4	V
Input Leakage Current	I_{LIH}	V_{IH} = 5.25			20	mA
	I_{LIH}	V_{IL} = 0.0			–10	mA
Input/Output Leakage Current	I_{LZH}	V_{IH} = 5.25			40	mA
	I_{LZH}	V_{IL} = 0.0			–40	mA
Power Supply Current	I_{CC}	Input Clock = 8 MHz All Output Pins Open			10	

AC CHARACTERISTICS

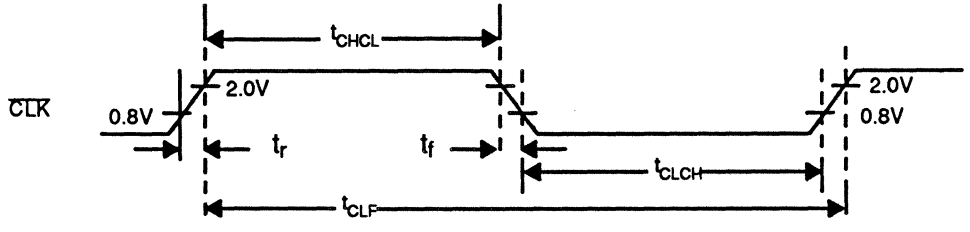
(Recommended operating conditions unless otherwise noted)

Clock Signal

$\overline{CLK}$ Input

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
$\overline{CLK}$ Cycle Time	t_{CLF}	125		200	ns
$\overline{CLK}$ High Time	t_{CHCL}	44			ns
$\overline{CLK}$ Pulse Width	t_{CLCH}	44			ns
$\overline{CLK}$ Rising Skew Time	t_r			10	ns
$\overline{CLK}$ Falling Skew Time	t_f			10	ns

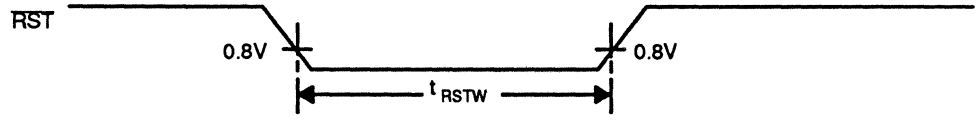
Clock Signal Timing



AC CHARACTERISTICS (Continued)

RST Input					
Parameter	**Designator**	**Values**			**Unit**
		Min.	**Typ.**	**Max.**	
RST Pulse Width	t_{RSTW}	100			ns

Reset Input Timing

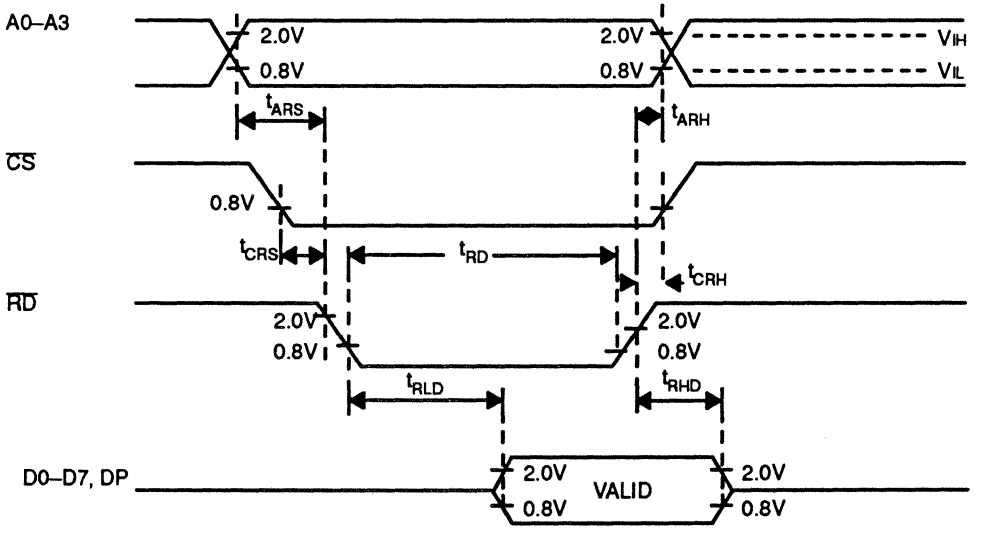


AC CHARACTERISTICS (Continued)

Register Read

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
Address Setup Time	t_{ARS}	40			ns
Address Hold Time	t_{ARH}	10			ns
$\overline{CS}$ Setup Time	t_{CRS}	25			ns
$\overline{CS}$ Hold Time	t_{CRH}	10			ns
Data Valid Time (from $\overline{RD}$ Low) (C_L = 80pF)	t_{RLD}			90	ns
Data Valid Time (from $\overline{RD}$ Hi gh) (C_L = 20pF)	t_{RHD}	10		60	ns
$\overline{RD}$ Pulse Width	t_{RD}	120			ns

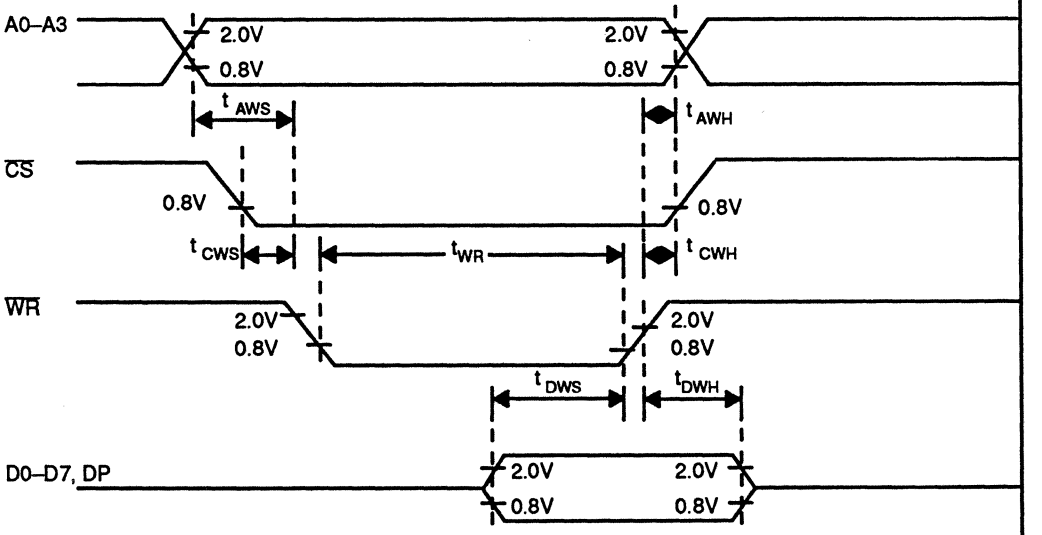
Register Read Timing



AC CHARACTERISTICS (Continued)

Register Write					
Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
Address Setup Time	t_{AWS}	40			ns
Address Hold Line	t_{AWH}	10			ns
$\overline{CS}$ Setup Time	t_{CWS}	25			ns
$\overline{CS}$ Hold Time	t_{CWH}	10			ns
Data Bus Setup Time	t_{DWS}	30			ns
Data Bus Hold Time	t_{DWH}	20			ns
WR Pulse Width	t_{WR}	100			ns

Register Write Timing

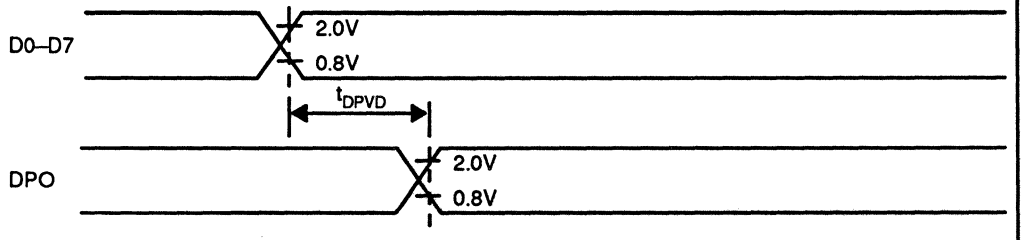


AC CHARACTERISTICS (Continued)

DPO (Data Parity Output)

Parameter	Designator	Test Conditions	Values			Unit
			Min.	Typ.	Max.	
Data Bus (D0 – D7) Valid to DPO Valid	t_{DPVD}	CL = 30pF			60	ns

DPO Timing



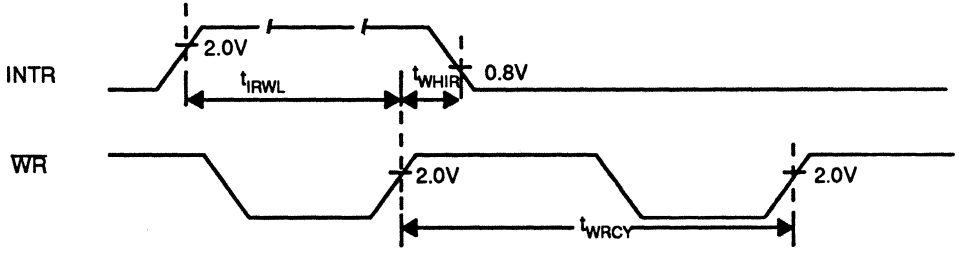
AC CHARACTERISTICS (Continued)

INTR (Interrupt Request) Output

Parameter	Designator	Test Conditions	Values			Unit
			Min.	Typ.	Max.	
$\overline{WR}$ High to INTR Low (Interrupt reset)	t_{WHIR}	CL = 10pf	t_{CLF}		$2t_{CLF} + 100$	ns
INTR High to $\overline{WR}$ High	t_{IRWL}		0			ns
INTR Reset Cycle Time	t_{WRCY}		$4t_{CLF}$			ns

NOTE: Applicable only when interrupt reset is executed.

Interrupt Output Timing



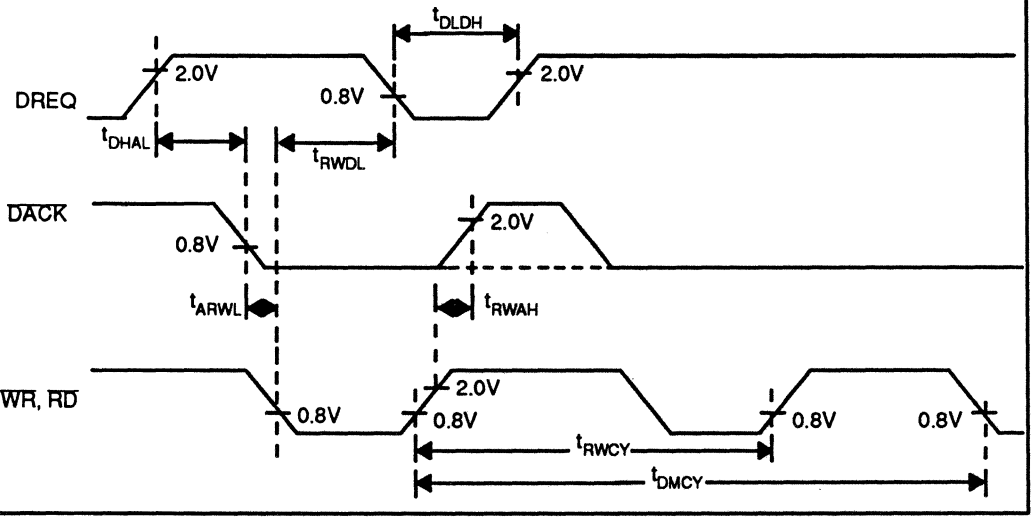
AC CHARACTERISTICS (Continued)

DMA Access

Parameter	Designator	Test Conditions	Values			Unit
			Min.	Typ.	Max.	
DREQ High to $\overline{DACK}$ Low	t_{DHAL}		0			ns
$\overline{WR}$ and $\overline{RD}$ Service Time (from $\overline{DACK}$ Low to $\overline{WR}$ or $\overline{RD}$ Low)	t_{ARWL}		40			ns
DREQ Release Time (from $\overline{WR}$ or $\overline{RD}$ Low to DREQ Low) (Note)	t_{RWDL}	CL = 30 pF	35		150	ns
DACK Hold Time (from $\overline{WR}$ or $\overline{RD}$ High to $\overline{DACK}$ Low)	t_{RWAH}		10			ns
DREQ Interval (from DREQ Low to DREQ High)	t_{DLDH}		0			ns
DREG Access Cycle Time (1)	t_{RWCY}		$2t_{CLF}$			ns
DREG Access Cycle Time (2)	t_{DMCY}		$3t_{CLF}$			ns

NOTE: The $\overline{WR}$ parameter is applicable when data buffer register will be full; the $\overline{RD}$ parameter is applicable when the data buffer register will be empty.

DMA Access Timing

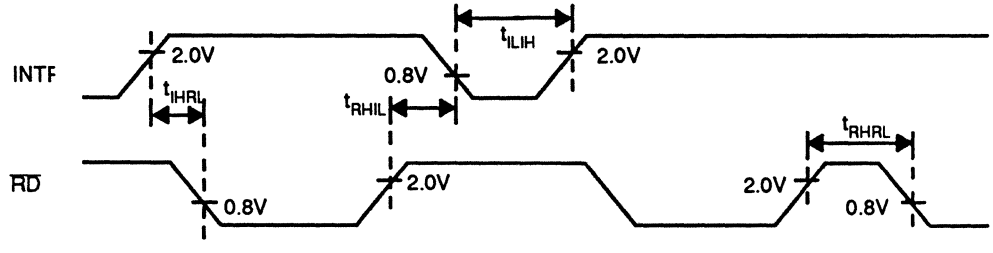


AC CHARACTERISTICS (Continued)

DREG Access – Program Transfer with INTR (Input Operation)

Parameter	Designator	Test Conditions	Values			Unit
			Min.	Typ.	Max.	
$\overline{RD}$ Service Time (from INTR High to $\overline{RD}$ Low)	t_{IHRL}		0			ns
INTR Release Time (from $\overline{RD}$ High to INTR Low) (Note)	t_{RHIL}	CL = 20 pF	35		150	ns
INTR Recovery Time (from INTR Low to INTR High)	t_{ILIH}		0			ns
$\overline{RD}$ Recovery Time (from $\overline{RD}$ High to $\overline{RD}$ Low)	t_{RHRL}		50			ns

DREG Access Timing (Input Operation)



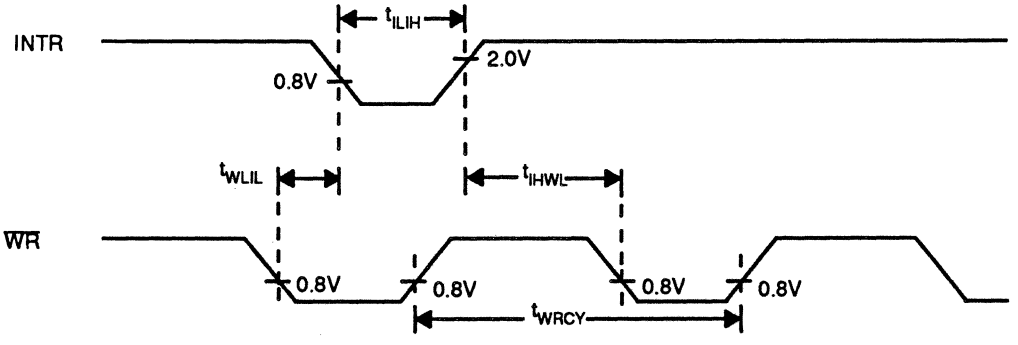
NOTE: This parameter is applicable when the data buffer register will be full in the output operation and empty in the input operation.

AC CHARACTERISTICS (Continued)

DREG Access – Program Transfer with INTR (Output Operation)

Parameter	Designator	Test Conditions	Values			Unit
			Min.	Typ.	Max.	
$\overline{WR}$ Service Time (from INTR High to $\overline{WR}$ Low)	t_{IHWL}		0			ns
INTR Release Time (from $\overline{WR}$ High to INTR Low) (Note)	t_{WLIL}	CL = 20 pF	35		150	ns
INTR Recovery Time (from INTR Low to INTR High)	t_{ILIH}		0			ns
$\overline{WR}$ Cycle Time	t_{WRCY}		$2t_{CLF}$			ns

DREG Access Timing (Output Operations)



NOTE: This parameter is applicable when the data buffer register will be full in the output operation and empty in the input operation.

AC CHARACTERISTICS (Continued)

SCSI Bus Interface Selection Phase Timing

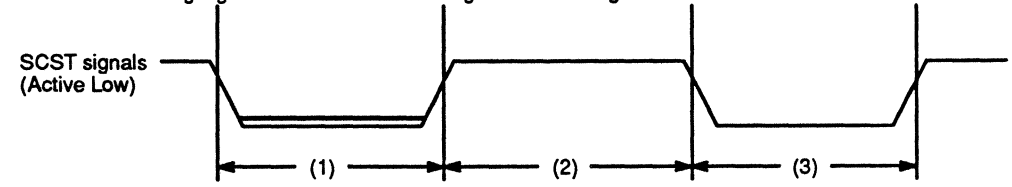
INITIATOR — SELECTION WITH ARBITRATION

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
Bus Free Time*2		t_{BFR}		$4t_{CLF}+50$	ns
Start of Arbitration	t_{BFBL}	$(6+n)$*1 x t_{CLF}		$(7+n) \times t_{CLF}+60$	
$\overline{BSY}$ Low to Self ID# Output	t_{BLID}	0		60	
$\overline{BSY}$ Low to Prioritize	t_{ARB}	$32t_{CLF}-60$			
Data Bus Valid to Prioritize	t_{AIDV}	200			
Bus Usage Permission Granted to $\overline{SEL}$ Low	t_{AWSL}	0		80	
$\overline{SEL}$ Low to Data Bus ID Output, $\overline{ATN}$ Low	t_{SIDA}	$11t_{CLF}-30$			
Select ID# Output to $\overline{BSY}$ High	t_{IDBH}	$2t_{CLF}-80$			
$\overline{BSY}$ Low to $\overline{SEL}$ High	t_{BLSH}	$2t_{CLF}$			
$\overline{BSY}$ Low to Select ID# Hold	t_{BIDH}	$2t_{CLF}$			
$\overline{SEL}$ High to INTR High	t_{SHIR}			60	
$\overline{SEL}$ Low to $\overline{BSY}$ High, ID Bit High	t_{SBCR}			$3t_{CLF}+180$	
Prioritize to $\overline{BSY}$ High, ID Bit High	t_{PBCR}			110	

*1 n : TCL register value
* 2 Bus Free Time : The minimum time period until the booked select command will be executed.

All SCSI Timing Signals

Note: All SCSI timing signals are defined according to the following rules:

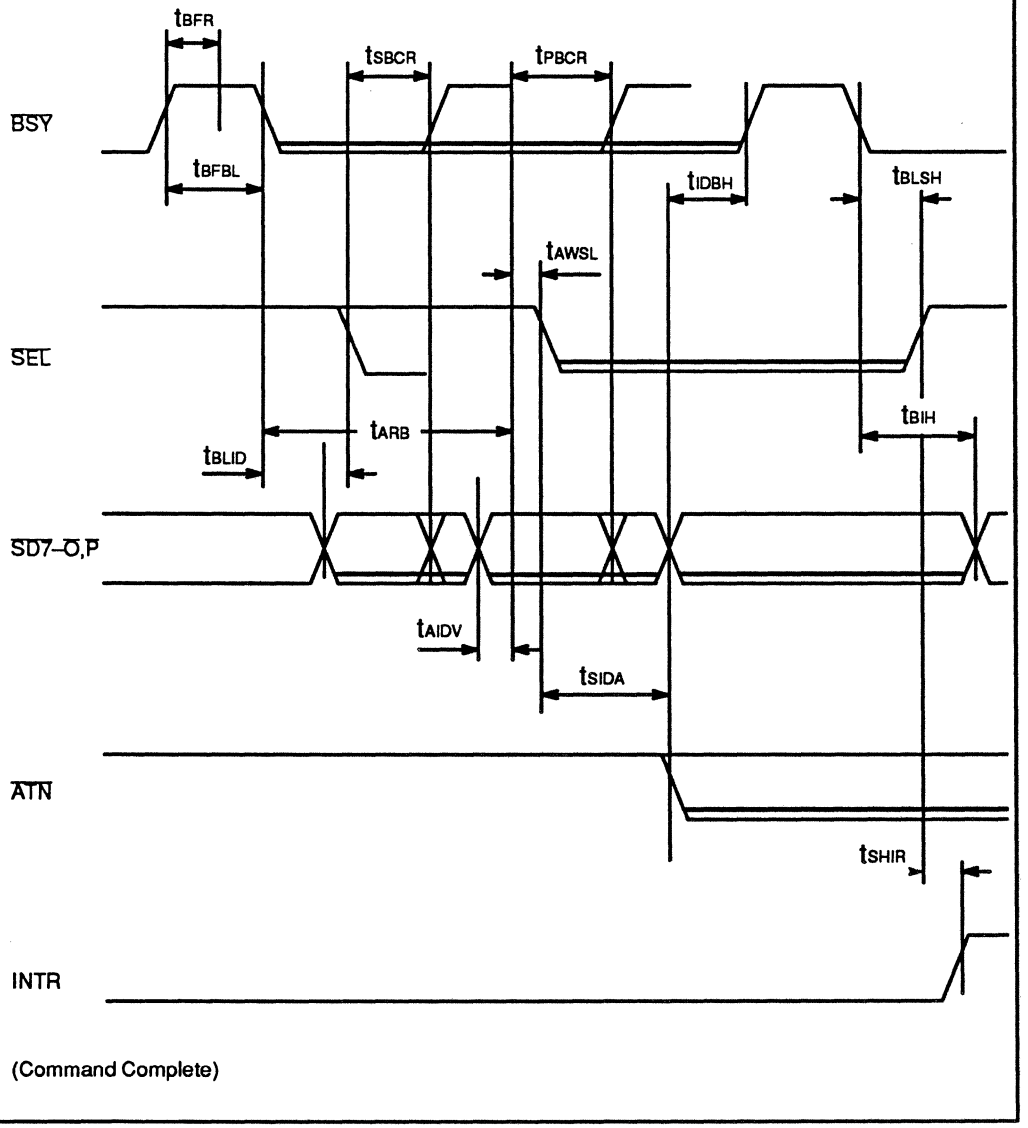


(1) The SPC outputs low level signal to the bus.
(2) All devices hooked up to the bus do not output low level signals.
(3) Other devices hooked up to the bus outputs low level signal.

AC CHARACTERISTICS (Continued)

SCSI Bus Interface Selection Phase Timing

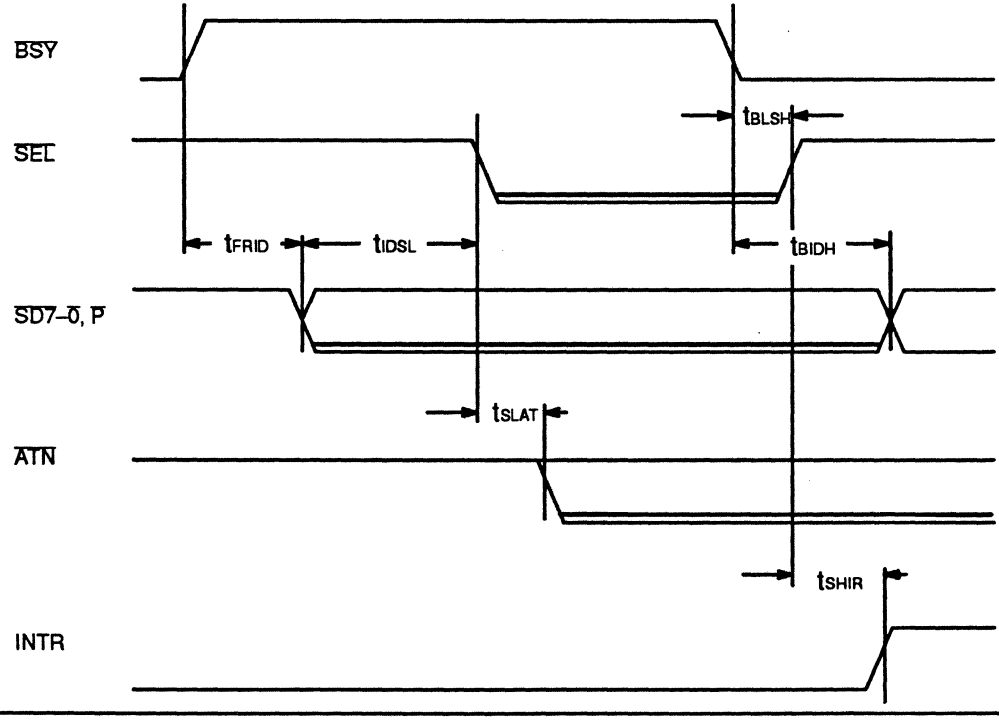
Initiator with Arbitration Timing



AC CHARACTERISTICS (Continued)

TARGET — SELECTION WITHOUT ARBITRATION					
Parameter	**Designator**	**Values**			**Unit**
		Min.	**Typ.**	**Max.**	
$\overline{BSY}$ High to Select ID# Output	t_{FRID}	(6+n) x t_{CLF}		(7+n) x t_{CLF}+140	ns
ID# Output to $\overline{SEL}$ Low	t_{IDSL}	11t_{CLF}–80			
$\overline{SEL}$ Low to $\overline{ATN}$ Low	t_{SLAT}	11t_{CLF}–80			
$\overline{BSY}$ Low to $\overline{SEL}$ High	t_{BLSH}	2t_{CLF}			
$\overline{BSY}$ Low to ID# Hold	t_{BIDH}	2t_{CLF}			
$\overline{SEL}$ High to INTR High	t_{SHIR}			60	

Initiator without Arbitration Timing

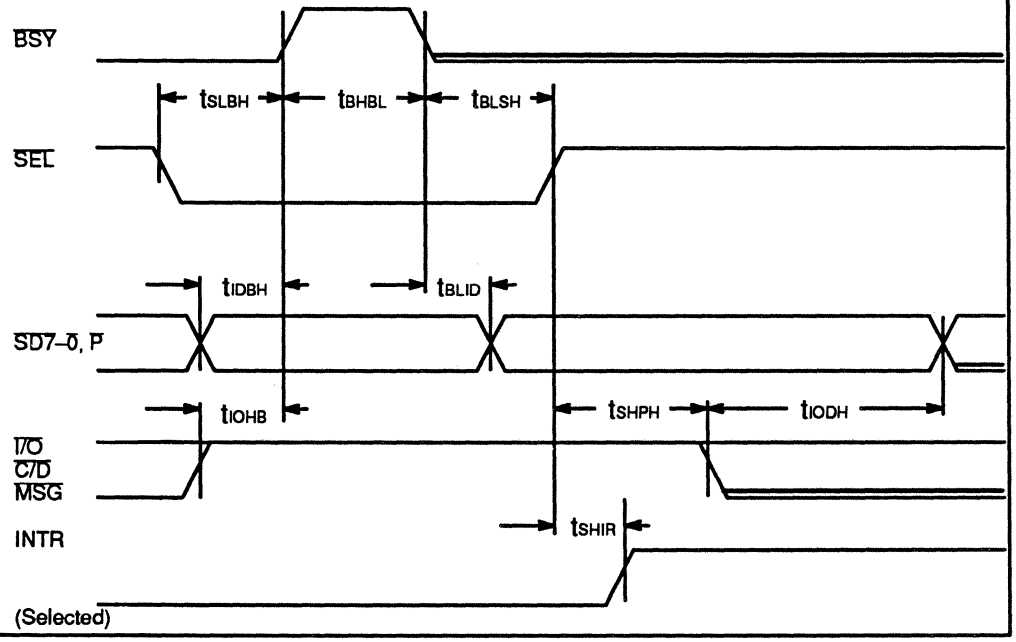


AC CHARACTERISTICS (Continued)

TARGET — SELECTION WITH ARBITRATION

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
SEL Low to BSY High	t_{SLBH}	0			ns
Data Bus Valid (ID#) to BSY High	t_{IDBH}	0			
I/O High to BSY High	t_{IOHB}	0			
BSY High to BSY Low	t_{BHBL}	$4t_{CLF}$		$5t_{CLF}+140$	
BSY Low to ID# Hold	t_{BLID}	60			
BSY Low to SEL High	t_{BLSH}	0			
SEL High to Phase Signal Output	t_{SHPH}	$3t_{CLF}$		$4t_{CLF}+160$	
I/O Low to Data Bus Output	t_{IODE}	$7t_{CLF}$			
SEL High to INTR High	t_{SHIR}			$3t_{CLF}+130$	

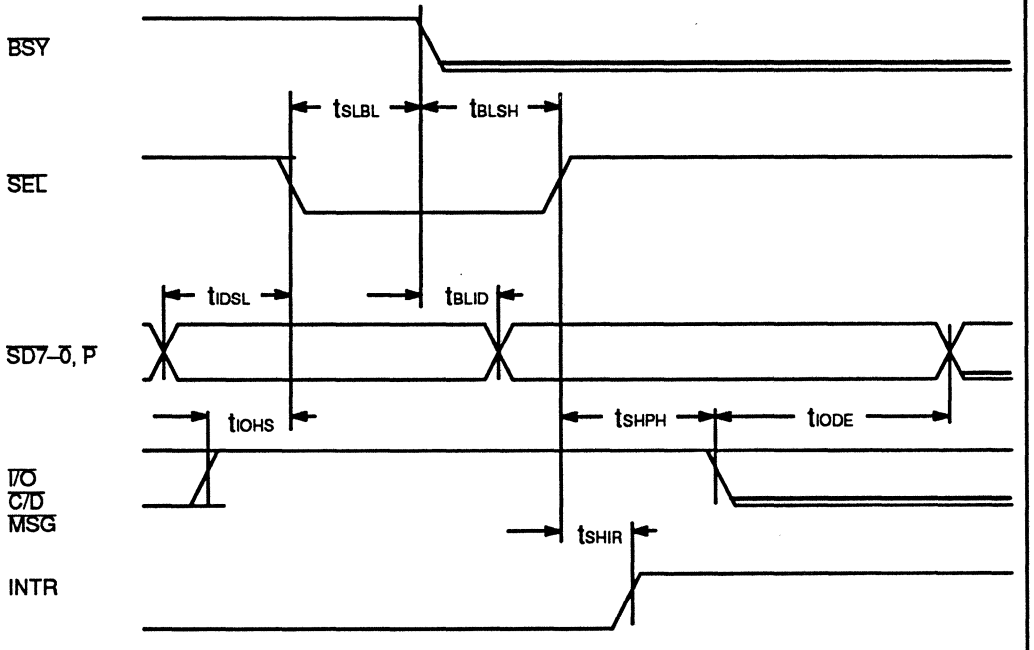
Target with Arbitration Timing



AC CHARACTERISTICS (Continued)

TARGET — SELECTION WITHOUT ARBITRATION					
Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
Data Bus Valid (ID#) to $\overline{SEL}$ Low	t_{IDSL}	0			ns
$\overline{I/O}$ High to $\overline{SEL}$ Low	t_{IOHS}	0			
$\overline{SEL}$ Low to $\overline{BSY}$ Low	t_{SLBL}	$2t_{CLF}$		$3t_{CLF}+130$	
$\overline{BSY}$ Low to ID# Hold	t_{BLID}	60			
$\overline{BSY}$ Low to $\overline{SEL}$ High	t_{BLSH}	0			
$\overline{SEL}$ High to Phase Signal Output	t_{SHPH}	$3t_{CLF}$		$4t_{CLF}+160$	
I/O Low to Data Bus Output	t_{IODE}	$7t_{CLF}$			
$\overline{SEL}$ High to INTR High	t_{SHIR}			$3t_{CLF}+130$	

Target without Arbitration Timing



(Selected)

AC CHARACTERISTICS (Continued)

SCSI BUS INTERFACE – RESELECTION PHASE TIMING

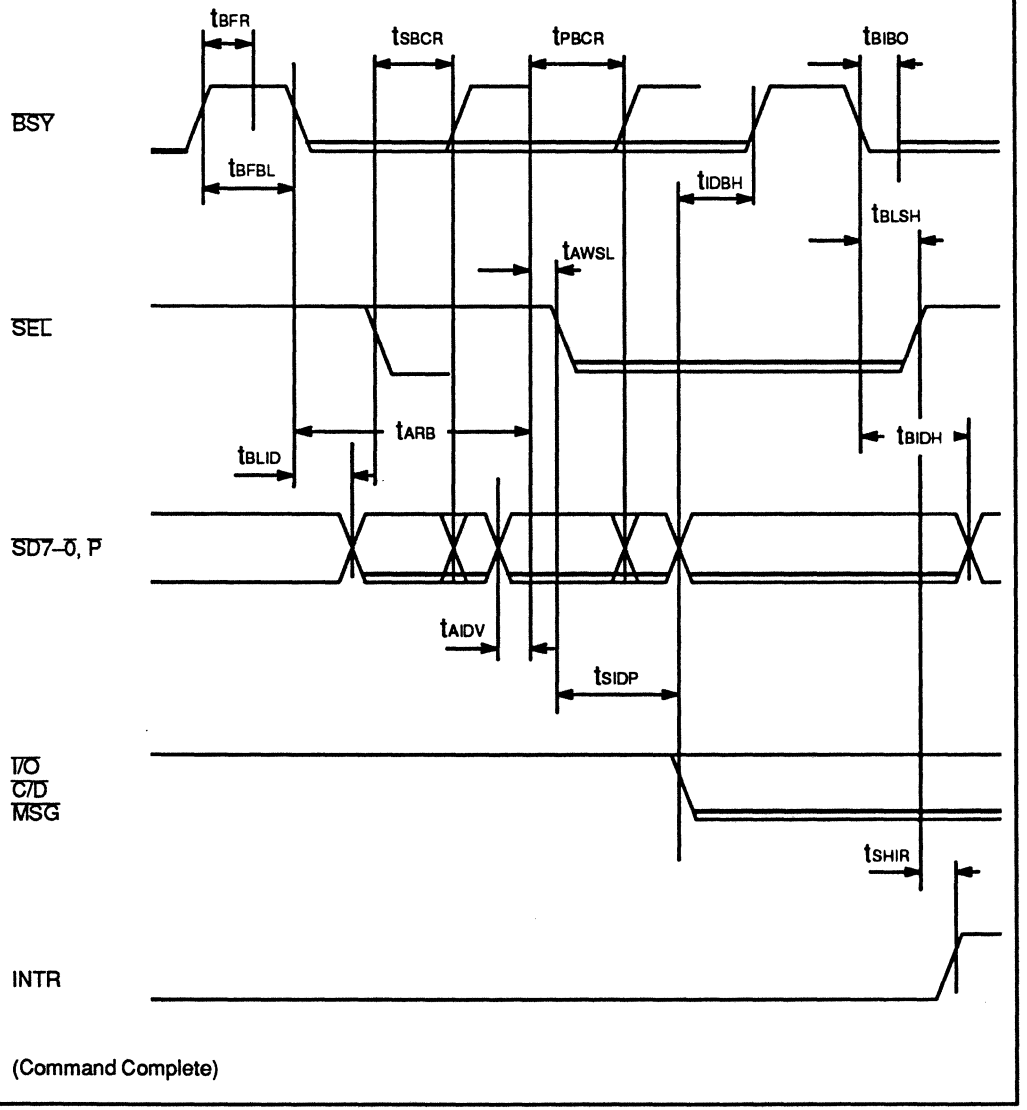
TARGET — RESELECTION PHASE TIMING

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
Bus Free Time*2	t_{BFR}	$4t_{CLF}+50$			ns
Start of Arbitration	t_{BFBL}	$(6+n^{*1}) \times t_{CLF}$		$(7+n) \times t_{CLF}+140$	
$\overline{BSY}$ Low to Self ID# Output	t_{BLID}	0		60	
$\overline{BSY}$ Low to Prioritize	t_{ARB}	$32t_{CLF}-60$			
Data Bus Valid to Prioritize	t_{AIDV}	200			
Bus Usage Permission Granted to $\overline{SEL}$ Low	t_{AWSL}	0		80	
$\overline{SEL}$ Low to Data Bus ID Output, Phase Signal Output	t_{SIDP}	$11t_{CLF}-50$			
Select ID# Output to $\overline{BSY}$ High	t_{IDBH}	$2t_{CLF}-80$			
$\overline{BSY}$ Low to $\overline{BSY}$ Low Output	t_{BIBO}	$2t_{CLF}+20$		$3t_{CLF}+140$	
$\overline{BSY}$ Low to $\overline{SEL}$ High	t_{BLSH}	$2t_{CLF}$			
$\overline{BSY}$ Low to Select ID# Hold	t_{BIDH}	$2t_{CLF}$			
$\overline{SEL}$ High to INTR High	t_{SHIR}			60	
$\overline{SEL}$ Low to $\overline{BSY}$ High, ID Bit High	t_{SBCR}			$3t_{CLF}+180$	
Prioritize to $\overline{BSY}$ High, ID Bit High	t_{PBCR}			110	

*1 n : TCL register value
*2 Bus Free Time : The minimum time period till the booked select command will be executed.

AC CHARACTERISTICS (Continued)

Reselection Timing (Target)

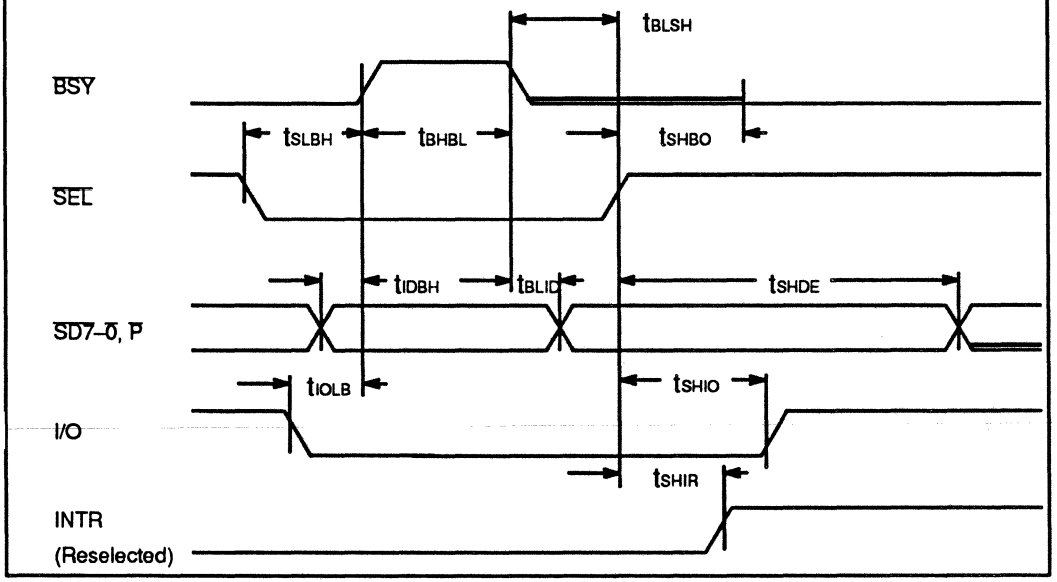


AC CHARACTERISTICS (Continued)

INITIATOR — RESELECTION PHASE TIMING

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
SEL Low to BSY High	t_{SLBH}	0			ns
Data Bus Valid (ID#) to BSY High	t_{IDBH}	0			
I/O Low to BSY High	t_{IOLB}	0			
BSY High to BSY Low	t_{BHBL}	$4t_{CLF}$		$5t_{CLF}+140$	
BSY Low to ID# Hold	t_{BLID}	60			
BSY Low to SEL High	t_{BLSH}	0			
SEL High to BSY Low Output	t_{SHBO}	$2t_{CLF}$		$3t_{CLF}+140$	
SEL High to Data Bus Valid (When I/O is High)	t_{SHDE}	$3t_{CLF}+30$		$4t_{CLF}+160$	
SEL High to I/O High	t_{SHIO}	200			
SEL High to INTR High	t_{SHIR}			$3t_{CLF}+130$	

Reselection Timing (Initiator)



AC CHARACTERISTICS (Continued)

SCSI BUS INTERFACE – INFORMATION TRANSFER PHASE TIMING

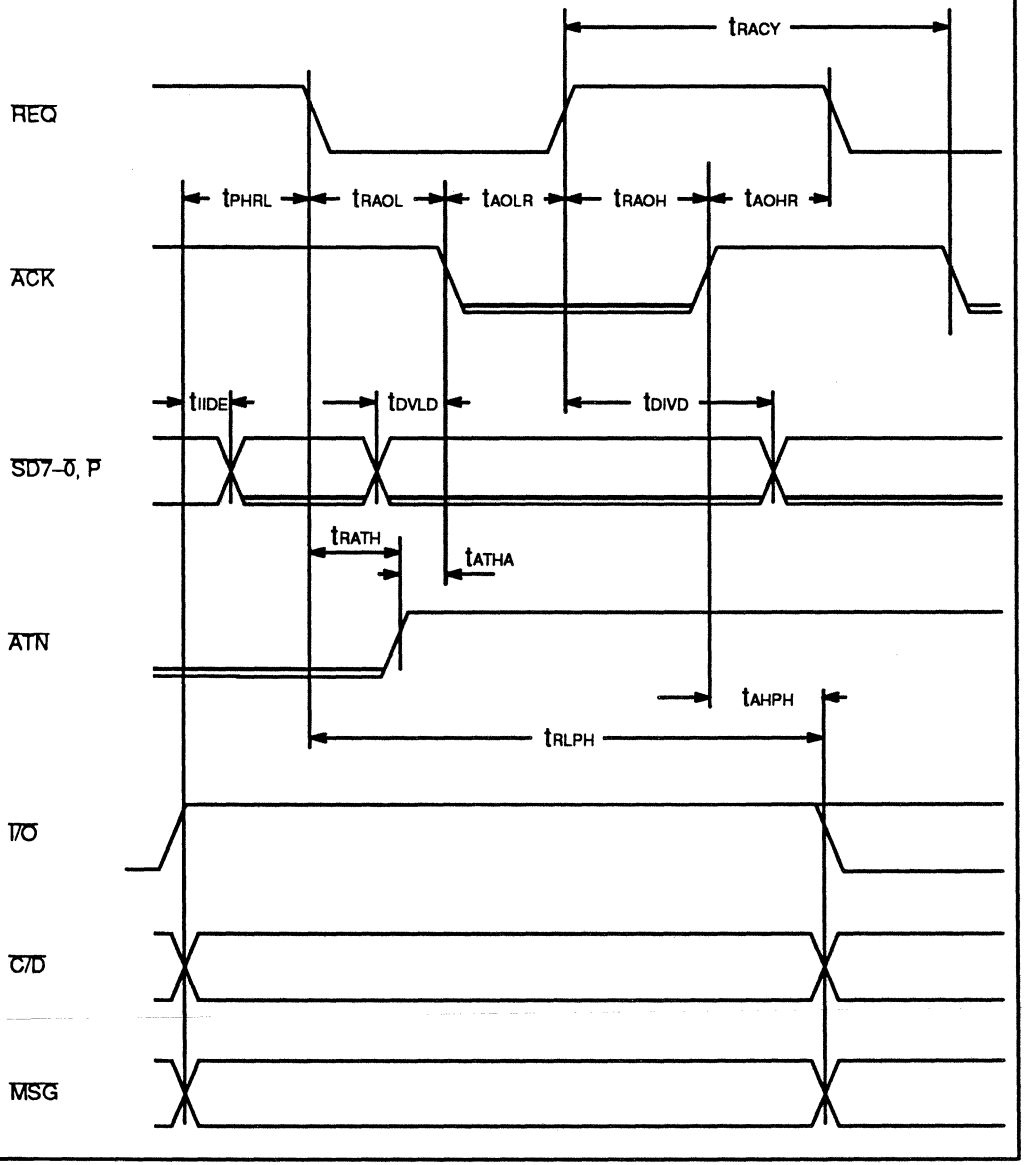
INITIATOR – ASYNCHRONOUS TRANSFER OUTPUT

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
$\overline{I/O}$ High to Data Bus Output	t_{IIDE}	10			ns
Phase Set to $\overline{REQ}$ Low	t_{PHRL}	100			
$\overline{REQ}$ Low to $\overline{ACK}$ Low	t_{RAOL}	20			
Data Bus Valid to $\overline{ACK}$ Low	t_{DVLD}	$2t_{CLF}$–80			
$\overline{ACK}$ Low to $\overline{REQ}$ High	t_{AOLR}	0			
$\overline{REQ}$ High to $\overline{ACK}$ High	t_{RAOH}	10			
$\overline{ACK}$ High to $\overline{REQ}$ Low	t_{AOHR}	0			
$\overline{REQ}$ High to $\overline{ACK}$ Low	t_{RACY}	$2t_{CLF}$			
$\overline{REQ}$ High to Data Bus Hold	t_{DIVD}	15			
$\overline{REQ}$ Low to $\overline{ATN}$ High *1	t_{RATH}	$2t_{CLF}$			
$\overline{ATN}$ High to $\overline{ACK}$ Low *1	t_{ATHA}	t_{CLF}–20			
$\overline{REQ}$ Low to Phase Change *2	t_{RLPH}	$3t_{CLF}$			
$\overline{ACK}$ High to Phase Change *2	t_{AHPH}	10			

*1 : This spec is applicable to the last byte transfer of message out phase in hardware transfer mode.
*2 : When the transfer phase is changed, both t_{RLPH} and t_{AHPH} should be specified.

AC CHARACTERISTICS (Continued)

Asynchronous Transfer Output Timing (Initiator)

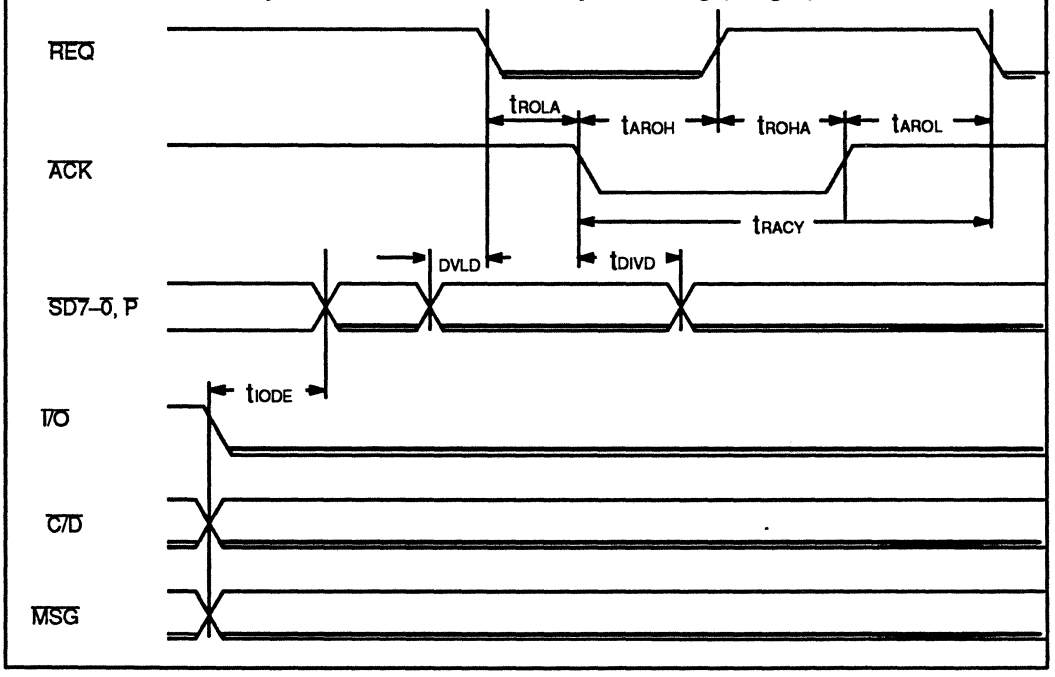


AC CHARACTERISTICS (Continued)

TARGET —ASYNCHRONOUS TRANSFER OUTPUT

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
$\overline{I/O}$ Low to Data Bus Output	t_{IODE}	$7t_{CLF}$			ns
Data Bus Valid to $\overline{REQ}$ Low	t_{DVLD}	$2t_{CLF}-80$			
$\overline{ACK}$ Low to Data Bus Hold	t_{DIVD}	15			
$\overline{REQ}$ Low to $\overline{ACK}$ Low	t_{ROLA}	0			
$\overline{ACK}$ Low to $\overline{REQ}$ High	t_{AROH}	10		180	
$\overline{REQ}$ High to $\overline{ACK}$ High	t_{ROHA}	0			
$\overline{ACK}$ High to $\overline{REQ}$ Low	t_{AROL}	10			
$\overline{ACK}$ Low to $\overline{REQ}$ Low	t_{RACY}	$2t_{CLF}$			

Asynchronous Transfer Output Timing (Target)



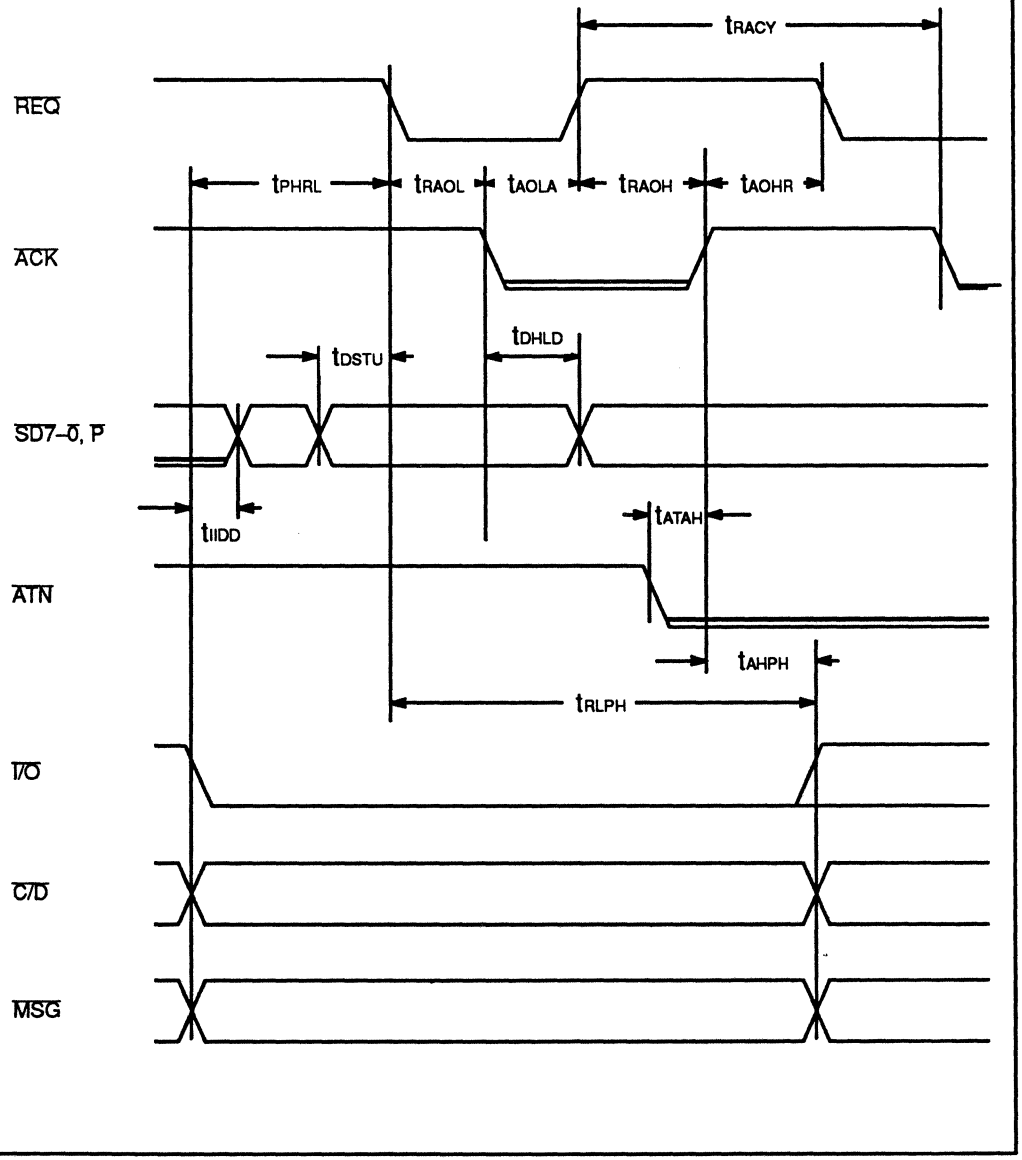
AC CHARACTERISTICS (Continued)

INITIATOR—ASYNCHRONOUS TRANSFER OUTPUT					
Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
$\overline{I/O}$ Low to Data Bus Output Terminate	t_{IIDE}			140	ns
Phase Set to $\overline{REQ}$ Low	t_{PHRL}	100			
Data Bus Valid to $\overline{REQ}$ Low	t_{DSTU}	10			
$\overline{REQ}$ Low to $\overline{ACK}$ Low	t_{RAOL}	20			
$\overline{ACK}$ Low to $\overline{REQ}$ High	t_{AOLR}	0			
$\overline{ACK}$ Low to Data Bus Hold	t_{DHLD}	15			
$\overline{REQ}$ High to $\overline{ACK}$ High	t_{RAOH}	10			
$\overline{ACK}$ High to $\overline{REQ}$ Low	t_{AOHR}	0			
$\overline{REQ}$ High to $\overline{ACK}$ Low	t_{RACY}	$2t_{CLF}$			
$\overline{ATN}$ Low to $\overline{ACK}$ High[1]	t_{ATAH}	t_{CLF}–20			
$\overline{REQ}$ Low to Phase Change[2]	t_{RLPH}	$3t_{CLF}$			
$\overline{ACK}$ High to Phase Change[2]	t_{AHPH}	10			

Notes:
1. Applicable to the last byte transfer of message out phase in hardware transfer mode.
2. When the transfer phase is changed, both t_{RLPH} and t_{AHPH} should be specified.

AC CHARACTERISTICS (Continued)

Asynchronous Transfer Input Timing (Initiator)

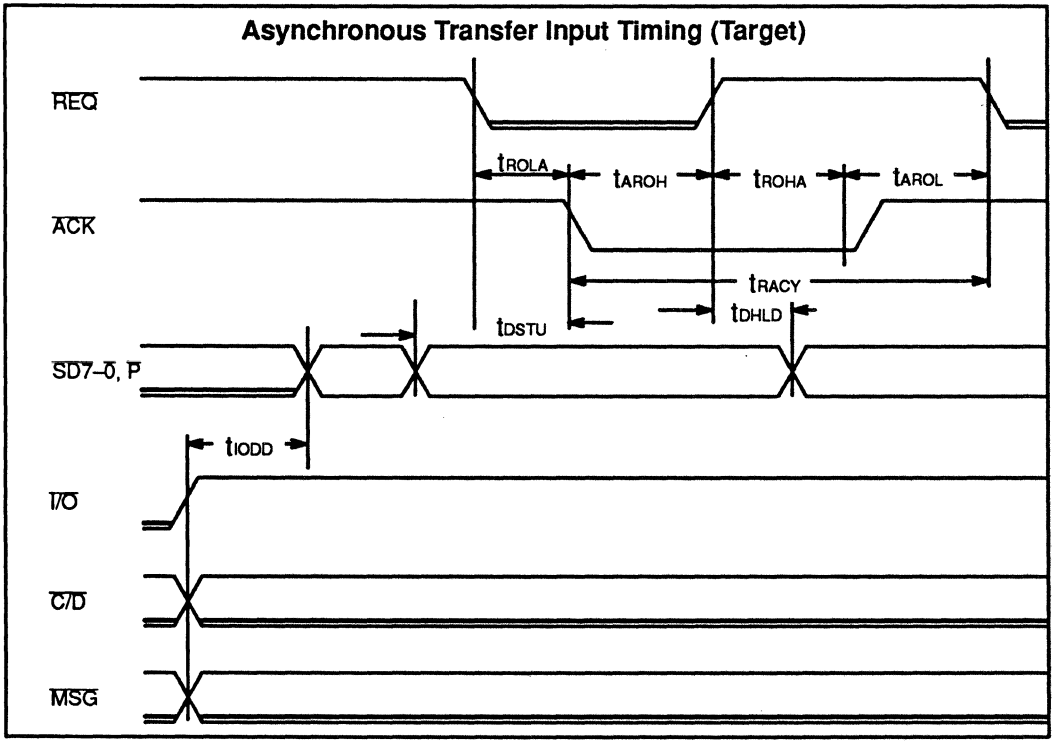


AC CHARACTERISTICS (Continued)

TARGET—ASYNCHRONOUS TRANSFER INPUT

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
$\overline{I/O}$ High to Data Bus Output Terminate	t_{IODD}			30	ns
Data Bus Valid to $\overline{ACK}$ Low	t_{DSTU}	10			
$\overline{REQ}$ High to Data Bus Hold	t_{DHLD}	15			
$\overline{REQ}$ Low to $\overline{ACK}$ Low	t_{ROLA}	0			
$\overline{ACK}$ Low to $\overline{REQ}$ High	t_{AROH}	10		180	
$\overline{REQ}$ High to $\overline{ACK}$ High	t_{ROHA}	0			
$\overline{ACK}$ High to $\overline{REQ}$ Low	t_{AROL}	10			
$\overline{ACK}$ Low to $\overline{REQ}$ Low	t_{RACY}	$2t_{CLF}$			

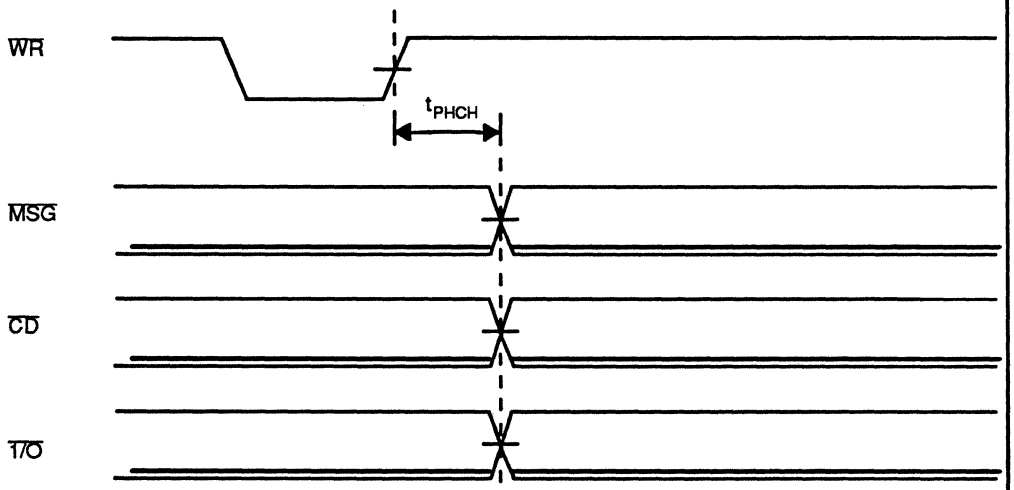
Asynchronous Transfer Input Timing (Target)



AC CHARACTERISTICS (Continued)

Transfer Phase Change (Target)					
Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
From $\overline{WR}$ High to $\overline{MSG}$, $\overline{C/D}$, $\overline{I/O}$ change	t_{PHCH}	10		130	ns

Transfer Phase Change Timing (Target)



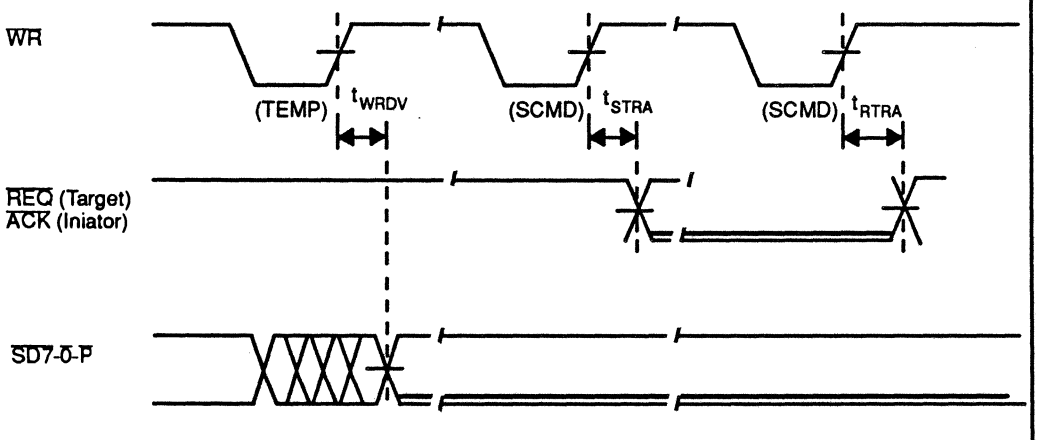
AC CHARACTERISTICS (Continued)

MANUAL TRANSFER (Note)

Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
From $\overline{WR}$ High to Data Bus Valid for TEMP Register	t_{WRDV}			130	ns
From $\overline{WR}$ High to REQ Low, ACK Low for SET ACK/REQ Command	t_{STRA}	$2t_{CLF}$		$3t_{CLF} + 90$	ns
From $\overline{WR}$ High to REQ High, ACK High for RESET ACK/REQ Command	t_{RTRA}	$2t_{CLF}$		$3t_{CLF} + 90$	ns

NOTE: Timing relationships not shown are the same as those for asynchronous transfers.

Manual Transfer Timing

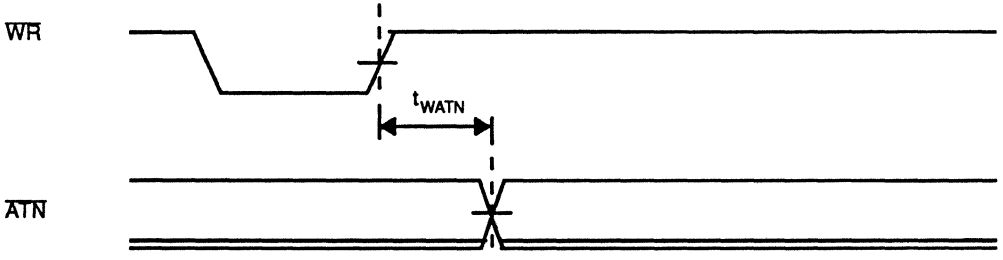


AC CHARACTERISTICS (Continued)

SCSI BUS INTERFACE – ATTENTION CONDITION

INITIATOR - ATTENTION CONDITION					
Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
From $\overline{WR}$ High to $\overline{ATN}$ Change (SET/RESET ATN Command)	t_{WATN}	$2t_{CLF}$		$3t_{CLF} + 90$	ns

Attention Condition Timing (Initiator)

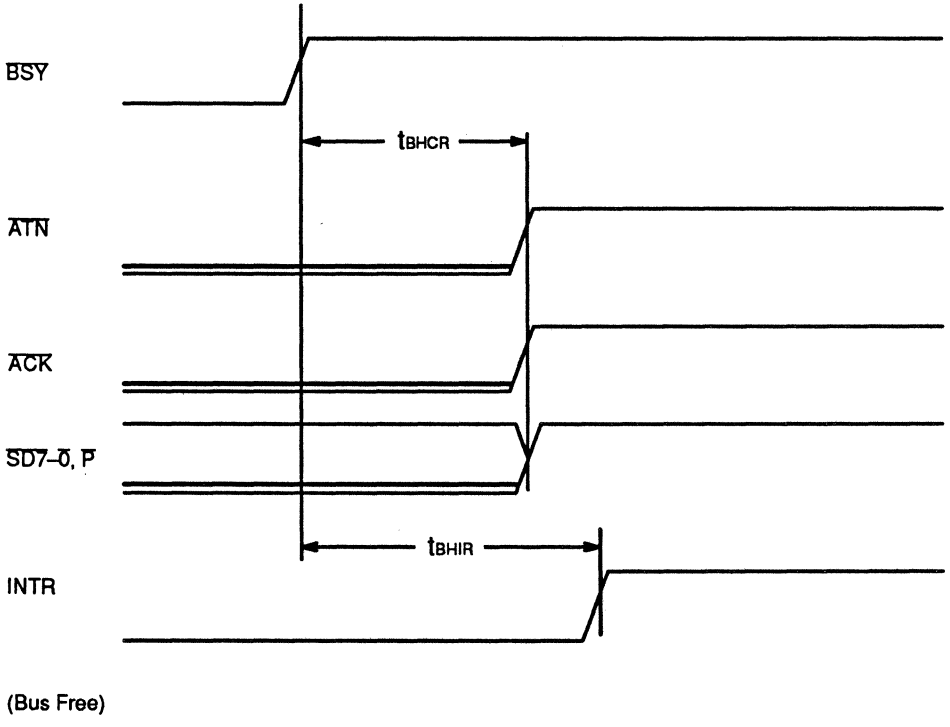


AC CHARACTERISTICS (Continued)

SCSI BUS INTERFACE – BUS FREE

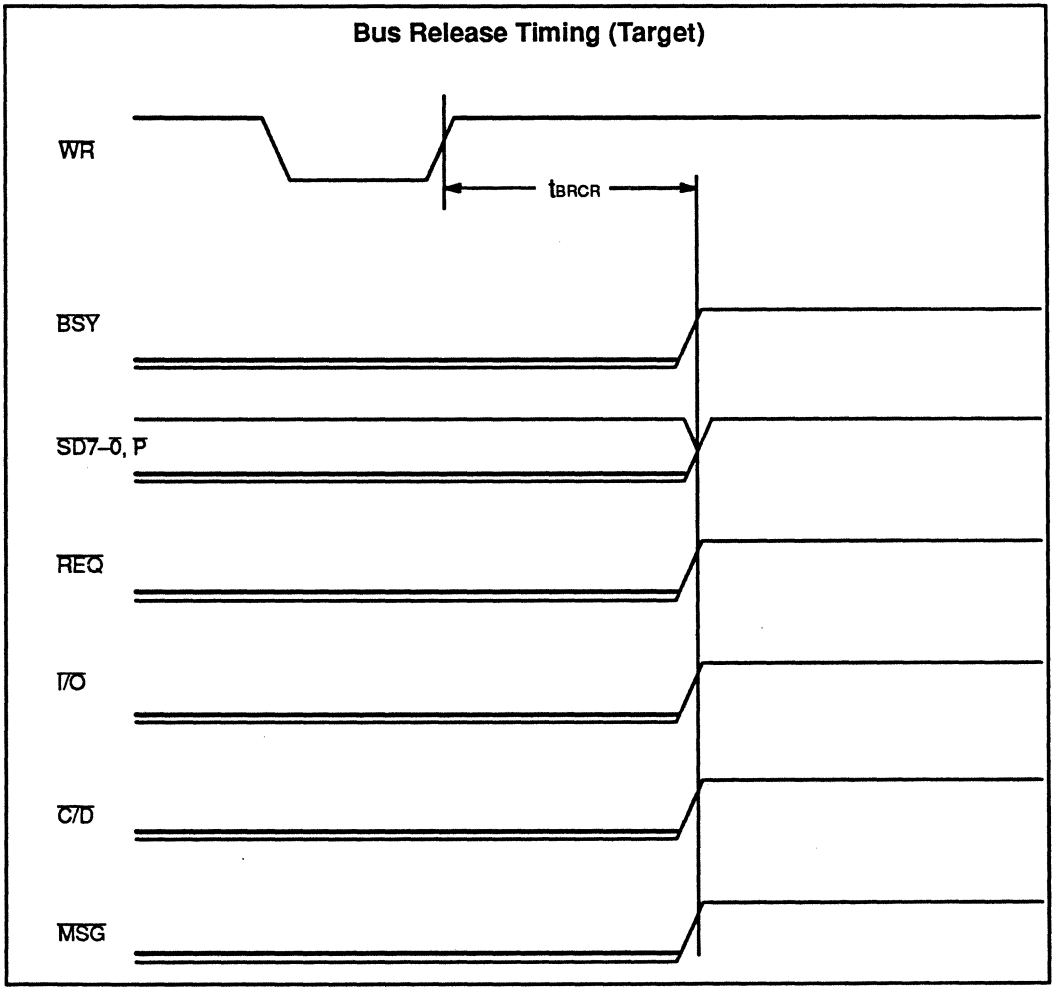
INITIATOR — BUS FREE (DISCONNECTION)					
Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
$\overline{BSY}$ High to Bus Clear	t_{BHCR}			$5t_{CLF}+140$	ns
$\overline{BSY}$ High to INTR High	t_{BHIR}			$6t_{CLF}+80$	ns

Initiator Disconnected Timing



AC CHARACTERISTICS (Continued)

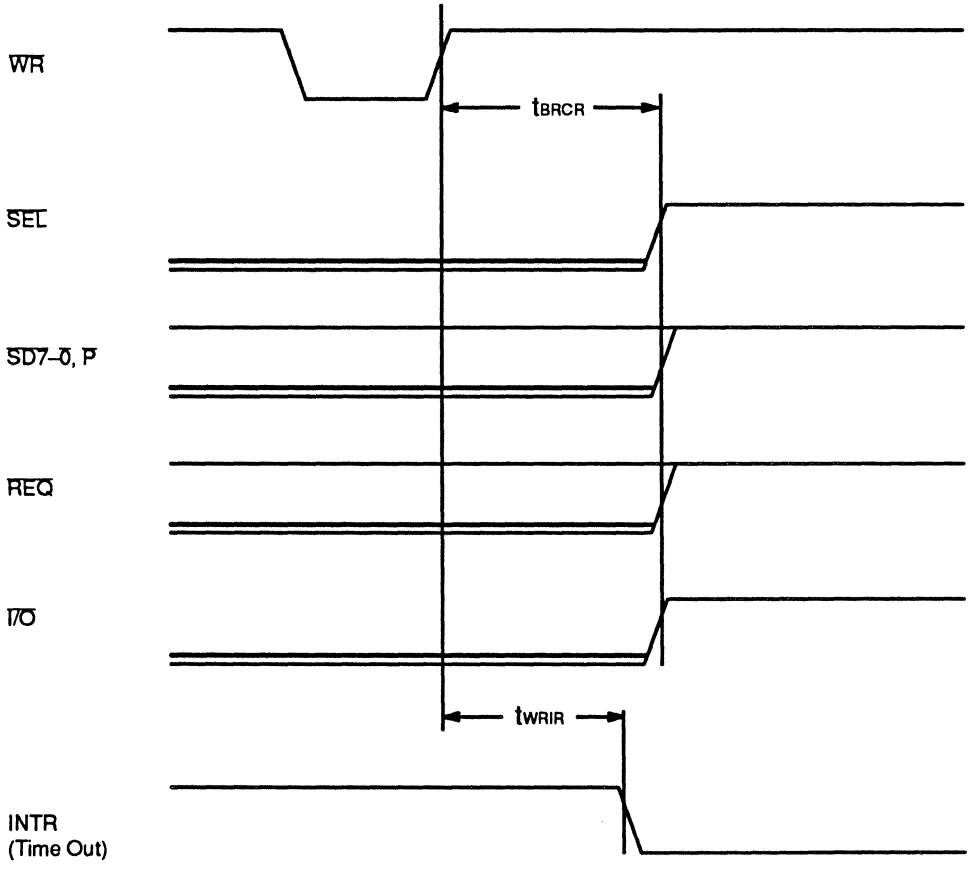
TARGET (BUS RELEASE COMMAND)					
Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
WR High to Bus Clear (Bus Release Command)	t_{BRCR}			$3t_{CLF}+100$	ns



AC CHARACTERISTICS (Continued)

TERMINATION (TIME OUT) – SELECTION AND RESELECTION PHASES					
Parameter	Designator	Values			Unit
		Min.	Typ.	Max.	
$\overline{WR}$ High to $\overline{SEL}$, $\overline{SD7-0}$, $\overline{P}$, $\overline{I/O}$ High (Reset Time Out Interruption)	t_{BRCR}			$3t_{CLF}+100$	ns
$\overline{WR}$ High to INTR Low	t_{WRIR}			$3t_{CLF}+60$	

Timeout Timing (Selection/Reselection Phases)



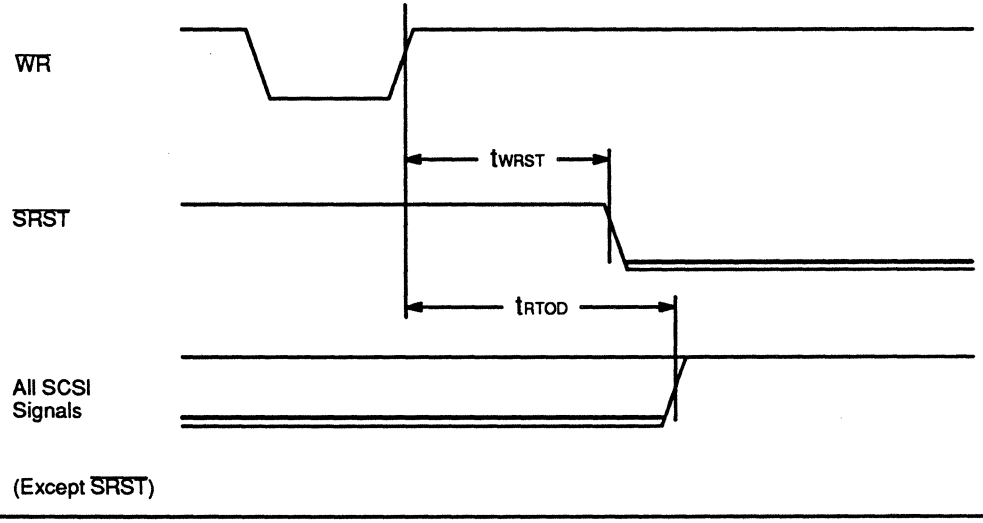
AC CHARACTERISTICS (Continued)

SCSI BUS INTERFACE – RESET CONDITION

$\overline{SRST}$ - RESET CONDITION (OUTPUT)

Parameter	Designator	Values			Unit
		Min	Typ	Max	
$\overline{WR}$ High to $\overline{SRST}$ Low (Write "1" to SCMD Bit-4)	t_{WRST}	10		110	ns
Reset Delay	t_{RTOD}			140	

$\overline{SRST}$ Signal Transfer Timing

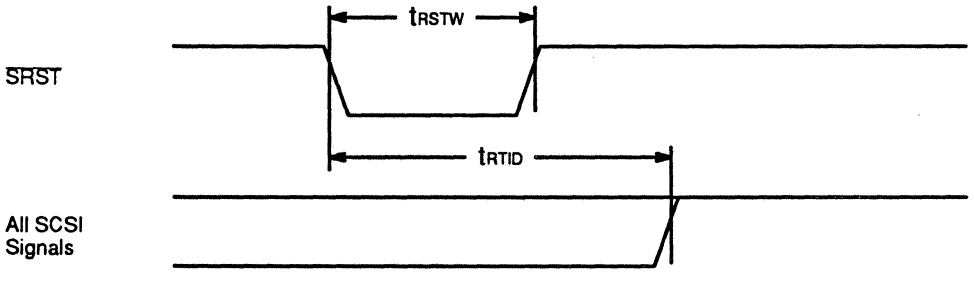


AC CHARACTERISTICS (Continued)

SCSI BUS INTERFACE – RESET CONDITION

$\overline{SRST}$ - RESET CONDITION (INPUT)					
Parameter	Designator	Values			Unit
		Min	Typ	Max	
$\overline{SRST}$ Pulse Width	t_{RSTW}	$3t_{CLF}$			ns
Reset Delay	t_{RTID}			$4t_{CLF}$+ 200	

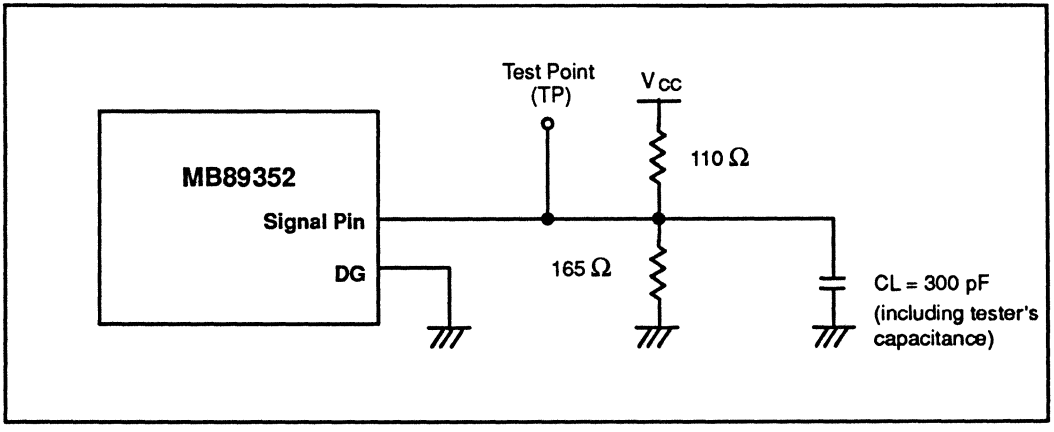
$\overline{SRST}$ Signal Acceptance Timing



AC CHARACTERISTICS (Continued)

CAPACITANCE			
Parameter	Values		Unit
	Typ.	Max.	
D7 – D0, DP	—	80	pF
DPO, INTR, DREQ	10	30	
$\overline{SD\&}$ - $\overline{SDO}$, $\overline{SDP}$	—	300	
$\overline{SRST}$, $\overline{SEL}$, $\overline{BSY}$, $\overline{I/O}$, $\overline{C/D}$, $\overline{MSG}$, $\overline{REQ}$, $\overline{ACK}$, $\overline{ATN}$	—	300	

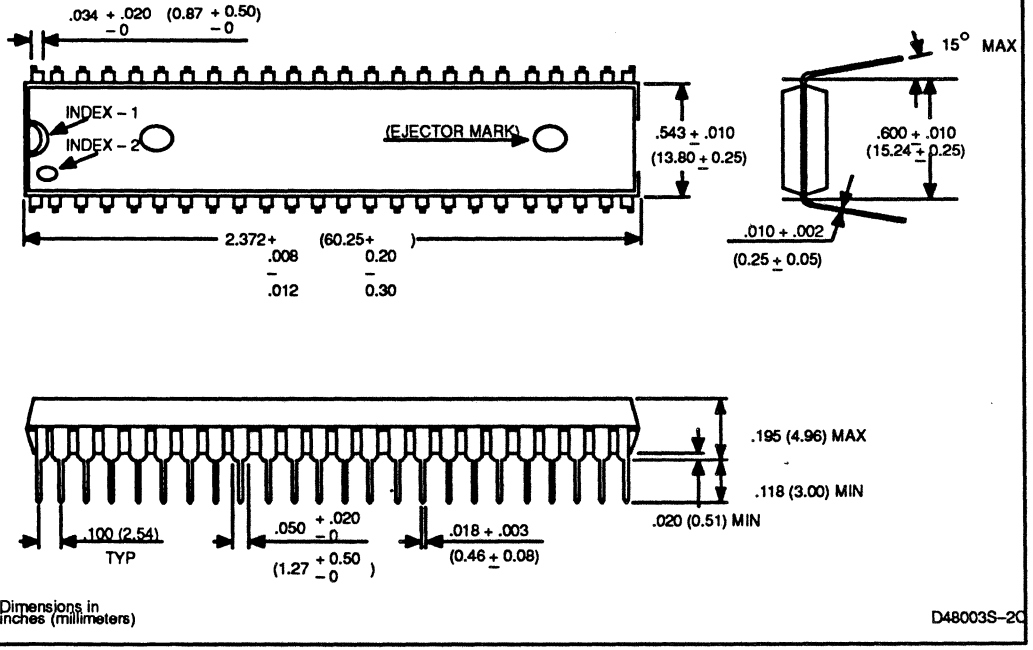
The AC characteristics of all SCSI bus signal pins are measured on the following test circuit.



PACKAGE DIMENSIONS

DIP–48P–M02

48–LEAD PLASTIC DUAL–IN–LINE PACKAGE
(CASE NO.: DIP–48P–M02)



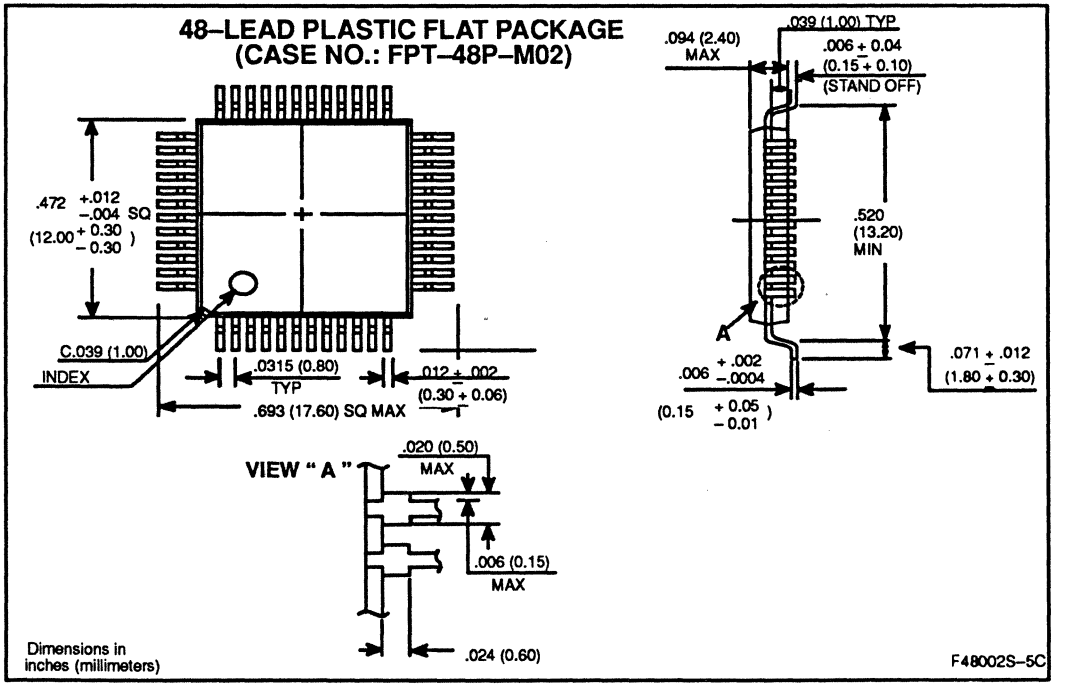
Dimensions in inches (millimeters)

D48003S–2C

PACKAGE DIMENSIONS (Continued)

48–Lead Plastic Flat Package

FPT–48P–M02



Appendix 1

SPC DRIVER EXAMPLES IN 8086 ASSEMBLY LANGUAGE

The following code is a group of routines, written in 8086 assembly language, that demonstrate the essential elements necessary for the implementation of a SCSI driver using Fujitsu's family of SCSI controllers. These routines make certain assumtions about the hardware interface, but are software compatible with all the Fujitsu SCSI controllers.

For this example the controller is memory mapped somewhere in the 1MB address space of the 8086. This example assumes that the controller will be used in a polled environment (not interrupt driven) using programmed I/O (no DMA). Using the device in this mode requires much more code than an interrupt driven/DMA environment but the code will be less dependent on system hardware. At the end of this example is a re–write of the data transfer routine using DMA transfers instead or programmed I/O. As can be seen the code is considerably smaller and simpler.

These routines use short (16–bit) pointers to keep the code relatively simple, but this limits the size of the transfer buffers to 64K bytes (due to the 8086 family's segmentation limitation). Even though the transfer routines and the SPC will take up to a 24–bit (28–bit MB87033) transfer length, allowing transfers up to 16M bytes, this size cannot be accomodated without re–writing the transfer routines to support segmented (20–bit) or 80386 extended (32–bit) pointers.

This code is intended to show the programming and operation of the major functions of the Fujitsu family of SCSI controllers. It is not intended as a tutorial of the SCSI protocol or to show the construction of a driver for an SCSI host adapter.

This code was tested for syntax errors under the following assemblers, using the SMALL model

Borland TASM 1.0

Microsoft MASM 5.1;

```
;*************************************************************;
;                                                             ;
;           For detailed information on the SCSI protocol see the ;
;           spec entitled                                     ;
;                                                             ;
;           ANSI Small Computer System Interface (SCSI)       ;
;           ANSI X3.131-1986                                  ;
;                                                             ;
;           American National Standards Institute, Inc.       ;
;           1430 Broadway                                     ;
;           New York, NY 10018                                ;
;                                                             ;
;*************************************************************;
;                                                             ;
;           For information on Fujistu's SCSI controllers     ;
;           contact:                                          ;
;                                                             ;
;           Fujitsu Microelectronics, Inc.                    ;
;           3545 North First Street                           ;
;           San Jose, CA.95134-1804                           ;
;           (408) 922-9000                                    ;
;                                                             ;
;           or your local Fujitsu representative.             ;
;                                                             ;
;*************************************************************;
;                                                             ;
;                                                             ;
;           Copyright (C) 1989 Fujitsu Microelectronics Inc.  ;
;           Copyright (C) 1989 Galbo and Associates, Inc.     ;
;                                                             ;
;                                                             ;
;                                                             ;
*************************************************************;
```

```
                        .model small
                        .code

;*************************************************************;
;       equates                                               ;
;*************************************************************;
;
;       SPC memory mapped register addresses
;
BDID    equ     <byte ptr es:[bx+0]>  ; bus ID
SCTL    equ     <byte ptr es:[bx+1]>  ; SPC control
SCMD    equ     <byte ptr es:[bx+2]>  ; SPC command
TMOD    equ     <byte ptr es:[bx+3]>  ; transfer mode
INTS    equ     <byte ptr es:[bx+4]>  ; interrupt sense
PSNS    equ     <byte ptr es:[bx+5]>  ; phase sense
SDGC    equ     <byte ptr es:[bx+5]>  ; SPC diagnostic control
SSTS    equ     <byte ptr es:[bx+6]>  ; SPC status
SERR    equ     <byte ptr es:[bx+7]>  ; SPC error status
PCTL    equ     <byte ptr es:[bx+8]>  ; phase control
MBC     equ     <byte ptr es:[bx+9]>  ; modified byte counter
DREG    equ     <byte ptr es:[bx+10]> ; data register (FIFO)
TEMP    equ     <byte ptr es:[bx+11]> ; temp register
TCH     equ     <byte ptr es:[bx+12]> ; transfer counter high
TCM     equ     <byte ptr es:[bx+13]> ; transfer counter middle
TCL     equ     <byte ptr es:[bx+14]> ; transfer counter low
EXBF    equ     <byte ptr es:[bx+15]> ; external buffer register
SPC_OFFSET    = 1ff0h                 ; assume SPC is memory mapped
SPC_SEGMENT   = 0dc00h                ; at this address 9seg:ofs0
;
;       Data transfer mode
;
READ                    = 0     ; read mode for data transfer routines
WRITE                   = 1     ; write mode for data transfer routines
;
;
SCSI    time-out values
;

SELECTION_TIMEOUT       = 4400  ; 275 msec. @ 8MHz clock
BUS_DELAY               = 4     ; 1200 nsec. @ 8MHz clock
;
```

```
;       Selection return values
;
SELECTED                = 100   ; target successfully selected
LOST_ARBITRATION        = -100  ; lost control of SCSI bus to another
                                  INITIATOR
TIMEOUT                 = -101  ; selected target did not respond
ERROR_INTR              = -102  ; hardware error occured during selection
;
;       Data transfer return values
;

OK                      = 200   ; transferred all bytes without error
OKPAD                   = 201   ; transferred all bytes plus padding
SERVICE_REQUIRED        = -200  ; early phase change, didn't transfer all
PARITY_ERROR            = -201  ; parity error during transfer
;
;       Phase decoder return values
;
NO_PHASE_REQUESTED      = 0     ; target's REQ signal negated
ILLEGAL_PHASE           = -300  ; target requesting undefined phase
```

```
;*************************************************************
;       constants                                             ;
;*************************************************************
.                       .data                   ; data area
Addr2ID                 db      1, 2, 4, 8, 16, 32, 64, 128
SPC_Pointer             label   dword
SPC_Ofs                 dw      SPC_OFFSET
SPC_Seg                 dw      SPC_SEGMENT

;*************************************************************
;       global variables                                      ;
;*************************************************************
                        .data?              ; unitialize data area
@datalength             label   dword       ; length of transfer for DATA
                                              IN/OUT phase
@datalength_lo          dw      0
@datalength_hi          dw      0
@dataptr                dw      0           ; pointer to byte buffer for DATA
                                              IN/OUT
@cmdlength              dw      0           ; length of command block
@cmdblk                 db      12 dup (0) ; command block, max = 12
@msgout                 db      0           ; message out byte
@msgin                  db      0           ; message in byte
@status                 db      0           ; status in byte

;*************************************************************
;       external routines                                     ;
;*************************************************************
                        .code
                        extrn   Setup_DMA:near ; setup system's DMA
                                               controller
```

```
;*************************************************************************;
;                                                                         ;
;       INITIALIZE SPC;                                                   ;
;       Initialize the Fujitsu SPC;                                       ;
;                                                                         ;
;       Entry: BYTE @ [bp+4] = initiator's SCSI address                   ;
;                                                                         ;
;                                                                         ;
;       Exit:  void;                                                      ;
;                                                                         ;
;*************************************************************************

InitSPC_record  struc
                dw      ?               ; bp
                dw      ?               ; ip - return address
InitAddr        db      ?               ; initiator's SCSI address
InitSPC_record  ends

InitSPC         proc    near
                push    bp
                mov     bp,sp
                les     bx,SPC_Pointer  ; point to SPC
                mov     SCTL,0c0h       ; reset SPC
                mov     al,[bp].InitAddr ;get initiator address
                mov     BDID,al         ; write initiator's addr
                                          into ID reg
                mov     TMOD,0          ; clear reg
                mov     SCTL,01ah       ; enable arbitration, parity
                                        ; ... and reselection
                mov     SDGC,0          ; clear diagnostics reg
                pop     bp
                ret
InitSPC         endp
```

```
;*****************************************************************;
;       SELECT TARGET DEVICE                                      ;
;       Select a target device on the SCSI bus with ATN line asserted;
;       Check if the target responds to the ATN line with the message;
;       out phase. If so, send an IDENTIFY message to the target.  ;
;                                                                 ;
;       Entry: BYTE @ [bp+4] = target SCSI address,  0..7         ;
;              BYTE @ [bp+6] = target SCSI logical unit number, 0..7 ;
;                                                                 ;
;       Exit:  AX = status message                                ;
 *****************************************************************
Select_record   struc
                dw    ?               ; bp
                dw    ?               ; ip - return address
TargAddr        db    ?,?             ; target address
TargLUN         db    ?,?             ; target LUN
Select_record   ends

Select          proc  near
                push  bp
                mov   bp,sp
                les   bx,SPC_Pointer  ; point to SPC
                mov   PCTL,0          ; phase control = out indicates
                                      ; ... selection phase
                mov   al,BDID         ; get initiator's SCSI ID,1..128
                push  bx              ; save SPC ptr
                xor   bh,bh
                mov   bl,[bp].TargAddr; get target SCSI address, 0..7
                or    al,Addr2ID[bx]  ; convert to SCSI ID, 1..128
                                      ; ...OR with initiator's ID
                                      ; ...(for reselection, see spec.)
                pop   bx              ; [es:bx] -> SPC
                mov   TEMP,al         ; put arbitration ID's in temp reg
                mov   TCH,high(SELECTION_TIMEOUT) ; load selection
                                                  time-out
                mov   TCM,low(SELECTION_TIMEOUT) ; ... value
                mov   TCL,BUS_DELAY   ; load bus free delay time-out
                mov   SCTL,060h       ; set ATN line, select with
                                        attention
                mov   SCTL,010h       ; start selection operation
```

```
SelectWaitIntr:
                cmp     INTS,0          ; any intr bit set?
                jne                     SelectIntr
;               yes
                test    SSTS,020h       ; no, select operation still
                                          pending?
                                          ;
                                        ... check for lost arbitration
                jz                      SelectWaitIntr
;               still pending
                mov     ax,LOST_ARBITRATION ; we've lost the arbitration
                jmp     SelectExit
SelectIntr:
                test    INTS,10h        ; command complete intr?
                jz                      SelectTimeout
;               no, check time-out
                mov     al,PSNS         ; yes, read phase
                and     al,087h         ; check REQ & phase bits
                cmp     al,086h         ; target requesting msg out?
                jne                     SelectNoMsg
;               no, skip msg out phase
                mov     al,0c0h         ; yes, send IDENTIFY message
                                        ;... with disconnect/reselect bits
                or      al,[bp].TargLUN ; ... and target LUN
                mov     @msgout,al      ; save in msg out byte
                xor     ah,ah           ; clear upper
                push    ax              ; send AX message to target
                call    Message_Out     ; send message
                pop     cx              ; remove parameter from stack
SelectNoMsg:
                jmp     SelectOK
SelectTimeout:
                test    INTS,04h        ; time-out intr occured?
                 jz     SelectOtherIntr ; no, unexpected interrupt
                mov     ax,TIMEOUT      ; return time-out error
                jmp                     SelectExit
SelectOtherIntr:
                mov     ax,ERROR_INTR   ; return error
                jmp     SelectExit
SelectOK:
                mov     ax,SELECTED     ; successful selection
SelectExit:
                pop     bp
                ret
Select          endp
```

```
;*************************************************************************;
;   TRANSFER DATA TO/FROM SCSI BUS                                        ;
;   General data transfer for all data transfer phases.                   ;
;   Read or write a stream of bytes to/from the SCSI bus and check        ;
;   for the following conditions:                                         ;
;                                                                         ;
;   1) # of bytes requested == # of bytes transferred - normal            ;
;   2) # of bytes requested > # of bytes transferred - pad                ;
;   3) # of bytes requested < # of bytes transferred - phase change       ;
;   This routine, and the SPC,  will accept a 24-bit number as a          ;
;   transfer length, allowing up to 16M bytes to be read or               ;
;   written to the SCSI device in one data phase transaction.             ;
;                                                                         ;
;   Entry:    WORD @ [bp+4] - pointer to list of bytes                    ;
;             LONG @ [bp+6] - length - number of bytes in list            ;
;             BYTE @ [bp10] - transfer type - READ=0 or WRITE=1           ;
;                                                                         ;
;   Exit:     AX - error status                                           ;
;             AX > 0 - transferred all bytes in list                      ;
;             AX < 0 - transfer terminated early                          ;
;                                                                         ;
;*************************************************************************;
Transfer_record     struc
                    dw      ?           ; bp
                    dw      ?           ; ip
data_ptr            dw      ?           ; ptr to byte to transfer
length_lo           dw      ?           ; LSW length of transfer
length_hi           dw      ?           ; MSW length of transfer
direction           dw      ?           ; direction, READ=0, WRITE=1
Transfer_record     ends

Transfer_Data       proc    near
                    push    bp
                    mov     bp,sp
                    cld                         ; auto increment di,si
                    mov     si,[bp].data_ptr; point to stream of bytes
                    mov     di,si               ; di also, for read operation
                    les     bx,SPC_Pointer  ; point to SPC
                    mov     al,byte ptr [bp].length_hi ; get length MSW
                    mov     TCH,al              ; load counter high
                    mov     ax,[bp].length_lo ; get length MSW
                    mov     TCM,ah              ; load counter mid
                    mov     TCL,al              ; load counter low
                    mov     SCMD,084h           ; do programmed i/o transfer
```

```
XferCmdWait:
                mov     al,SSTS         ; read status, wait for
                                          tranfer
                and     al,0f0h         ; ... operation to start
                cmp     al,0b0h         ; connected as init and
                                        ; ... transfer in progress
                                          bit set?
                  jne   XferCmdWait     ; no, wait for SPC to start
XferLoop:
                cmp     INTS,0          ; intr occured?
                  jne   XferCheckIntr   ; yes
                cmp     [bp].direction,WRITE ; writing?
                  jne   Reading         ; no, reading
Writing:
                test    SSTS,02h        ; fifo not full?
                  jne   XferLoop        ; no wait, fifo is full
                lodsb                   ; read byte, inc si
                mov     DREG,al         ; write byte to fifo
                jmp     XferDec         ; decrement and check count
Reading:
                test    SSTS,1          ; fifo not empty?
                  jne   XferLoop        ; no, wait for byte
                mov     al,DREG         ; get byte
                stosb                   ; write byte, inc di
                jmp     XferDec         ; decrement/check count
XferCheckIntr:
                test    INTS,08h        ; service required intr?
                  je    TestErrIntr     ; no
                mov     INTS,08h        ; yes, clear it
                mov     ax,SERVICE_REQUIRED ; exit with error
                jmp     XferExit
TestErrIntr:
                cmp     INTS,0          ; any more intr pending?
                  je    TestParityErr   ; no
                mov     ax,ERROR_INTR   ; yes, exit with error
                jmp     XferExit
TestParityErr:
                test    SERR,0c0h       ; parity error?
                  je    XferLoop        ; no, wait for bytes
                mov     ax,PARITY_ERROR ; yes, exit with error
                jmp     XferExit
```

```
XferDec:
                sub     [bp].length_lo,1; decrement count
                sbb     [bp].length_hi,0; 32-bit count
                mov     ax,[bp].length_lo ; get length
                or      ax,[bp].length_his; =0?
                   jne  XferLoop          ; no, continue
XferDone:
                test    INTS,010h         ; command complete intr?
                   je   XferDone          ; no, wait
                mov     INTS,010h         ; yes, reset intr
                mov     al,PCTL           ; read phase control
                and     al,7              ; get bus phase bits
                or      al,080h           ; add REQ bit
                mov     dl,PSNS           ; read phase sense reg
                and     al,087h           ; mask phase bits & REQ
                cmp     al,dl             ; are we in new phase?
                mov     ax,OK             ; assume good status
                   jne  XferExit          ; jmp if in new phase
                mov     TEMP,0            ; no, load temp with padding
                                            value
                mov     SCTL,085h         ; do padding transfer
                                          ; ...to cause target to go to
                                          ; ... next phase
XferPad:
                mov     al,INTS
                and     al,018h           ; wait for cmd complete & svc
                                            req
                cmp     al,018h
                   jne  XferPad           ; no, wait for pad to finish
                mov     INTS,18h          ; reset intrs
                mov     ax,OKPAD          ; return good with PAD status
XferExit:
                pop     di
                pop     si
                pop     bp
                ret
Transfer_Data   endp
```

```
                                        RD, RDG
                                        WT, WTG
;*************************************************************************;
;   SEND COMMAND                                                          ;
;   Send a SCSI command block to the target device.                       ;
;   Assume TCH, TCM and TCL registers = 0                                 ;
;                                                                         ;
;   Entry:    WORD @ [bp+4] - pointer to SCSI command block               ;
;             WORD @ [bp+6] - length of command block, 6, 10, 12 bytes    ;
;                                                                         ;
;   Exit:     AX  - transfer status                                       ;
;             > 0 - transferred all bytes in list                         ;
;             < 0 - transfer terminated early, error                      ;
;*************************************************************************;
Select_record   struc
                dw    ?               ; bp
                dw    ?               ; ip
cmd_ptr         dw    ?               ; ptr to cmd bytes
cmd_length      dw    ?               ; length of cmd
Select_record   ends

Send_Command    proc  near
                push  bp
                mov   bp,sp
                les   bx,SPC_Pointer  ; point to SPC
                mov   PCTL,02h        ; set command out phase
                mov   ax,WRITE        ; write to target for transfer
                push  ax              ; put parameters on stack
                xor   ax,ax
                push  ax              ; MSW of length = 0
                push  [bp].cmd_length ; LSW length of cmd
                push  [bp].cmd_ptr    ; ptr to cmd bytes
                call  Transfer_Data   ; do SCSI transfer
                mov   sp,bp           ; AX has the transfer status
                pop   bp
                ret
Send_Command    endp
```

```
;*************************************************************************
;                                                                        ;
;       READ SCSI STATUS                                                 ;
;       Read the 1 byte status code from the target                      ;
;                                                                        ;
;       Entry: WORD @ [bp+4] - pointer to byte to hold SCSI status       ;
;                                                                        ;
;       Exit:  AX - transfer status                                      ;
;              > 0 - transferred SCSI status to pointer location         ;
;              < 0 - transfer terminated early, error                    ;
;*************************************************************************

Status_record   struc
                dw      ?               ; bp
                dw      ?               ; ip
status_ptr      dw      ?               ; ptr to status byte
Status_record   ends

Status_In       proc    near
                push    bp
                mov     bp,sp
                les     bx,SPC_Pointer  ; point to SPC
                mov     PCTL,03h        ; set phase to status in
                mov     ax,READ         ; read mode
                push    ax
                xor     ax,ax
                push    ax              ; MSW of transfer length
                inc     ax              ; LSW of length = 1 byte
                push    ax
                push    [bp].status_ptr ; pointer to status byte
                call    Transfer_Data   ; read 1 byte from SCSI target
                mov     sp,bp           ; AX = result code
                pop     bp
                ret
Status_In       endp
```

```
;*****************************************************************;
;                                                                 ;
;     SEND MESSAGE OUT                                            ;
;     Send a 1 byte message the the target device.                ;
;     When the SPC is in the INITIATOR mode, it will automatically ;
;     negate the ATN signal after the message byte has been       ;
;     successfully sent to the target device, this is in comformance ;
;     with the SCSI spec. (See spec. for further details)         ;
;                                                                 ;
;     Entry:  WORD @ [bp+4] - pointer to message byte to send     ;
;                                                                 ;
;     Exit:   AX - transfer status                                ;
;             > 0 - transferred message byte to target device     ;
;             < 0 - transfer terminated early, error              ;
;*****************************************************************;


Message_Out_rec struc
                dw    ?               ; bp
                dw    ?               ; ip
msg_out_ptr     dw    ?               ; ptr to message out byte
Message_Out_rec ends

Message_Out     proc  near
                push  bp
                mov   bp,sp
                les   bx,SPC_Pointer  ; point to SPC
                mov   PCTL,06h        ; set desired phase to message out
                mov   ax,WRITE        ; send byte to SCSI bus
                push  ax              ; param's on stack
                xor   ax,ax
                push  ax              ; MSW of transfer count
                inc   ax              ; LSW = 1 byte transfer
                push  ax
                push  [bp].msg_out_ptr; pointer to message byte to send
                call  Transfer_Data   ; send it
                mov   sp,bp           ; AX has transfer status
                pop   bp
                ret
Message_Out     endp
```

```
;*************************************************************
;     READ MESSAGE IN                                        ;
;     Read a 1 byte message from the SCSI target device.     ;
;     When the SPC is in the INITIATOR mode it will automatically ;
;     hold the ACK line active on the last byte of the message ;
;     received from the target device. This is to allow the  ;
;     INITIATOR to interpret the message and accept or, if   ;
;     necessary,reject it according to the SCSI protocol's message ;
;     reject sequence. (See spec. for further details)       ;
;                                                            ;
;     Entry:  WORD @ [bp+4] - pointer to address to hold message byte ;
;                                                            ;
;     Exit:   AX - transfer status                           ;
;             > 0 - transferred SCSI message to pointer location ;
;             < 0 - transfer terminated early, error;        ;
;*************************************************************

Message_In_rec  struc
                dw    ?               ; bp
                dw    ?               ; ip
msg_in_ptr      dw    ?               ; ptr to message in byte address
Message_In_recends

Message_Inprocnear
                push  bp
                mov   bp,sp
                les   bx,SPC_Pointer  ; point to SPC
                mov   PCTL,06h        ; set desired phase to message in
                mov   ax,READ         ; read byte from device
                pus   hax
                xor   ax,ax           ; MSW of transfer length
                push  ax
                inc   ax              ; LSW of length = 1 byte
                push  ax
                push  [bp].msg_in_ptr ; ptr to message byte
                call  Transfer_Data   ; read message
                add   sp,6            ; remove param's from stack
                or    ax,ax           ; if < 0 then error
                 jl   MIError         ; error
                les   bx,SPC_Pointer  ; no error, point to SPC
                mov   SCMD,0c0h       ; manually negate the ACK signal
                mov   INTS,010h       ; reset command complete intr
MIError:
                pop   bp
                ret
Message_In      endp
```

```
;*****************************************************************;
;                                                                 ;
;       READ DATA STREAM FROM SCSI DEVICE                         ;
;       Read up to 16M bytes from the target device.              ;
;                                                                 ;
;       Entry: WORD @ [bp+4] - pointer to data buffer             ;
;              LONG @ [bp+6] - length of transfer (24-bit)        ;
;                                                                 ;
;       Exit:  AX - transfer status                               ;
;              > 0 - read byte from SCSI device                   ;
;              < 0 - transfer terminated early, error             ;
;*****************************************************************;

Data_In_record  struc
                dw    ?           ; bp
                dw    ?           ; ip
data_in_ptr     dw    ?           ; ptr to byte to transfer
data_in_len_lo  dw    ?           ; length of transfer
data_in_len_hi  dw    ?
Data_In_record  ends

Data_In         proc  near
                push  bp
                mov   bp,sp
                les   bx,SPC_Pointer  ; point to SPC
                mov   PCTL,01h        ; desired phase = data in
                mov   ax,READ         ; read bytes from target device
                push  ax
                push  [bp].data_in_len_hi; MSW of length
                push  [bp].data_in_len_lo; LSW of length
                push  [bp].data_in_ptr; pointer to receive buffer
                call  Transfer_Data   ; read bytes
                mov   sp,bp           ; AX has result code
                pop   bp
                ret
Data_In         endp
```

```
;*************************************************************
;                                                            ;
;       WRITE DATA STREAM TO SCSI DEVICE                     ;
;       Write up to 16M bytes to the target device.          ;
;                                                            ;
;       Entry: WORD @ [bp+4] - pointer to data buffer        ;
;              LONG @ [bp+6] - length of transfer (24-bit)   ;
;                                                            ;
;       Exit:  AX - transfer status                          ;
;              > 0 - wrote bytes to SCSI device              ;
;              < 0 - transfer terminated early, error        ;
;*************************************************************;

Data_Out_record struc
                dw    ?           ; bp
                dw    ?           ; ip
data_out_ptr    dw    ?           ; ptr to byte to transfer
data_out_len_lo dw    ?           ; LSW length of transfer
data_out_len_hi dw    ?           ; MSW
Data_Out_record ends

Data_Out        proc  near
                push  bp
movbp,sp
                les   bx,SPC_Pointer  ; point to SPC
                mov   PCTL,0          ; set desired phase to data out
                mov   ax,WRITE        ; write bytes to target device
                push  ax
                push  [bp].data_out_len_hi ; MSW of length
                push  [bp].data_out_len_lo ; LSW of length
                push  [bp].data_out_ptr; address of bytes to send
                call  Transfer_Data   ; send bytes to target
                mov   sp,bp           ; AX has result code
                pop   bp
                ret
Data_Out        endp
```

```
;*************************************************************************
;                                                                        ;
;       DECODE SCSI BUS PHASE                                            ;
;       After selection, read target's requested SCSI bus phase          ;
;       and go to the appropriate handler, continue calling this         ;
;       routine until the COMMAND COMPLETE message is sent from the      ;
;       target, indicating the end of the SCSI operation.                ;
;                                                                        ;
;       Entry: void                                                      ;
;                                                                        ;
;       Exit:  > 0 successful operation                                  ;
;              = 0 no phase requested (REQ signal not active)            ;
;              < 0 unsuccessful operation                                ;
;                                                                        ;
;*************************************************************************
Decode_Phase    proc  near
                les   bx,SPC_Pointer  ; point to SPC
                mov   al,PSNS         ; read phase sense
                and   al,087h         ; mask REQ + phase
                sub   al,080h         ; subtract bias
                cmp   al,7            ; decode 0..7
                 jbe  DPA
                jmp   NoPhaseReq
DPA:
                mov   bl,al           ; make index into table
                xor   bh,bh           ; 16-bit
                shl   bx,1            ; * 2 = jump table ptr
                jmp   cs:JmpTable[bx] ; go to routine

JmpTable        label word
                dw    p_data_out      ; target requesting data out phase
                dw    p_data_in       ; target requesting data in phase
                dw    p_send_command  ; target requesting command phase
                dw    p_status_in     ; target requesting status phase
                dw    p_illegal       ; reserved
                dw    p_illegal       ; reserved
                dw    p_message_out   ; target requesting msg out phase
                dw    p_message_in    ; target requesting msg in phase
p_data_out:                           ; send data to target device
                push  @datalength_hi  ; MSW number of bytes to send
                push  @datalength_lo  ; LSW
                push  @dataptr        ; ptr to bytes to send
                call  Data_Out        ; write bytes
                add   sp,6            ; remove stack parameters
                jmp   DecodeDone      ; return with AX = transfer status
```

```
p_data_in:                          ; receive data from target device
                push  @datalength_hi  ; MSW number of bytes to receive
                push  @datalength_lo  ; LSW
                push  @dataptr        ; ptr to byte buffer
                call  Data_In         ; read bytes
                add   sp,6            ; remove stack parameters
                jmp   DecodeDone      ; AX = transfer status
p_send_command:                     ; send SCSI command block to target
                push  @cmdlength      ; number of bytes in cmd blk
                mov   ax,offset @cmdblk; ptr to cmd blk
                push  ax              ; stack it
                call  Send_Command    ; send bytes
                add   sp,6            ; remove stack param's
                jmp   DecodeDone      ; AX = status
p_status_in:                        ; read SCSI status from target
                mov   ax,offset @status ; ptr to status bytes
                push  ax              ; stack it
                call  Status_In       ; read status byte
                pop   cx              ; remove stack param's
                jmp   DecodeDone      ; AX = status
p_illegal:                          ; undefined/reserved SCSI phases
                mov   ax,ILLEGAL_PHASE ; AX = error
                jmp   DecodeDone
p_message_out:                      ; send message to target
                mov   ax,offset @msgout; ptr to msg byte
                push  ax              ; stack it
                call  Message_Out     ; send it
                pop   cx              ; remove param's
                jmp   DecodeDone      ; AX = status
p_message_in:                       ; read message from target
                mov   ax,offset @msgin ; ptr to msg byte
                push  ax              ; stack it
                call  Message_In      ; read it
                pop   cx              ; remove param's
                jmp   DecodeDone      ; AX = status
NoPhaseReq:
                xor   ax,ax           ; target not requesting, REQ
                                        negated
DecodeDone:
                ret                   ; return to caller, AX=transfer
                                        status
Decode_Phaseendp
```

```
;*****************************************************************;
;                                                                 ;
;     MANUALLY TRANSFER DATA TO/FROM SCSI BUS                     ;
;     Read or write a stream of bytes to/from the SCSI bus and check ;
      for completion.                                             ;
;                                                                 ;
;     This example illustrates the use of the manual transfer mode ;
;     of the SPC. This mode is useful for "spoon-feeding" each    ;
byte to a target device, this could be helpful when debugging     ;
;     SCSI software step-by-step or testing the behavior of a SCSI ;
;     peripheral under certain conditions.                        ;
;                                                                 ;
;     This mode of operation does not use the transfer counter of ;
;     the SPC. The transfer size is limited only by the size of the ;
;     transfer length parameter passed to this routine.           ;
;                                                                 ;
;     Entry:  WORD @ [bp+4] - pointer to list of bytes            ;
;             LONG @ [bp+6] - length - number of bytes in list    ;
;             BYTE @ [bp10] - transfer type - READ=0 or WRITE=1   ;
;                                                                 ;
;     Exit:   AX - error status                                   ;
;             AX > 0 - transferred all bytes in list              ;
;             AX < 0 - transfer terminated early                  ;
;*****************************************************************
Manual_Xfer_rec     struc
                    dw      ?       ; bp
                    dw      ?       ; ip
Man_data_ptr        dw      ?       ; ptr to byte to transfer
Man_length_lo       dw      ?       ; length of transfer, up to 4G bytes
Man_length_hi       dw      ?       ; ... 32-bits
Man_direction       dw      ?       ; direction, READ=0, WRITE=1
Manual_Xfer_rec     ends

Man_Xfer_Data       proc    near
                    push    bp
                    mov     bp,sp
                    cld                 ;  auto increment di,si
                    mov     si,[bp].data_ptr; point to stream of bytes
                    mov     di,si       ;  di also, for read operation
                    les     bx,SPC_Pointer  ; point to SPC

ManXfer:
                    cmp     [bp].Man_direction,READ ; reading ?
                     je     ManRead     ;  yes
```

```
ManWrite:
                test    PSNS,80h        ;  wait for REQ asserted
                 jz     ManWrite        ;  not yet
                lodsb                   ; read byte from buffer, inc si
                mov     TEMP,al         ;  load byte into temp register
                mov     SCMD,0e0h       ;  assert ACK
MWNoReq:
                test    PSNS,80h        ;  wait for REQ negated
                                        ;  ... indicating byte accepted
                 jnz    MWNoReq         ;  not yet
                mov     SCMD,0c0h       ;  negate ACK
                jmp     ManDec          ;  decrement count
ManRead:
                test    PSNS,80h        ;  wait for REQ asserted
                 jz     ManRead         ;  not yet
                mov     SCMD,0e0h       ;  assert ACK
MRNoReq:
                test    PSNS,80h        ;  wait for REQ negated
                                        ;  ... indicating byte sent
                 jnz    MRNoReq         ;  not yet
                mov     al,TEMP         ;  read byte from temp
                stosb                   ;  write to buffer, inc di
                mov     SCMD,0c0h       ;  negate ACK
ManDec:
                sub     [bp].Man_length_lo,1    ; decrement count
                sbb     [bp].Man_length_hi,0    ; 32-bit count
                mov     ax,[bp].Man_length_lo   ; get length
                or      ax,[bp].Man_length_hi   ; =0?
                 jne    ManXfer                 ; not done
ManXferExit:
                mov     ax,OK
                pop     bp
                ret
Man_Xfer_Data   endp
```

```
;*************************************************************;
;                                                             ;
;   DMA TRANSFER DATA TO/FROM SCSI BUS                        ;
;   Read or write a stream of bytes to the SCSI bus using DMA ;
;   and check for the following conditions:                   ;
;                                                             ;
;   1) # of bytes requested == # of bytes transferred - normal ;
;   2) # of bytes requested  > # of bytes transferred - pad    ;
;   3) # of bytes requested < # of bytes transferred - phase change ;
;   Entry:    WORD @ [bp+4] - pointer to list of bytes        ;
;             LONG @ [bp+6] - length - number of bytes in list ;
;             BYTE @ [bp10] - transfer type - READ=0 or WRITE=1 ;
;                                                             ;
;   Exit:     AX - error status                               ;
;             AX > 0 - transferred all bytes in list          ;
;             AX < 0 - transfer terminated early              ;
;                                                             ;
;   Note: If the SPC is interrupt driven along with using DMA, ;
;             the code below becomes even smaller since it is not ;
;             necessary to test the INTS register within a loop. ;
;             In fact, the overhead to start a data transfer in ;
;             the SPC takes only 3 steps                      ;
;                                                             ;
;   1) Write desired phase to PCTL register                   ;
;   2) Write 1 to 3 bytes to the transfer counters, TCH/M/L   ;
;   3) Write the DMA transfer cmd to SCMD register (0x80)     ;
;                                                             ;
;*************************************************************;
DMA_Xfer_record     struc
                    dw      ?           ; bp
                    dw      ?           ; ip
DMA_data_ptr        dw      ?           ; ptr to byte to transfer
DMA_length_lo       dw      ?           ; LSW length of transfer
DMA_length_hi       dw      ?           ; MSW length of transfer
DMA_direction       dw      ?           ; direction, READ=0, WRITE=1
DMA_Xfer_record     ends

DMA_Transfer_Data   proc    near
                    push    bp
                    mov     bp,sp
                    push    [bp].DMA_direction      ; get direction
                    push    [bp].DMA_length_hi      ; ... length
                    push    [bp].DMA_length_lo      ; ...
                    push    [bp].DMA_data_ptr       ; ... and address
```

```
                call    Setup_DMA       ; set up the system's DMA
                                          controller
                mov     sp,bp           ; remove stack param's
                les     bx,SPC_Pointer  ; point to SPC
                mov     al,byte ptr [bp].DMA_length_hi ; MSW of
                                        transfer size
                mov     TCH,al          ; store in counter high
                mov     ax,[bp].DMA_length_lo ; LSW of transfer size
                mov     TCM,ah          ; store in counter mid
                mov     TCL,al          ; store in counter low
                mov     SCTL,080h       ; start DMA transfer
NoIntr:
                cmp     INTS,0          ; wait for intr, signaling
                                        ; ... the end of the DMA
                                          transfer
                 je     NoIntr          ; no intr yet
                test    INTS,010h       ; cmd complete intr?
                 je     NoCmdCmp        ; no, transfer error
                mov     ax,OK           ; transfer successful
                jmp     DMADone
NoCmdCmp:
                mov     ax,ERROR_INTR   ;      transfer didn't
complete                                        properly
                jmp     DMADone
DMADone:
                pop     bp
                ret
DMA_Transfer_Data   endp

;*************************************************************;
;                                                             ;
;*************************************************************;
```

Appendix 2

SPC DRIVER EXAMPLES IN C LANGUAGE

The following code is a group of routines that demonstrate the essential elements necessary for the implementation of a SCSI driver using Fujitsu's family of controllers. These routines make certain assumtions abou the hardware interface, but are software compatible with all the Fujitsu SCSI control. For this example the controller is memory mapped somewhere in the address space of the host CPU. This example assumes that the controller will be used in a polled environment (not interrupt driven) using programmed I/O (no DMA). Using the device in this mode requires much more code than an interrupt driven/DMA environment but the code will be less dependent on system hardware. At the end of this example is a re–write of the data transfer routine DMA transfers instead or programmed I/O. As can be seen the code is considerably smaller and simpler.
This code is intended to show the programming and operation of the major functions of the Fujitsu family of SCSI controllers. It is not intended as a tutorial of the SCSI protocol or to show the construction of a driver for an SCSI host adapter.
This code was tested for syntax errors under the following compilers:
Borland Turbo'C' 2.0

```
/*****************************************************************/
/*            For detailed information on the SCSI protocol      */
/*            see the spec entitled:                             */
/*                                                               */
/*            ANSI Small Computer System Interface (SCSI)        */
/*            ANSI X3.131-1986                                   */
/*                                                               */
/*            American National Standards Institute, Inc.        */
/*            1430 Broadway                                      */
/*            New York, NY  10018                                */
/*                                                               */
/*****************************************************************/
/*            For information on Fujistu's SCSI controllers      */
/*            contact:                                           */
/*                                                               */
/*            Fujitsu Microelectronics, Inc.                     */
/*            3545 North First Street                            */
/*            San Jose, CA.95134-1804                            */
/*            (408) 922-9000                                     */
/*                                                               */
/*            or your local Fujitsu representative.              */
/*                                                               */
/*****************************************************************/
/*                                                               */
/*            Copyright (C) 1989 Fujitsu Microelectronics Inc.   */
/*            Copyright (C) 1989 Galbo and Associates, Inc.      */
/*                                                               */
/*****************************************************************/
```

```
/*************************************************************************/
/*Constants                                                              */
/************************************************************************* /

/*selection time-out delays                                              */

#define SELECTION_TIMEOUT 4400L  /* 275 msec @ 8 MHz */
#define BUS_DELAY         4      /* 1200 nsec @ 8 MHz */

/*      data transfer direction */

#define READ              0      /* read from SCSI bus */
#define WRITE             1      /* write to SCSI bus */

/*      selection return values */

   #define SELECTED       100    /* target successfully selected*/
#define LOST_ARBITRATION  -100   /* selection failed */
#define TIMEOUT           -101   /* selection failed */
#define ERROR_INTR        -102   /* unexpected interrupt occurred */

/*      data transfer return values */

#define OK                200    /* transfer complete */
#define OKPAD             201    /* transfer complete with padding */
#define SERVICE_REQUIRED  -200   /* early phase change */
#define PARITY_ERROR      -201   /* read parity error */

/*      phase decoder return values */

#define NO_PHASE_REQUESTED      0
#define ILLEGAL_PHASE     -300   /* illegal SCSI phase requested */

#define DEFAULT_SPC_ADDR  0xdc001ff0L;
```

```
/*****************************************************************/
/*      All Fujitsu SCSI controllers contain the following 16    */
/*      8-bit registers. This structure assumes the compiler that can*/
/*      pack bytes within structures.                            */
/*****************************************************************/

structSPCREGPACK
{
volatile unsigned char
        _BDID,          /* bus device ID */
        _SCTL,          /* SPC control */
        _SCMD,          /* SPC command */
        _TMOD,          /* transfer mode */
        _INTS,          /* interrupt sense */
        _PSNS,          /* phase sense */
        _SSTS,          /* SPC status */
        _SERR,          /* SPC error status */
        _PCTL,          /* phase control */
        _MBC,           /* modified byte counter */
        _DREG,          /* FIFO data register */
        _TEMP,          /* temporary register */
        _TCH,           /* transfer counter high */
        _TCM,           /* transfer counter mid */
        _TCL,           /* transfer counter low */
        _EXBF;          /* external buffer */
} far *spc = (struct SPCREGPACK far *) DEFAULT_SPC_ADDR;

/*****************************************************************/
/*      Assume the Fujitsu SCSI Protocol Controller is accessed via  */
/*      memory mapped addressing, the following constants are used   */
/*      to read and write to the registers.                      */
/*****************************************************************/

#define BDID            ( spc->_BDID )
#define SCTL            ( spc->_SCTL )
#define SCMD            ( spc->_SCMD )
#define TMOD            ( spc->_TMOD )
#define INTS            ( spc->_INTS )
#define PSNS            ( spc->_PSNS )
#define SSTS            ( spc->_SSTS )
#define SERR            ( spc->_SERR )
#define PCTL            ( spc->_PCTL )
#define MBC             ( spc->_MBC  )
#define DREG            ( spc->_DREG )
#define TEMP            ( spc->_TEMP )
#define TCH             ( spc->_TCH  )
```

```
#define TCM                ( spc->_TCM  )
#define TCL                ( spc->_TCL  )
#define EXBF               ( spc->_EXBF )
#define SDGC               ( spc->_PSNS )     /* the SDGC reg has the */
                                             /* ... same addr as PSNS */

/*************************************************************************/
/*      Array to convert SCSI addresses to SCSI ID's (for arbitration*/
/*************************************************************************/

unsigned charAddr2ID[]   =  { 0x01, 0x02, 0x04, 0x08,
                            0x10, 0x20, 0x40, 0x80 };

/*************************************************************************/
/*      Variables needed for this example.                               */
/*************************************************************************/

unsigned char              msgin, msgout, status;
int                        cmdlength, datalength;
unsigned char              *cmdptr,*dataptr;

extern void                setup_dma(unsigned char *, unsigned long, int);

/*************************************************************************/
/*      function prototypes                                              */
/*************************************************************************/

void    init_SPC(int);
int     select(int, int);
int     transfer_data(unsigned char *, unsigned long, int);
int     send_command(unsigned char *, int);
int     status_in(unsigned char *);
int     message_in(unsigned char *);
int     message_out(unsigned char *);
int     data_in(unsigned char *, unsigned long);
int     data_out(unsigned char *, unsigned long);
int     decode_phase(void);
int     manual_transfer_data(unsigned char *, unsigned long, int);
int     dma_transfer_data(unsigned char *, unsigned long, int);
```

```
/*********************************************************************/
/*      INITIALIZE SPC                                               */
/*      Initialize SCSI controller.                                  */
/*                                                                   */
/*      Entry:init_addr - SCSI address of initiator,0..7             */
/*      Exit:void                                                    */
/*********************************************************************/
void    init_SPC(init_addr)
int     init_addr;          /* this initiator's SCSI address, 0..7 */
{
  SCTL = 0xc0;              /* reset controller */
  BDID = init_addr;
  TMOD = 0x00;
/*                                                                   */
/*      Release reset with arbitration on, parity checking on,       */
/*      reselection on (INITIATOR), hardware interrupts disabled.    */
/*                                                                   */
  SCTL = 0x1a;
  SDGC = 0x00;              /* no xfer intr for 89352
                */
}

/*********************************************************************/
/*SELECT TARGET*/
/*      Arbitrate for the SCSI bus and select a target device.       */
/*                                  */
/*      Entry:    targ_addr - Target's SCSI bus address, 0..7        */
/*                targ_lun - Target's LUN (logical unit number)      */
/*      Exit:     > 0 if successful selection                        */
/*                < 0 if unsuccessful                                */
/*********************************************************************/
int     select(targ_addr, targ_lun)
int     targ_addr;          /* the selected target's SCSI address, 0..7  */
int     targ_lun;           /* the selected target's logical unit number */
{
  PCTL = 0x00;              /* phase control reg indicates selection phase*/
/*                                                                   */
/*      Set the TEMP register to the target's SCSI ID (not address)  */
/*      OR'ed with the initiator's ID (in the BDID register). This   */
/*      is done so that the target knows the ID of the initiator for */
/*      reselection purposes.                                        */
/*                                                                   */
  TEMP = BDID | Addr2ID[targ_addr];
/*                                                                   */
/*      The transfer counter registers double as selection time-out  */
```

```
/*      registers during arb/selection.  Set these registers to the  */
/*      desired value.The time-out values are dependent upon the     */
/*      SPC's clock speed.                                           */
/*                                                                   */
  TCH = SELECTION_TIMEOUT >> 8; /* high byte of selection time-out   */
  TCM = SELECTION_TIMEOUT & 0xff; /* low byte of selection time-out  */
  TCL = BUS_DELAY;          /* bus settle and bus free delay        */
/*                                                                   */
/*      Assert the SCSI attention signal during the selection phase, */
/*      this will notify the target that the initiator wants to send */
/*      an IDENTIFY message after successful selection.  The ATN     */
/*      signal will not be activated until AFTER the selection       */
/*      phase has successfully completed.                            */
/*                                                                   */
  SCMD = 0x60;              /* assert ATN signal                    */
/*                                                                   */
/*      Start the arbitration and selection sequence.                */
/*                                                                   */
  SCMD = 0x10;              /* send select command                  */
/*                                                                   */
/*      Wait for the interrupt register to signal the end of the     */
/*      selection phase or check if the SPC lost the bus arbitration */
/*      phase.                                                       */
/*                                                                   */
   while (! INTS)             /* wait for interrupt indicated */
   {                          /* ... if no interrupt, check for lost arb*/
    if (! (SSTS & 0x20))      /* ... test if arb/selection still in
                                 progress */
    {                         /* ... arb/selection finished, no
                                 intr pending */
       return (LOST_ARBITRATION);  /* indicate unsuccessful selection*/
    }
   }
/*                                                                   */
/*      An interrupt occurred, check if it's the command complete    */
/*      interrupt, indicating successful arb/selection.              */
/*                                                                   */
   if (INTS & 0x10)           /* command complete intr bit set ? */
   {
/*                                                                   */
/*      Command complete interrupt indicated.                        */
/*      Arb/Selection was successful, check if target is requesting  */
/*      a message out, if so, send the IDENTIFY message with the     */
/*      disconnection and reselection supported bits set and an      */
/*      initiator LUN of 0.                                          */
/*                                                                   */
```

```
    if ((PSNS & 0x87) == (0x86))   if target requesting msg out      */
    {                              /* ... phase after selection      */
     msgout = 0xc0 | targ_lun;     /* support disconnect/reselect */
     message_out(&msgout);         /* send message */
    }                              /* ... and select the target LUN */
   }
   else                            /* no command complete intr */
   {
/*                                                                   */
/*     An interrupt occurred but it was not command complete,       */
/*     indicating that the arb/selection was not successful.        */
/*                                                                   */
    if (INTS & 0x04)               /* time-out interrupt ? */
    {
     return (TIMEOUT);             /* indicate time-out */
    }
    else                           /* not time-out intr */
    {
     return (ERROR_INTR);          /* must be a bus reset, disconnect or   */
    }                              /* ... or SPC error intr                */
   }
    return (SELECTED);             /* indicate successful arb/selection */
   }                               /* ... we are connected to the target   */

/*********************************************************************/
/*   TRANSFER DATA TO/FROM SCSI BUS                                  */
/*   Read or write a stream of bytes to the SCSI bus and             */
/*   check for the following conditions:                             */
/*                                                                   */
/*   1) # of bytes requested == # of bytes transferred - normal      */
/*   2) # of bytes requested > # of bytes transferred - pad          */
/*   3) # of bytes requested < # of bytes transferred - phase change */
/*                                                                   */
/*   Entry:      dp - pointer to list of bytes                       */
/*               length - number of bytes in list                    */
/*               transfer type - READ or WRITE                       */
/*   Exit:       > 0 - transferred all bytes in list                 */
/*               < 0 - transfer terminated early                     */
/*********************************************************************/

int     transfer_data(dp, Length, io)
unsigned char             *dp;
unsigned long             Length;
int                       io;    /* transfer type READ or WRITE */
{
   TCH = length >> 16;            /* high byte of transfer count*/
   TCM = (length >> 8) & 0xff;      /* middle byte          */
```

```
   TCL = length & 0xff;          /* low byte                */
/*                                                          */
/*      Start SCSI programmed I/O transfer                  */
/*                                                          */
   SCMD = 0x84;
/*                                                          */
/*      Wait for SPC to start processing the command        */
/*                                                          */
   while ((SSTS & 0xf0) != 0xb0) ;
/*                                                          */
/*      Wait for an available location in the output FIFO, then   */
    send byte.                                              */
/*                                                          */
   while (length)                /* while not finished sending */
   {
    if (! INTS)                  /* send while no interrupts pending */
    {
/*                                                          */
/*      Read or write a byte to the SPC's FIFO                 */
/*                                                          */
     if (io == WRITE)
     {
/*                                                          */
/*      Write a byte to the FIFO if the FIFO is NOT FULL.   */
/*                                                          */
       if (! (SSTS & 0x02))           /* if space in FIFO */
       {
         DREG = *dp++;             /* write byte to FIFO, bump ptr */
        length--;                  /* decrement byte count */
       }
     }
     else
     {
/*                                                          */
/*      Read a byte from the FIFO if the FIFO is NOT EMPTY. */
/*                                                          */
       if (! (SSTS & 0x01))           /* if space in FIFO */
       {
         *dp++ = DREG;             /* read byte to FIFO, bump ptr */
         length--;               /* decrement byte count */
       }
     }
    }
    else                         /* an interrupt is pending */
    {                            /* ... check type */
     if (INTS & 0x08)            /* if SVC REQ intr */
```

```
      {
        INTS = 0x08;              /* reset service required */
        return (SERVICE_REQUIRED); /* exit early */
      }
      else if (INTS)              /* if other intr */
      {
        return (ERROR_INTR);      /* bus release, reset, SPC error intr*/
      }
      else if (SERR & 0xc0)       /* read parity error ? */
      {
        return (PARITY_ERROR);
      }
    }                             /* ... process later */
    }
/*                                                                    */
/*     All bytes (length) in data stream have been transferred.       */
/*     Check for command complete interrupt after all data sent.      */
/*                                                                    */
   while (! (INTS & 0x10)) ;      /* wait for cmd complete */
   INTS = 0x10;                   /* reset cmd complete intr */
/*                                                                    */
/*     If target has not changed phase after sending all bytes        */
/*     then we should pad the transfer to let the target continue     */
/*     to the next phase.                                             */
/*                                                                    */

   if (((PCTL & 0x07) | 0x80) == (PSNS & 0x87)) /* new phase ? */
   {                                            /* same phase */
     TEMP = 0;       /* do pad transfer, pad with 0's for write */
     SCMD = 0x85;    /* start programmed transfer PAD out */
/*                                                                    */
/*     The completion of the padding transfer is indicated by the     */
/*     command complete intr and the service required intr (since     */
/*      the target will normally change phase at the end of the       */
/*     transfer)                                                      */
/*                                                                    */
     while ((INTS & 0x18) != 0x18) ; /* wait for pad to complete */
     INTS = 0x18;          /* reset intr's */
     return (OKPAD);
   }
   else                    /* target has gone to next phase */
    return (OK);           /* transfer completed normally */
}
```

```
/*********************************************************************/
/*      SEND COMMAND                                                 */
/*      Send a SCSI command to the target                            */
/*      Assume: TCH, TCM, TCL regs = 0                               */
/*                                                                   */
/*      Entry:    cp - pointer to SCSI command stream                */
/*                length - length of the command, 6, 10, or 12 bytes */
/*      Exit:     > 0 - transferred all bytes in list                */
/*                < 0 - transfer terminated early                    */
/*********************************************************************/

int     send_command(cp, length)
unsigned char       *cp;        /* pointer to command stream */
int                 length;     /* length of command stream */
{
   PCTL = 0x02;           /* indicate command phase */
/*                                                                   */
/*       Start transfer                                              */
/*                                                                   */
   return (transfer_data(cp, length, READ));
}

/*********************************************************************/
/*      READ STATUS IN                                               */
/*      Read the 1 byte status code from the target.                 */
/*                                                                   */
/*      Entry:    sp - pointer to status byte                        */
/*      Exit:     > 0 successful read                                */
/*                < 0 unsuccessful read                              */
/*********************************************************************/

int     status_in(sp)
unsigned char*sp;               /* pointer to status byte */
{
   PCTL = 0x03;                 /* indicate status phase */
   return (transfer_data(sp, 1, READ));
}
```

```
/*********************************************************************/
/*      SEND MESSAGE OUT                                             */
/*      Send a 1 byte message to the target.                         */
/*      When the SPC is in the INITIATOR mode, it will automatically */
/*      negate the ATN signal after the message byte has been        */
/*      successfully sent to the target device, this is in comformance*/
/*      with the SCSI spec. (See spec. for further details)          */
/*                                                                   */
/*      Entry:    mp - pointer to message byte                       */
/*      Exit:     > 0 successful read                                */
/*                < 0 unsuccessful read                              */
/*********************************************************************/

int      message_out(mp)
unsigned char       *mp;
{
   PCTL = 0x06;                    /* indicate message phase */
   return (transfer_data(mp, 1, WRITE));
}

/*********************************************************************/
/*      READ MESSAGE IN                                              */
/*      Read a 1 byte message from the target                        */
/*      When the SPC is in the INITIATOR mode it will automatically  */
/*      hold the ACK line active on the last byte of the message     */
/*      received from the target device. This is to allow the        */
/*      INIATOR to interpret the message and accept or, if           */
/*      necessary, reject it according to the SCSI protocol's        */
/*      message reject sequence. (See spec. for further details)     */
/*      reject it according to the SCSI protocol's message reject    */
/*      sequence. (See spec. for further details)                    */
/*                                                                   */
/*      Entry:    mp - pointer to message byte                       */
/*      Exit:     > 0 successful read                                */
/*                < 0 unsuccessful read                              */
/*********************************************************************/

int     message_in(mp)
unsigned char*mp;
{
   int  rv;

   PCTL = 0x06;                    /* indicate message phase */
   rv = transfer_data(mp, 1, WRITE);  /* save return value */
   if (rv > 0)                     /* if message read successfully */
   {
     SCMD = 0xc0;                  /* manually negate the ACK signal */
     INTS = 0x10;                  /* reset cmd complete intr */
```

```
   }
   return (rv);
}

/*********************************************************************/
/*      READ SCSI DATA                                               */
/*      Read up to 256 MBytes from the target device.                */
/*                                                                   */
/*      Entry:mp - pointer to byte stream                            */
/*      length - number of bytes to read                             */
/*********************************************************************/

int     data_in(mp, length)
unsigned char        *mp;
unsigned long        length;
{
   PCTL = 0x01;            /* indicate data in phase */
   return (transfer_data(mp, length, READ));
}

/*********************************************************************/
/*      WRITE SCSI DATA                                              */
/*      Write up to 256 MBytes to the target device.                 */
/*                                                                   */
/*      Entry:    mp - pointer to byte stream                        */
/*                length - number of bytes to write                  */
/*********************************************************************/

int data_out(mp,length)
unsigned char        *mp;
unsigned long        length;
{
   PCTL = 0x00;            /* indicate data out phase */
   return (transfer_data(mp, length, WRITE));
}
```

```
/*********************************************************************/
/*      DECODE SCSI BUS PHASE                                        */
/*      After selection, read target's requested SCSI bus phase      */
/*      and go to the appropriate handler, continue calling this     */
/*      routine until the COMMAND COMPLETE message is sent from the  */
/*      target.                                                      */
/*                                                                   */
/*      Exit:     > 0 successful operation                           */
/*                = 0 no phase requested (REQ signal not active)     */
/*                < 0 unsuccessful operation                         */
/*********************************************************************/

int      decode_phase()
{
   switch (PSNS & 0x87)              /* read phase sense reg */
   {                                 /* REQ bit (0x80) must be active */
    case 0x80:
     return (data_out(dataptr, datalength));
    case 0x81:
     return (data_in(dataptr, datalength));
    case 0x82:
     return (send_command(cmdptr, cmdlength));
    case 0x83:
     return (status_in(&status));
    case 0x84:                      /* undefined SCSI phases */
    case 0x85:                     /* ... should never see these */
     return (ILLEGAL_PHASE);
    case 0x86:
     return (message_out(&msgout));
    case 0x87:
     return (message_in(&msgin));
    default:
     return (NO_PHASE_REQUESTED);  /* REQ bit is not active */
   }
}
```

```
/*************************************************************************/
/*    MANUALLY TRANSFER DATA TO/FROM SCSI BUS                            */
/*    Read or write a stream of bytes to/from the SCSI bus and           */
/*    check for completion.                                              */
/*                                                                       */
/*    This example illustrates the use of the manual transfer mode       */
/*    of the SPC.  This mode is useful for "spoon-feeding" each      */
/*    byte to a target device, this could be helpful when debugging      */
/*    SCSI software step-by-step or testing the behavior of a SCSI       */
/*    peripheral under certain conditions.                               */
/*                                                                       */
/*    This mode of operation does not use the transfer counter of        */
/*    the SPC.  The transfer size is limited only by the size of         */
/*    the transfer length parameter passed to this routine.              */
/*                                                                       */
/*    Entry:dp - pointer to list of bytes                                */
/*    length - number of bytes in list                                   */
/*    type - READ or WRITE                                               */
/*    Exit:      > 0 - transferred all bytes in list                     */
/*               < 0 - transfer terminated early                         */
/*************************************************************************/

int      manual_transfer_data(dp, length, type)
unsigned char        *dp;
unsigned long        length;
intt                 ype;
{
   do
   {
    if (type == WRITE)
   {                              /* WRITE to SCSI */
     while (!(PSNS & 0x80)) ;/* wait for REQ asserted, indicating */
                              /* ... target ready for byte */
     TEMP = *dp++;            /* write byte */
     SCMD = 0xe0;             /* assert ACK, indicating byte on bus */
     while (PSNS & 0x80) ;    /* wait for REQ negated, indicating  */
                              /* ... target received byte */
     SCMD = 0xc0;             /* negate ACK, indicating operation
                                 complete                          */
   }
   else
   {                       /* READ from SCSI */
    while (!(PSNS & 0x80)) ;/* wait for REQ asserted, indicating */
                           /* ... target placed byte on bus */
    SCMD = 0xe0;           /* assert ACK, indicating INIT accepted byte */
    while (PSNS & 0x80) ; /* wait for REQ negated, indicating */
                           /* ... target finished driving bus */
```

```
    *dp++ = TEMP;          /* read byte */
    SCMD = 0xc0;           /* negate ACK, indicating operation complete */
   }
} while (--length);        /* decrement/test length */
return (OK);               /* exit */
}
/*********************************************************************/
/*   DMA TRANSFER DATA TO/FROM SCSI                                  */
/*   Read or write a stream of bytes to the SCSI bus using DMA       */
/*   and check for the following conditions:                         */
/*                                                                   */
/*   1) # of bytes transferred == # of bytes requested - normal      */
/*   2) # of bytes transferred > # of bytes requested - pad          */
/*   3) # of bytes transferred < # of bytes requested - phase change */
/*                                                                   */
/*   Entry:dp - pointer to list of bytes                             */
/*   length - number of bytes in list                                */
/*   type - READ or WRITE                                            */
/*   Exit:       > 0 - transferred all bytes in list                 */
/*               < 0 - transfer terminated early                     */
/*                                                                   */
/*   Note:       If the SPC is interrupt driven along with using DMA,*/
/*               the code below becomes even smaller since it is not */
/*               necessary to test the INTS register within a loop.  */
/*               In fact, the overhead to start a data transfer in   */
/*               the SPC takes only 3 steps:                         */
/*                                                                   */
/*               1) Write desired phase to PCTL register             */
/*               2) Write 1 to 3 bytes to the transfer counters,     */
/*                  TCH/M/L                                          */
/*               3) Write the DMA transfer cmd to SCMD register      */
/*                   (0x80)                                          */
/*********************************************************************/

int      dma_transfer_data(dp, length, type)
unsigned char       *dp;
unsigned long       length;
int                 type;          /* transfer type READ or WRITE */
{
   setup_dma(dp, length, type);     /* set up the DMA controller */
   TCH = length >> 16;              /* high byte of transfer count*/
   TCM = (length >> 8) & 0xff;                /* middle byte*/
   TCL = length & 0xff;                          /* low byte*/
   SCMD = 0x80;                     /* start DMA transfer */
   while (! INTS) ;                 /* wait for an interrupt */
   if (INTS & 0x10)                 /* if cmd complete intr */
```

```
    return (OK);                    /* ... done */
   return (SERVICE_REQUIRED);       /* handle other interrupt later */
}
/*************************************************************************/
/*                                                                       */
/*************************************************************************/
```

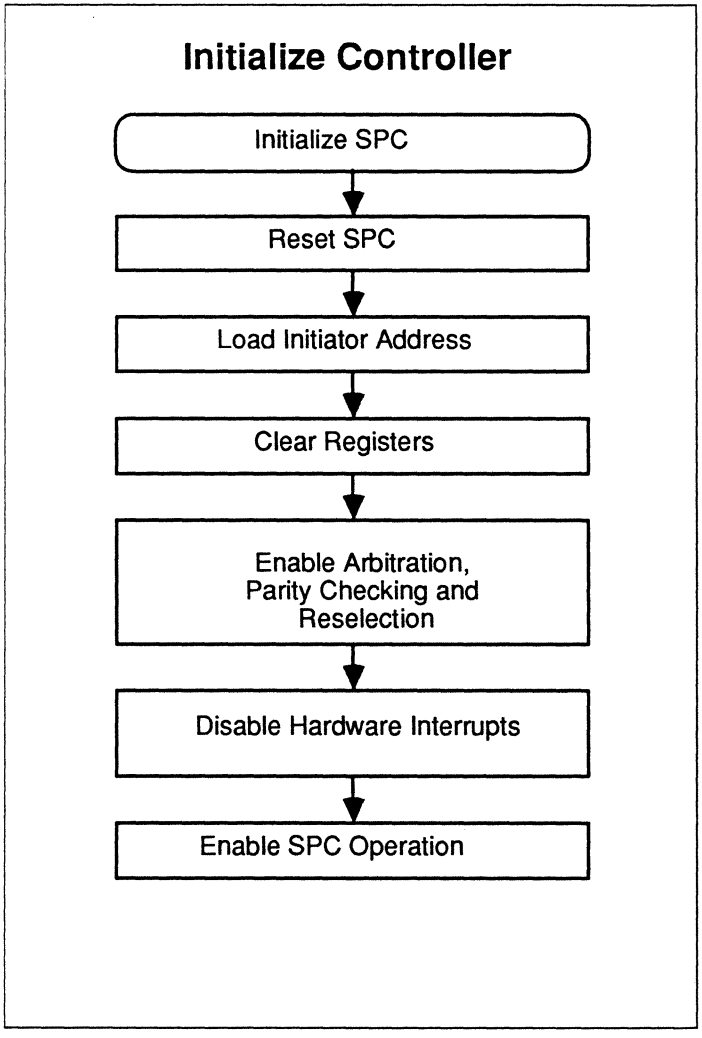
Appendix 3

CONTROL SEQUENCE FOR SPC DRIVER

The following examples show the essential elements necessary for the creation of a SCSI software driver using Fujitsu's family of SCSI controllers. Flowcharts are provided to outline the overall sequence of operation. Code that is dependent on the operating environment such as error processing, exception handling, and the operating system interface are not shown for the sake of brevity and clarity; This example assumes that the SCSI controller is accessible via memory-mapped I/O and is controlled by an 8086 family processor. This code will operate on any of the Fujitsu SCSI controllers.

Initialize Controller

```
INITADDR  EQU    7

INITIALIZE:
      MOV   SCTL, OCOH        ;  Reset controller
      MOV   BDID, INITADDR    ;  Load INIT's SCSI addr, 0..7
      MOV   TMOD, 0           ;  ASYNC I/O
      MOV   SDGC, 0           ;  Diagnostic mode off
      MOV SCTL, 01AH          ;  Relese reset with Arbitration on,
      RET                     ;  Parity Check on, Reselect on,
                              ;  H/W Interrupt off
```


Initialize Controller
Initialize SPC
Reset SPC
Load Initiator Address
Clear Registers
Enable Arbitration, Parity Checking and Reselection
Disable Hardware Interrupts
Enable SPC Operation

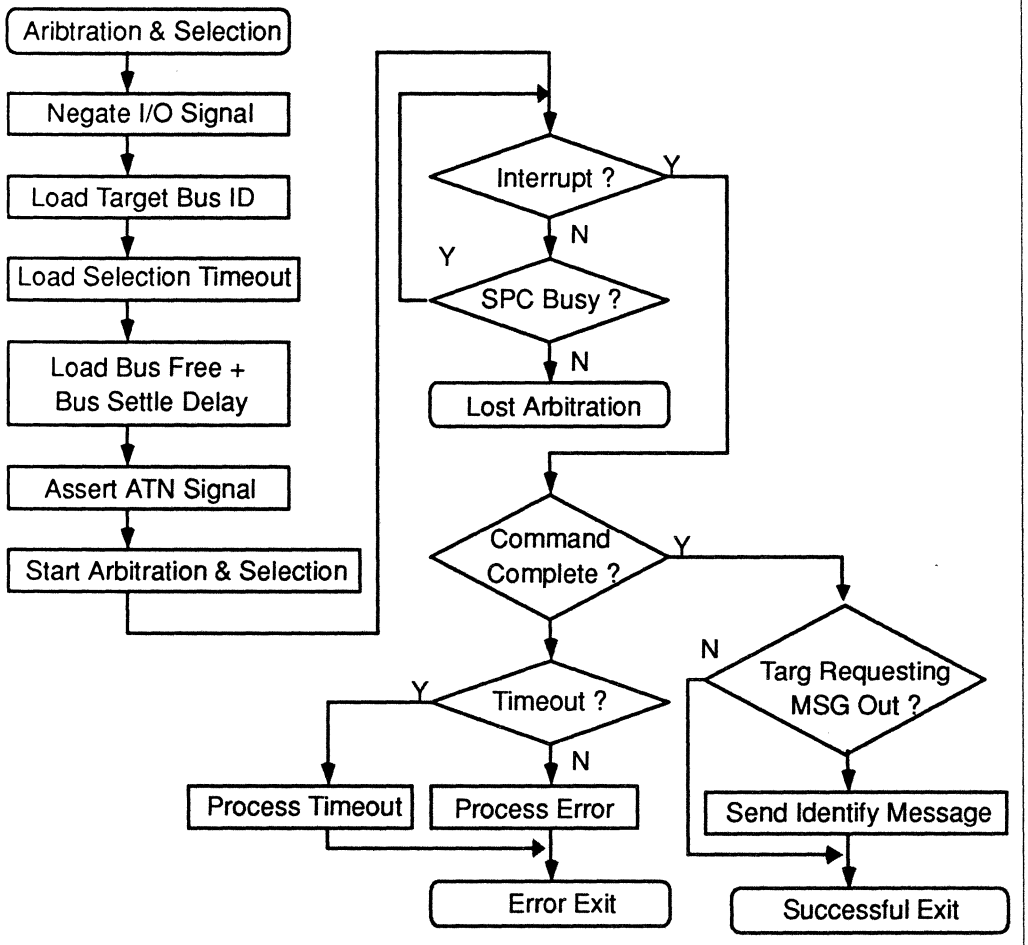
Arbitrate and Select Phase

```
; ENTRY:      BL = TARGET BUS ID (1,2,4,8,16,32,64,128)
;             BH = TARGET LUN
;

STD    EQU    4400              ; 275 msec
BFD    EQU    4                 ; 1200 nsec

       MOV    PCTL,00H          ; Indicate selection phase,I/O = 0
       OR     BL,BDID           ; 'OR' in Initiator Bus ID
                                ; ... So Target can reselect
       MOV    TEMP,BL           ; Set Target & Init bus ID
                                ; ... during selection
       MOV    TCH,HIGH(STD)     ; Set MSB of sel timeout value
       MOV    TCH,LOW(STD)      ; Set LSB of sel timeout value
       MOV    TCL,BFD           ; Set bus settle + bus free delay
       MOV    SCMD,60H          ; Assert ATN signal
       MOV    SCMD,10H          ; Start ARB & Selection sequence
WAIT:
       CMP    INTS,0            ; Intr occur ?
       JNZ    INTR              ; Yes
       TEST   SSTS,20H          ; Operation in progress ?
       JNZ    WAIT              ; Yes
LOSTARB:                        ; No, lost Arbitration
 ... process lost arbitration
```

Arbitrate and Select Phase (continued)

```
INTR:
      TEST    INTS,10H        ; Selection successful ?
      JNZ     CONNECTED       ; Yes, continue
      TEST    INTS,04H        ; Target, timeout error ?
      JNZ     TIMEOUT         ; Yes, process
      JMP     OTHER           ; Process error
CONNECTED:
      CALL    CHECKMSGOUT     ; Check for MSG OUT Phase in order
                              ; ... to send Identify MSG
      JC      NOIDENT
      MOV     AH,BH           ; Load desired Target LUN
      OR      AH,0C0H         ; Send Identify, with support for
                              ; ... Disconnection & Reselection
      CALL    MSGOUT          ; Send message
NOIDENT:
      RET                     ; Target successfully selected

OTHER:
 ... process unexpected condition
 ... bus reset or SPC hardware error
```

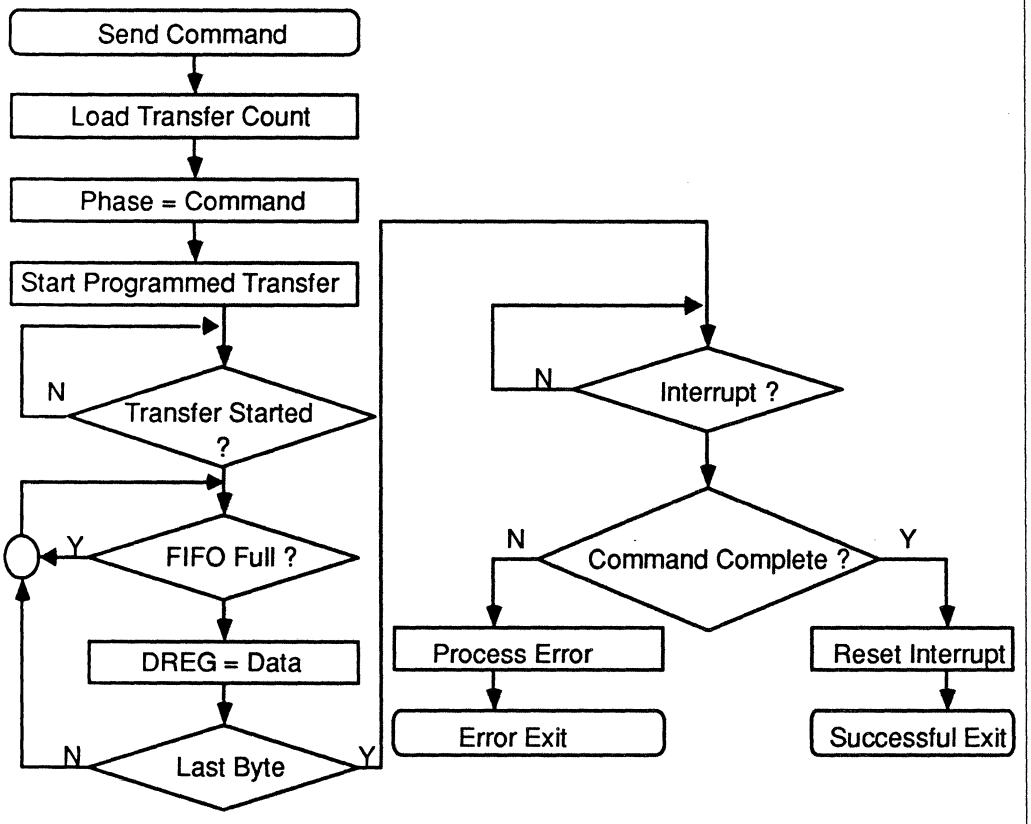
Arbitrate and Select Phase



Command Phase

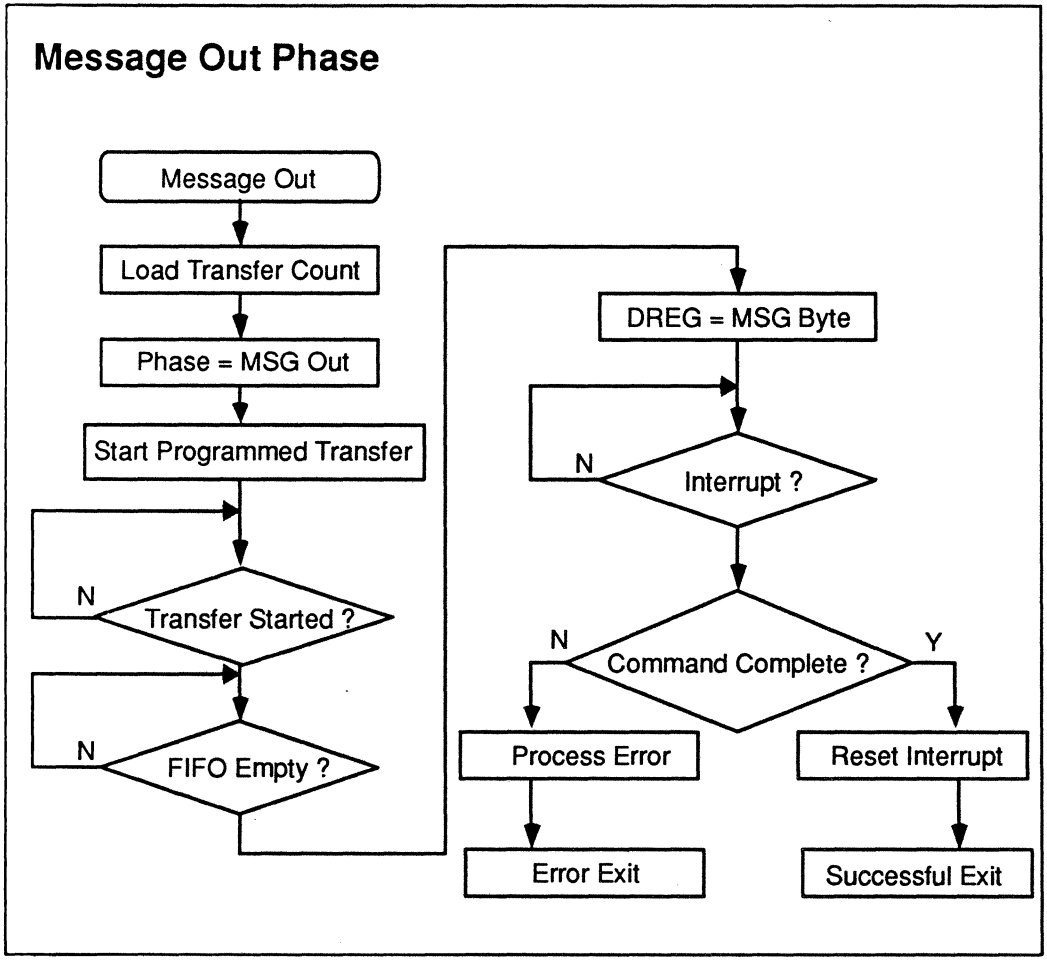
```
; ENTRY:        [ES:SE] ?> SCSI Command
;               CL = Command Length - 6,10 or 12
;               TCH, TCM, TCL = 0

SENDCOMMAND:  MOV   TCL,CL          ; Load transfer count
              MOV   PCTL,02H        ; Command Phase
              MOV   SCMD,84H        ; Program transfer
              MOV   AL,0F0H         ; Wait for SPC command to begin
SYNC:         AND   AL,SSTS         ; Check Init & Busy bit
              CMP   AL,0B0H         ; Ready ?
              JNE   SYNC            ; No, Wait
              TEST  SSTS,02H        ; FIFO Full?
              JNE   COUT            ; Yes, wait
COUT:         LODS  BYTE PTR ES:[SI]; Read byte, inc ptr
              MOV   DREG,AL         ; Send byte to SCSI
              DEC   CL              ; Count done?
              JNZ   COUT            ; No
              CMP   INTS,0          ; Intr Occur ?
              JZ    CWAIT           ; No, wait
              TEST  INTS,10H        ; Command completed ?
CWAIT:        JZ    CERR            ; No, process error
              MOV   INTS,10H        ; Reset CMD complete Intr
              RET                   ; Finished



CERR:
 ...process unexpected condition
    ...bus reset, disconnect or SPC hardware error
```

Command Phase



Message Out Phase

```
; ENTRY:      AH = Message Byte
;             TCH, TCM, TCL, = 0
;             ATN Signal Is Asserted

MSGOUT:
      MOV     TCL,1               ; Transfer 1 byte
      MOV     PCTL,06H            ; Specify MSG OUT phase
      MOV     SCMD,84H            ; Program transfer
SYNC:
      MOV     AL,0F1H             ; Wait for SPC command to begin
      AND     AL,SSTS             ; Check Init & Busy bit
      CMP     AL,0B1H             ; Ready and FIFO empty ?
      JNE     SYNC                ; No, wait
      MOV     DREG,AH             ; Send byte & automatically
                                  ; ... negate ATN signal
MOWAIT:
      CMP     INTS,0              ; Intr ?
      JZ      MOWAIT              ; Not yet
      TEST    INTS,10H            ; Command completed ?
      JZ      MERR                ; No, process error
      MOV     INTS,10H            ; Reset CMD complete Intr
      RET                         ; Finished
MERR:
 ...process unexpected condition
 ...bus reset, disconnect or SPC hardware error
```


Message Out Phase
Message Out
Load Transfer Count
Phase = MSG Out
Start Programmed Transfer
N
Transfer Started ?
N
FIFO Empty ?
DREG = MSG Byte
N
Interrupt ?
N
Command Complete ?
Y
Process Error
Reset Interrupt
Error Exit
Successful Exit

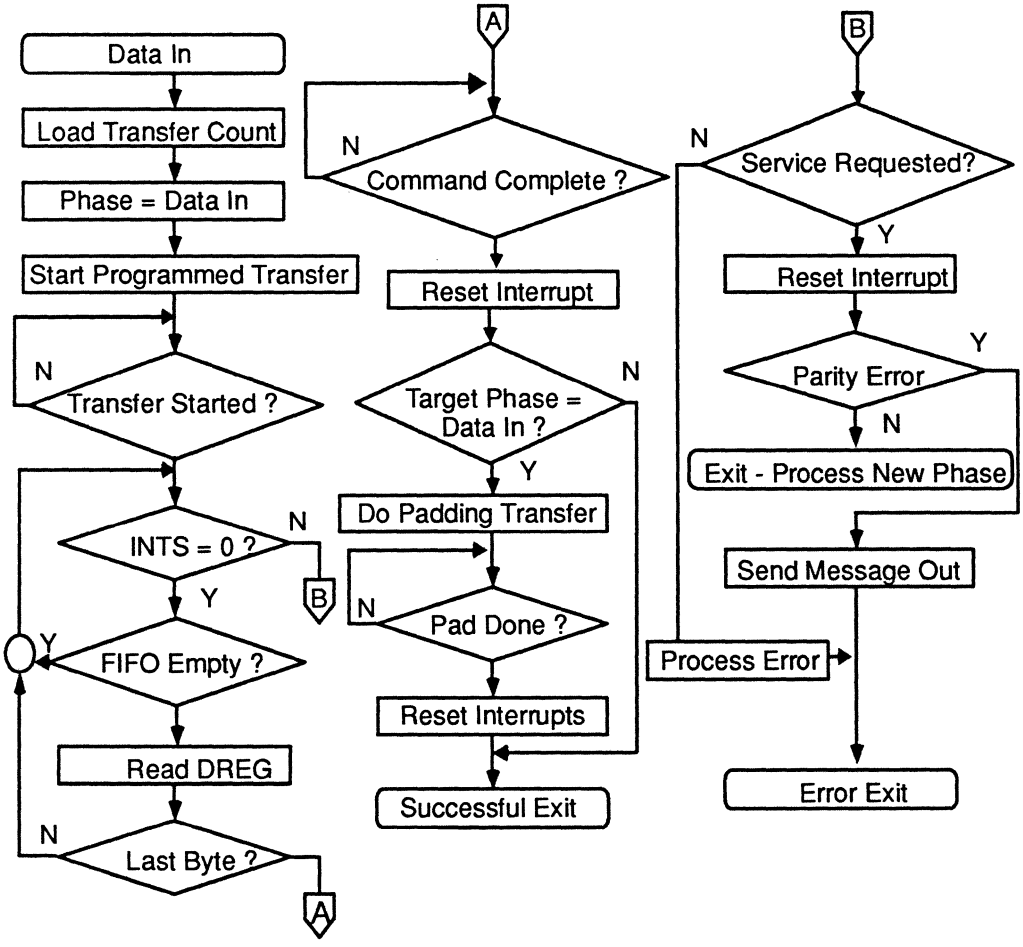
Data In Phase

```
; ENTRY:                BX = Byte Count
;                       [ES:DI] = Data Buffer Pointer
; Must handle 3 conditions:
; 1) Target Xfers exact byte count requested
; 2) Target Xfers more bytes than requested (Pad Input)
; 3) Target Xfers less bytes than requested (Phase Change)

DATAIN:
    CLD                         ; Clear direction flag
    MOV     TCM,BH              ; Load count, 0..65535
    MOV     TCL,BL
    MOV     PCTL,01H            ; Data In Phase
    MOV     SCMD,84H            ; Programmed transfer
SYNC:
    MOV     AL,0F0H             ; Wait for SPC command to begin
    AND     AL,SSTS             ; Check Init & Busy bit
    CMP     AL,0B0H             ; Ready ?
    JNE     SYNC                ; No, wait
DILOOP:
    CMP     INTS,0              ; Any Intrs ?
    JNE     DISVC               ; Yes, process
    TEST    SSTS,01H            ; FIFO empty ?
    JNZ     DILOOP              ; Yes, wait
    MOV     AL,DREG             ; Read byte
    STOS    BYTE PTR [ES:DI]    ; Store & inc ptr
    DEC     BX                  ; Done ?
    JNE     DILOOP
```

Data In Phase (continued)

```
DICC:
      TEST    INTS,10H
      JZ      DICC
      MOV     INTS,10H; SPC Command complete ?
      MOV     AL,87H  ; No, wait
      AND     AL,PSNS ; Reset Intr
      CMP     AL,81H  ; Target still in Data-In phase ?
      JNE     DIEXIT  ; Read phase sense register
DIPAD:                ; Req + Data-In Phase ?
      MOV     SCMD,85H; No, exit normally (exact transfer)
DICCSR:               ; Target has more, pad input
      TEST    INTS,18H; Programmed xfer pad (long xfer)
      JZ      DICCSR
      MOV     INTS,18H; Command complete + service required intr ?
DIEXIT:               ; No, wait for pad to finish
      RET             ; Reset Intr's
DISVC:
      TEST    INTS,08H; Done
      JZ      DIERROR
      MOV     INTS,08H; Service req Intr (Phase Change) ?
      TEST    SERR,0C0H No, process error Intr
      JNZ     DIERROR ; Yes, xfer halted (short xfer)
 ... process phase change Parity error ?
                      ; Yes
DIERROR:              ; No, short transfer
 ...process unexpected condition
 ...bus reset, disconnect, parity or SPC
hardware error
```

Data In Phase



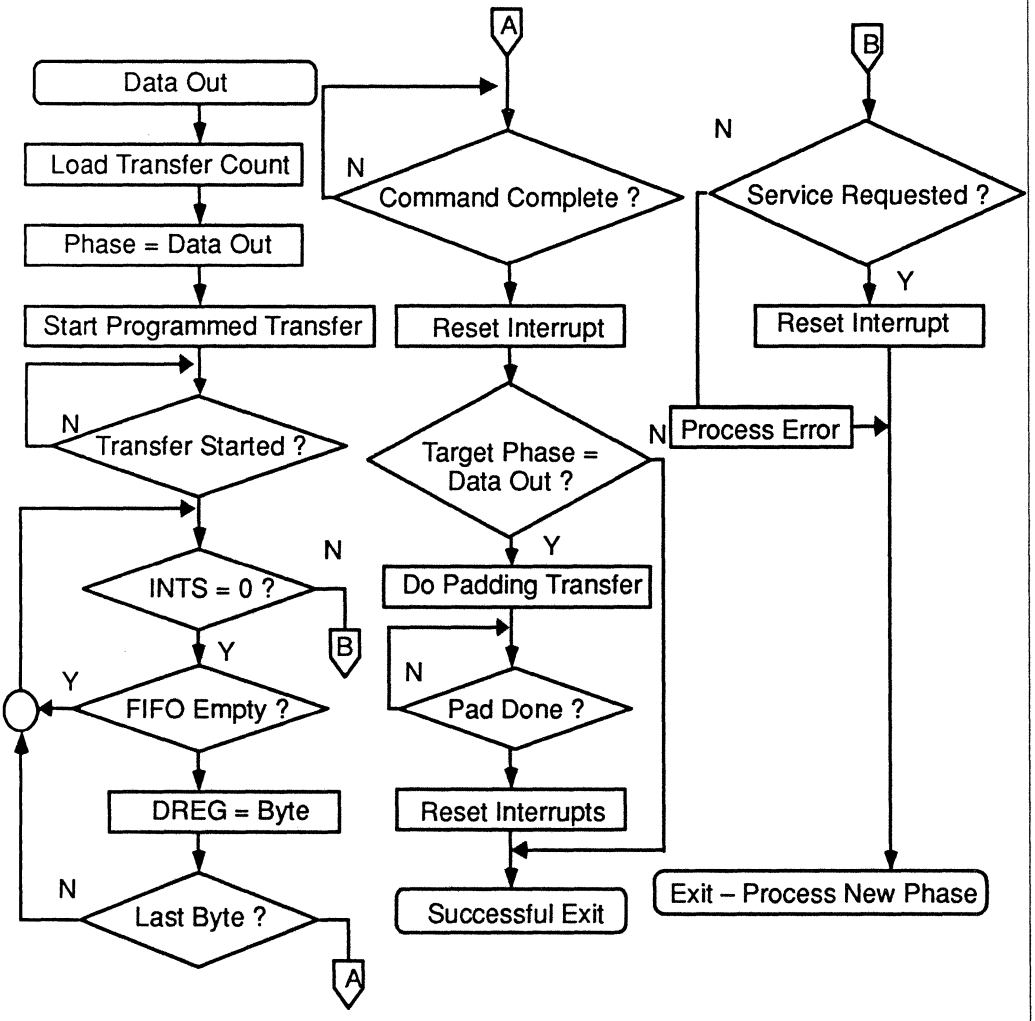
Data Out Phase

```
; ENTRY:      BX = Transfer Count
;             [ES:SI] = Data Buffer Address

DMADATAOUT:
      CALL    SETDMA              ; Set up DMA controller
      MOV     TCM,BH              ; Load transfer count
      MOV     TCL,BL
      MOV     PCTL,00H            ; Set Data Out phase
      MOV     SCMD,80H            ; DMA transfer
DMAOUTWAIT:
      CMP     INTS,0              ; Wait for DMA to complete
      JE      DMAOUTWAI
      TEST    T                   ; Command complete Intr ?
      JZ      INTS,10H            ; Process unexpected interrupt
      MOV     PUI                 ; Reset Intr
      RET     INTS,10H            ; Exit
```

Data Out Phase (continued)

```
DOCC:
      TEST    INTS,10H  ; SPC Command complete ?
      JZ      DOCC      ; No, wait
      MOV     INTS,10H  ; Reset Intr
      MOV     AL,87H    ; Target still in Data-Out phase ?
      AND     AL,PSNS   ; Read phase sense register
      CMP     AL,80H    ; Req + Data-In Phase ?
      JNE     DOEXIT    ; No, exit normally (exact transfer)
DOPAD:                  ; Target has more, pad output
      MOV     TEMP,0    ; Send 0's
      MOV     SCMD,85H  ; Programmed transfer pad
DOCCSR:
      TEST    INTS,18H  ; Command complete + service required Intr ?
      JZ      DOCCSR    ; No, wait for pad to finish
      MOV     INTS,18H  ; Reset Intr's
DOEXIT:
      RET               ; Done
DOSVC:
      TEST    INTS,08H  ; Service Req Intr (phase change) ?
      JZ      DOERROR   ; No, process error Intr
      MOV     INTS,08H  ; Yes, xfer halted (short xfer)
 ...process phase change
DOERROR:
 ...process unexpected condition
 ...bus reset, disconnect or SPC hardware error
```

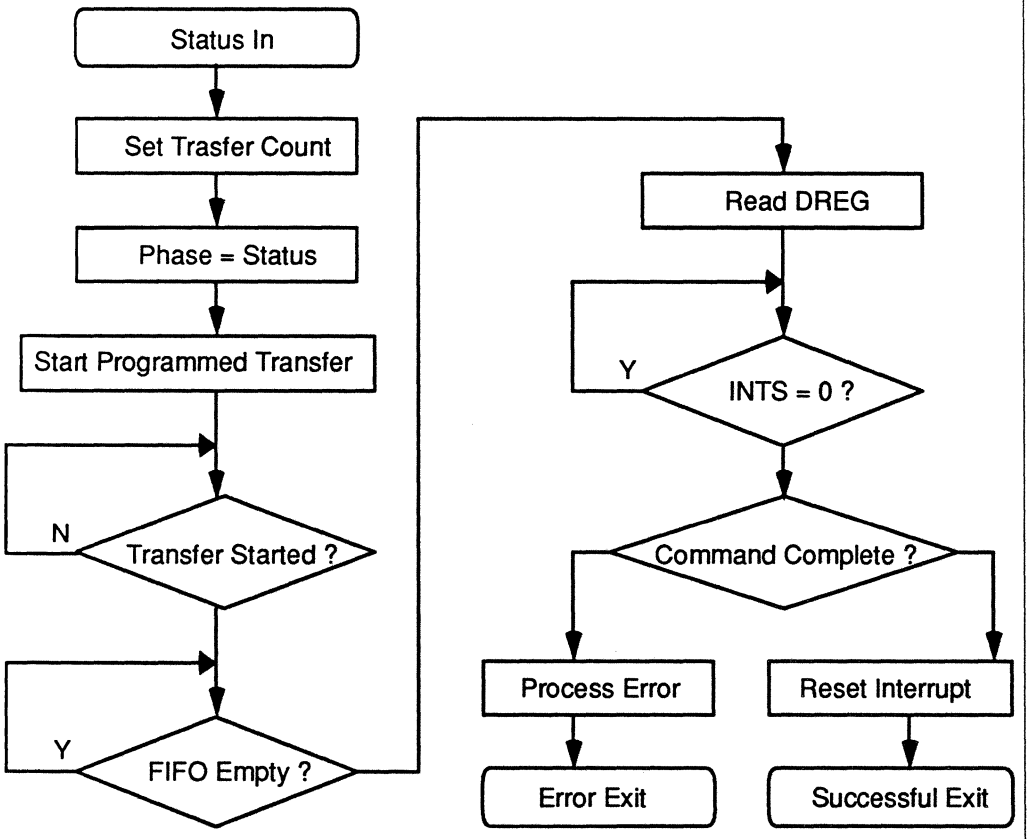
Data Out Phase



Status Phase

```
; ENTRY:      TCH, TCM, TCL = 0
;
STATUSIN:
      MOV     TCL,1             ; Xfer 1 byte
      MOV     PCTL,03H          ; Status Phase
      MOV     SCMD,84H          ; Programmed transfer
SYNC:
      MOV     AL,0F1H           ; Wait for SPC command to begin
      AND     AL,SSTS           ; Check Init, Busy bit & Empty bit
      CMP     AL,0B0H           ; Ready and FIFO not empty ?
      JNE     SYNC              ; No, wait
      MOV     AL,DREG           ; Read status byte
SWAIT:
      CMP     INTS,0            ; Intr ?
      JZ      SWAIT             ; No
      TEST    INTS,10H          ; Command completed ?
      JZ      SERR              ; No, process error
      MOV     INTS,10H          ; Reset CMD complete Intr
      RET                       ; Finished
SERR:
 ...process unexpected condition
 ...bus reset, disconnect or SPC hardware error
```

Status Phase

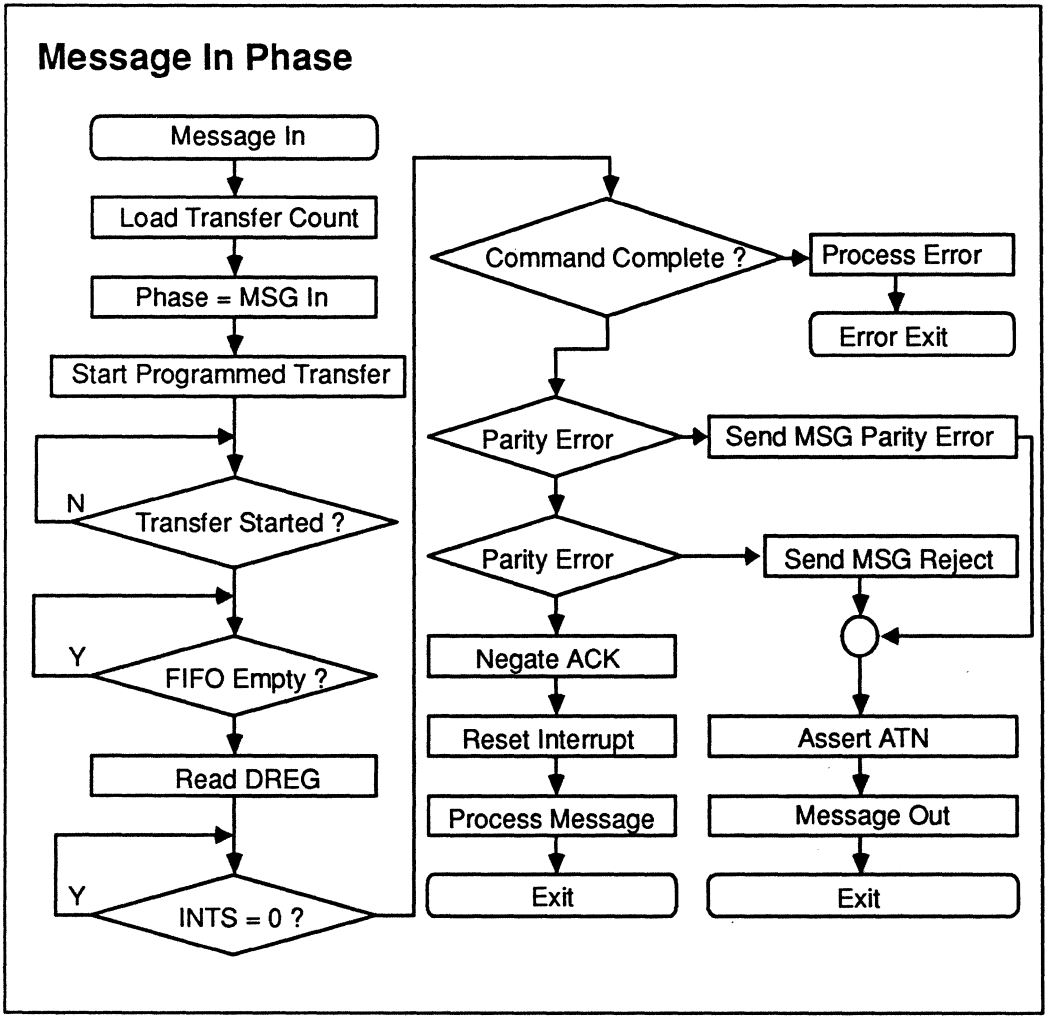


Message In Phase

```
; ENTRY:    TCH, TCM, TCL = 0
;
MGSIN:
     MOV    TCL,1       ; Read 1 byte
     MOV    PCTL,07H    ; Message In phase
     MOV    SCMD,84H    ; Programmed transfer
SYNC:
     MOV    AL,0F1H     ; Wait for SPC command to begin
     AND    AL,SSTS     ; Check Init & Busy bit
     CMP    AL,0B0H     ; Ready and FIFO not empty ?
     JNE    SYNC        ; No, wait
     MOV    AL,DREG     ; Read message byte & automatically hold
                        ; ...ACK active for possible message reject
MIWAIT:
     CMP    INTS,0      ; Intr ?
     JZ     MIWAIT      ; No
     TEST   INTS,10H    ; Command completed ?
     JZ     MIERR       ; No, process error
     TEST   SERR,080H   ; MSG parity error ?
     MOV    AH,09H      ; Parity err code
     JNZ    MSGEJECT    ; Yes, send MSG parity err message
     CALL   PROCESSMS   ; Perform message action
     MOV    G           ; MSG reject code
     JC     AH,07H      ; Not valid MSG, reject it
MIEXIT     MSGREJECT
     MOV                ; Negate ACK
     MOV    SCMD,C0H    ; Reset CMD complete Intr
     RET    INTS,10H    ; Finished
```

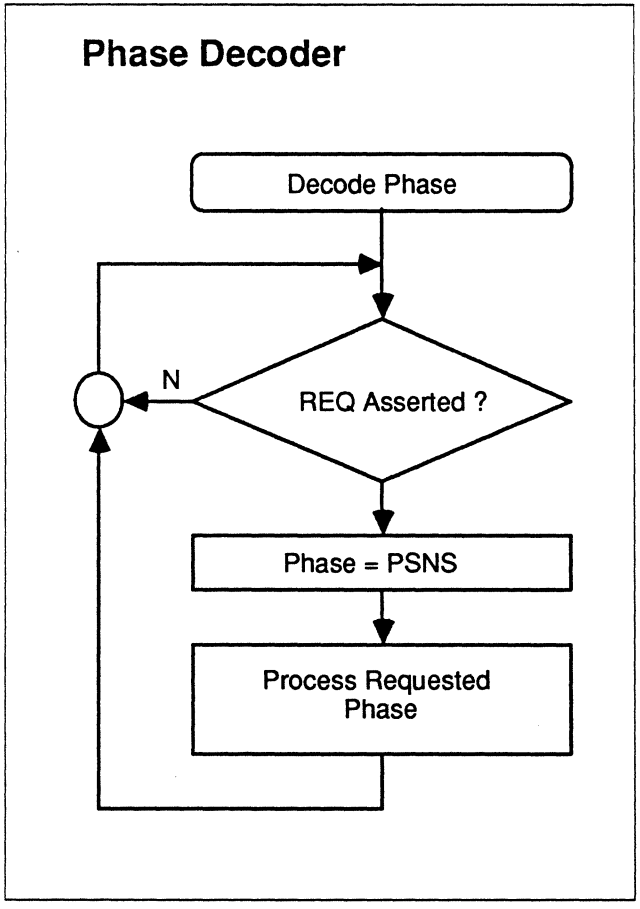
Message In Phase (cont.)

```
MSGREJECT:
      MOV     SCMD,30H      ; Assert ATN, notify target that
                            ; ...last MSG is rejected
      MOV     SCMD,C0H      ; Negate ACK
      MOV     INTS,10H      ; Reset CMD complete Intr
      CALL    CHECMSGOU     ; See if target supports Message Out
      JNE     T             ; No, exit
      CALL    MIEXIT        ; Send MSG
      JMP     MSGOUT        ; Done
MIERR:        MIEXIT
  ...process unexpected condition
  ...bus reset, disconnect or SPC hard-
ware error
```


Message In Phase
Message In
Load Transfer Count
Phase = MSG In
Start Programmed Transfer
N
Transfer Started ?
Y
FIFO Empty ?
Read DREG
Y
INTS = 0 ?
Command Complete ?
Process Error
Error Exit
Parity Error
Send MSG Parity Error
Parity Error
Send MSG Reject
Negate ACK
Reset Interrupt
Process Message
Exit
Assert ATN
Message Out
Exit

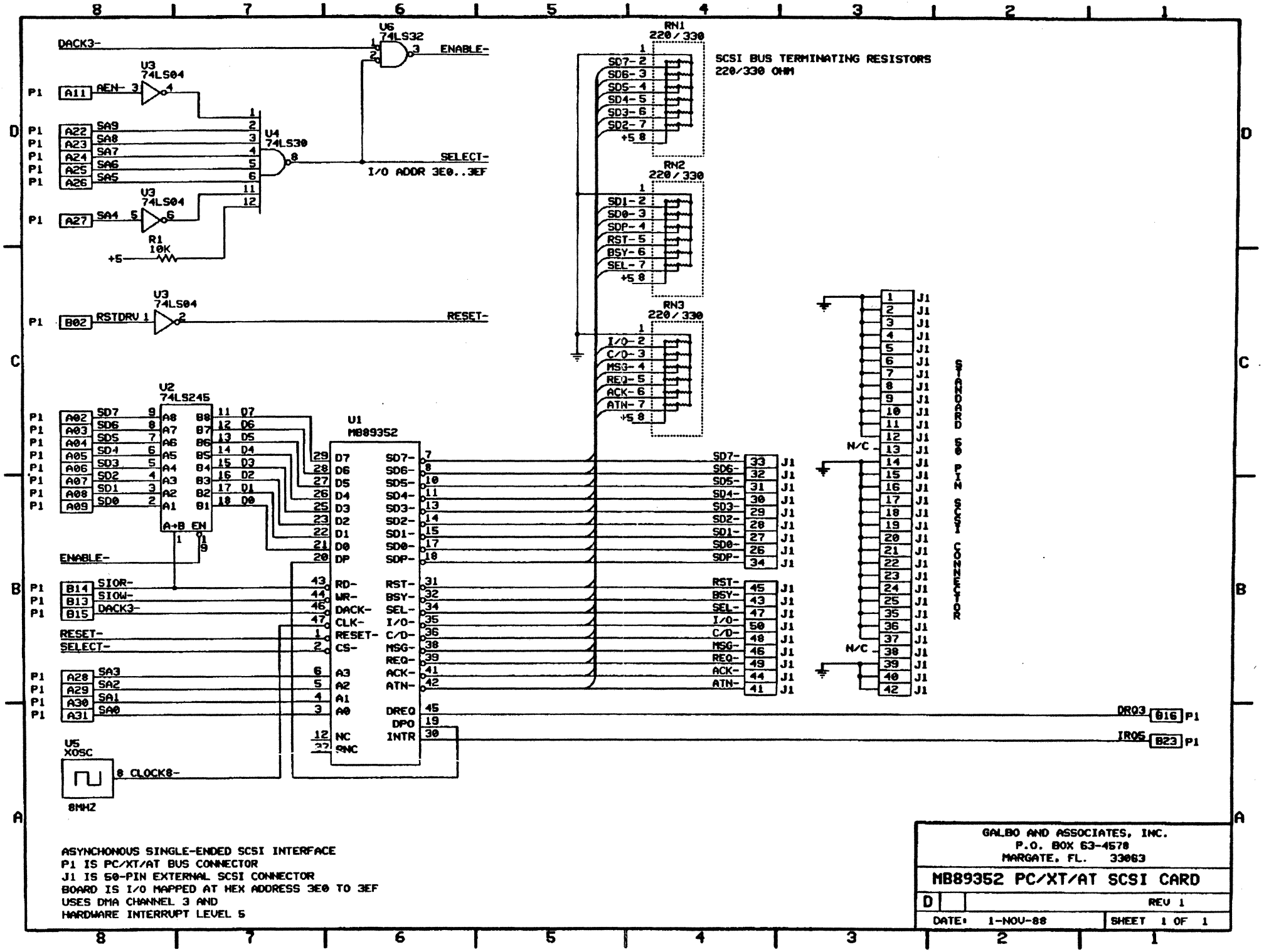
Phase Decoder

```
PHASEDECODE:
     MOV   BL,PSNS             ; Read phase sense registger
     TEST  BL,80H              ; Req asserted ?
     JZ    PHASEDECODE         ; No, wait
     AND   BL,07H              ; Mask out SCSI phase bits
     XOR   BH,BH
     SHL   BX,1                ; *2
     JMP   CS:PHASETABLE[BX]   ; Go to phase processing routine

PHASETABLE:
     DW    DATAOUT             ; 0 = Data Out Phase
     DW    DATAIN              ; 1 = Data In Phase
     DW    SENDCOMMAND         ; 2 = Send Command Phase
     DW    STATUSIN            ; 3 = Status Phase
     DW    PHASEDECODE         ; 4 = Reserved
     DW    PHASEDECODE         ; 5 = Reserved
     DW    MSGOUT              ; 6 = Message Out Phase
     DW    MSGIN               ; 7 = Message In Phase
```


Phase Decoder
Decode Phase
N
REQ Asserted ?
Phase = PSNS
Process Requested Phase

Appendix 4

HARDWARE SCHEMATIC FOR PC HOST ADAPTER



ASYNCHONOUS SINGLE-ENDED SCSI INTERFACE
P1 IS PC/XT/AT BUS CONNECTOR
J1 IS 50-PIN EXTERNAL SCSI CONNECTOR
BOARD IS I/O MAPPED AT HEX ADDRESS 3E0 TO 3EF
USES DMA CHANNEL 3 AND
HARDWARE INTERRUPT LEVEL 5

GALBO AND ASSOCIATES, INC.
P.O. BOX 63-4578
MARGATE, FL. 33063

MB89352 PC/XT/AT SCSI CARD

D | REV 1

DATE: 1-NOV-88 | SHEET 1 OF 1

Appendix 5

SCSI HOST ADAPTER EVALUATION BOARD FOR THE PC, XT, OR AT

- 3.0 Megabyte/Second Asynchronous Transfer Rate*
- Supports up to Eight Arbitrating SCSI Devices and 64 LUNs.

The Fujitsu PC to SCSI host adaptor board is a complete solution to connecting the PC, XT, or AT to the SCSI. The board supports arbitration and disconnection. An optional ROM includes the Interface Common Command Set with extended commands and driver software to control the on-board Fujitsu SCSI Protocol Controller (SPC). The board and ROM is everything you need to thoroughly evaluate the operation of the Fujitsu SPC. It can also be used as a tool for debugging you own SCSI driver, by replacing the socketed ROM with a static RAM. Your code can then be down-loaded directly from the PC to the RAM. The ROM or RAM may be located on any 8K boundary within the PC's one megabyte address range via a DIP switch included on the board. The PC will look between addresses C8000 and E0000 for the extended EIOs, so the board memory should realistically be located somewhere in this range

The 16 SPC registers are memory mapped for speed and flexibility. Memory mapping the registers eliminates I/O Port contention and allows the controller to coexist with the standard disk controller.

The DMA channel is selected via a jumper and can be located on channel 1, 2, or 3 (normally the disk controller uses channel 3). The interrupt level is also selectable via a jumper and can be set at levels 5, 6, or 7 (normally the disk controller uses level 5)..

* Actual data transfer rate will be limited by the PC I/O or DMA bandwidth

Hardware

- Half card for IBM PC, XT, AT and compatibles
- Selectable SCSI ID: 0 to 7
- Selectable PC wait state generator for PC data bus: 0 to 3 wait states
- programmable memory map decode and socket for 8K or 16K EPROM, EEPROM, ROM, and RAM
- 32 selectable PC memory maps in 8K or 16K increments
- External and internal SCSI connectors to interface with external or internally mounted SCSI peripherals
- Odd parity generation for PC bus to SPC
- PC DMA channel interface support
- SPC interrupt mask register

Description

Software

- On-board 8086 code to drive an SCSI mass storage device which in turn can boot the PC at power-up
- Common Command Set accessed as either an independent PC DOS device driver or as a memory resident program via software interrupts
- Direct disk I/O function available through standard ML-DOS interrupt 25H and 26H function calls
- Contains all logic to interface to standard PC or AT bus DMA control

Software Layers

User Interface Layer:	Provides the computer "User" (programmer) a pseudo high-level support mechanism for supporting SCSI I/O functions. It executes commands via the 8086's software interrupt instructions in much the same way as PC-DOS operating system calls. The SCSI instructions can be accessed from this layer using common 8086 languages such as an Assembler, Pascal and C.
Machine Interface Layer:	Interface between PC/XT/AT machine and layers one and two. DMA, Synchronous and Asynchronous I/O are supported in this layer. Also, device driver and /or memory resident program support.
SCSI Layer	Supports the Common Command Set and extended commands.
Hardware I/O Layer:	Supports the low level primitive I/O control between the computer and the SPC.

Feature List for the FM1030M SCSI Host Adapter

General

Interface;	PC/XT/AT
MS/PC DOS	DOS Driver for DOS 2.0 to 4.0 partitioning software User application interface
BIOS ROM	Available
CD-ROM	Initiator or target One internal standard 50 pin dual row SCSI connector One external standard 50 pin D-shell SCSI connector Supports DMA or programmed I/O transfers Supports hardware or polled interrupts Memory mapped I/O for increased performance Low power CMOS controller Half-size card 8-bit bus interface 8K ROM socket EEPROM configuration memory available Address selectable on any 8K boundary Jumperable DMA channel, 1..3 Jumperable interrupt request Level, 2..7 Removable termination resistors Jumperable termination power Fused termination power
Diagnostics:	On-chip
Form Factor:	5.75 x 4.4 x .7

Protocol

Version:	ANSI SCSI-I and SCSI-II
SCSI Message Support:	All non-extended messages
Version:	Common Command Set CCS 4B
Mode:	Hardware – Initiator or Target

Physical

Bus:	Supports Arbitration/Disconnect/Reselect
Parity Checking:	Yes
Transfer type:	Asynchronous
Handshake:	Hardware controlled
Electrical:	Single-ended
Bus Chip:	Fujitsu MB87030
Burst Rate	.3 to 1.μ MB/s (dependent on PC speed/DMA controller)
Connector:	Standard SCSI 50 pin D-shell external Standard SCSI 50 pin dual row internal
Terminators:	Removable
Termination Power:	Selectable

Logical

Firmware:	DOS driver or ROM or both
Initial Configuration:	Automatic
Reset:	Hard or soft

Contact your nearest Fujitsu Sales Office for information on demonstrations and loan agreements for the SCSI board.

Glossary

SCSI TERMS

As with most advances in computer technology, SCSI brings new terms and new meanings to the area of system design. Terms and meanings directly related to SCSI are described in this section.

Term	Meaning
Asynchronous Mode	SCSI operating mode where data transfers between initiator and target occurs at discrete rates without regard to specific amounts of time.
Basis SCSI	SCSI product that does not support arbitration.
Bus Arbitration	In a SCSI system, the ability of both initiators and targets to compete and gain control of the SCSI bus for data transmissions.
Bus Free Phase	SCSI bus is idle.
Command Descriptor Block (CDB)	Series of bytes used to define and initiate a SCSI command.
Command Phase	SCSI-bus phase where initiator is sending CDB to target.
Daisy-Chained	A method of electrically connecting multiple devices together so all devices use a common set of command/data transfer signals.
Direct Memory Access (DMA)	Method of data transfer to main memory without use of the CPU.
Full SCSI	SCSI product that supports arbitration and the Reconnect/Disconnect options.

Term	Meaning
Handshake	Control signals that initiate an asynchronous data transfer between two devices. A handshake is required for each byte transferred and no specific response time is required for either the initiator or the target.
Host Adapter	Generally, the host adapter is a plug-in board that performs the host bus -to-SCSI conversion.
Initiator	SCSI device that is capable of beginning a data transfer operation. Usually, the initiator is the host computer; however, in systems with arbitration. The target can also serve as the initiator.
Intelligent Interface	Any interface (like SCSI) which can execute a high-level protocol and a supporting set of commands.
Mandatory Command	SCSI command that must be supported by the products to remain compatible with the ANSI specifications.
Message Phase	SCSI-bus phase at which time the initiator and target exchange information regarding recent transfers of data.
Non-arbitrating	A non-arbitrating SCSI configuration is made up of a single initiator and one or more targets. The initiator is the bus master and the targets ar slave. Reselection s not supported since the reselection process requires the targets to arbitrate for the bus. In this basic SCSI configuration, the initiator selects a target and remains connected to the target until the transfer is complete
Optional Command	SCSI command which may or may not be implemented in an SCSI product; the absence of such a command does not affect compatibility.

Term	Meaning
Reconnect/Disconnect	Ability of SCSI device to "disconnect" from the SCSI bus before completion of certain commands and "reconnect" at a later time.
Reselection Phase	SCSI-bus phase where a device has won arbitration and a connection between an initiator and a target is re-established.
Reserved	Command code or byte in SCSI protocol reserved for a future function. The use of reserved bytes violates the requirements for SCSI compatibility.
SCSI ID	A one-of-eight code used to address a device connected to the SCSI bus.
Status Phase	SCSI-bus phase where the target sends status information to the initiator.
Synchronous	SCSI method of transferring data from initiator-to-target (or vice-versa_ in a specified time interval. In synchronous mode, a data transfer rate of 4.0 megabytes-per-second is possible.
Target	Any SCSI device capable of receiving commands or data from another SCSI device.
Vendor Unique	SCSI command codes which are undefined.
X3T9.2	The ANSI number designation for the small computer systems interface.,