NCR 5385E SCSI

Protocol Controller

Data Sheet

NCR

Microelectronics Division, Colorado Springs

This document contains the latest information available at the time of publication. However, NCR reserves the right to modify the contents of this material at any time. Also, all features. functions and operations described herein may not be marketed by NCR in all parts of the world. Therefore, before using this document, consult your NCR representative or NCR office for the information that is applicable and current.

TABLE OF CONTENTS

SECTION 1
GENERAL DESCRIPTION

The NCR SCSI Protocol Controller (SPC) is designed to accomodate the Small Computer Systems Interface (SCSI) as defined by the ANSI X3T9.2 committee. The SPC operates in both the Initiator and Target roles and can therefore be used in host adapter and control unit designs. This device supports arbitration, including reselection, and is intended to be used in systems that require either open collector or differential pair transceivers.

The NCR 5385E SCSI Protocol Controller communicates with the system microprocessor as a peripheral device. The chip is controlled by reading and writing several internal registers which may be addressed as standard or memory mapped I/O. A 24-bit Transfer Counter and the appropriate handshake signals accommodate large DMA transfers with minimal processor intervention. Since the NCR 5385E interrupts the MPU when it detects a bus condition that requires servicing, the MPU is freed from polling or controlling any of the SCSI bus signals.

Below is a list of important features:

SCSI INTERFACE

*Supports ANSI X3T9.2 SCSI Standard

- Asynchronous data transfers to 1.5 MBPS
- Supports both Initiator and Target roles
- **Parity generation with optional checking**
- Supports arbitration
- Controls all bus signals except Reset
- Doubly-buffered Data Register

MPU INTERFACE

*Versatile MPU Bus Interface

- Memory or I/O mapped MPU interface
- DMA or programmed I/O transfers
- 24-bit Internal Transfer Counter
- Programmable (Re)Selection timeouts
- Interrupts MPU on all bus conditions requiring service

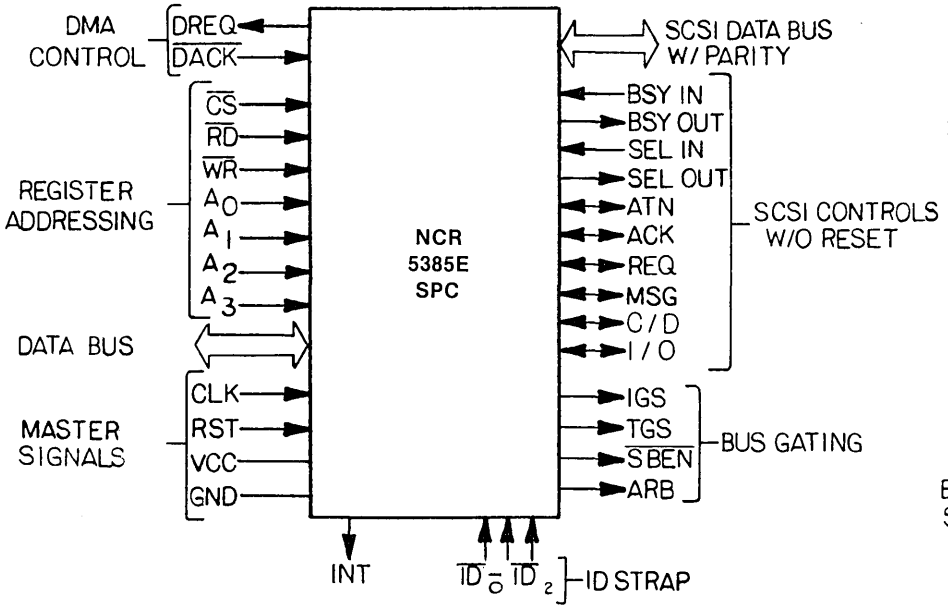


FIG. 1.1 FUNCTIONAL PIN GROUPING

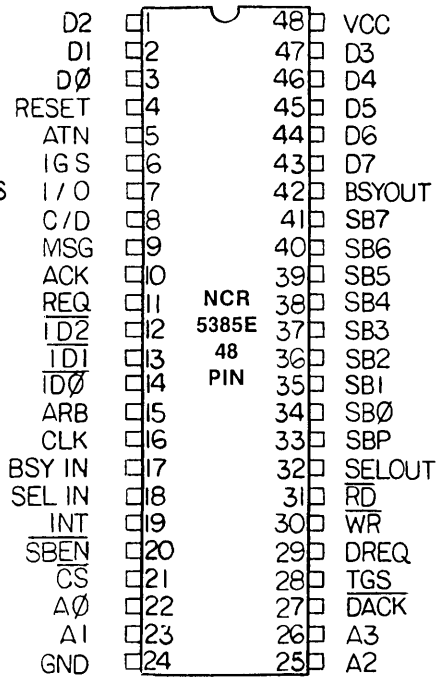


FIG. 1.2 PINOUT

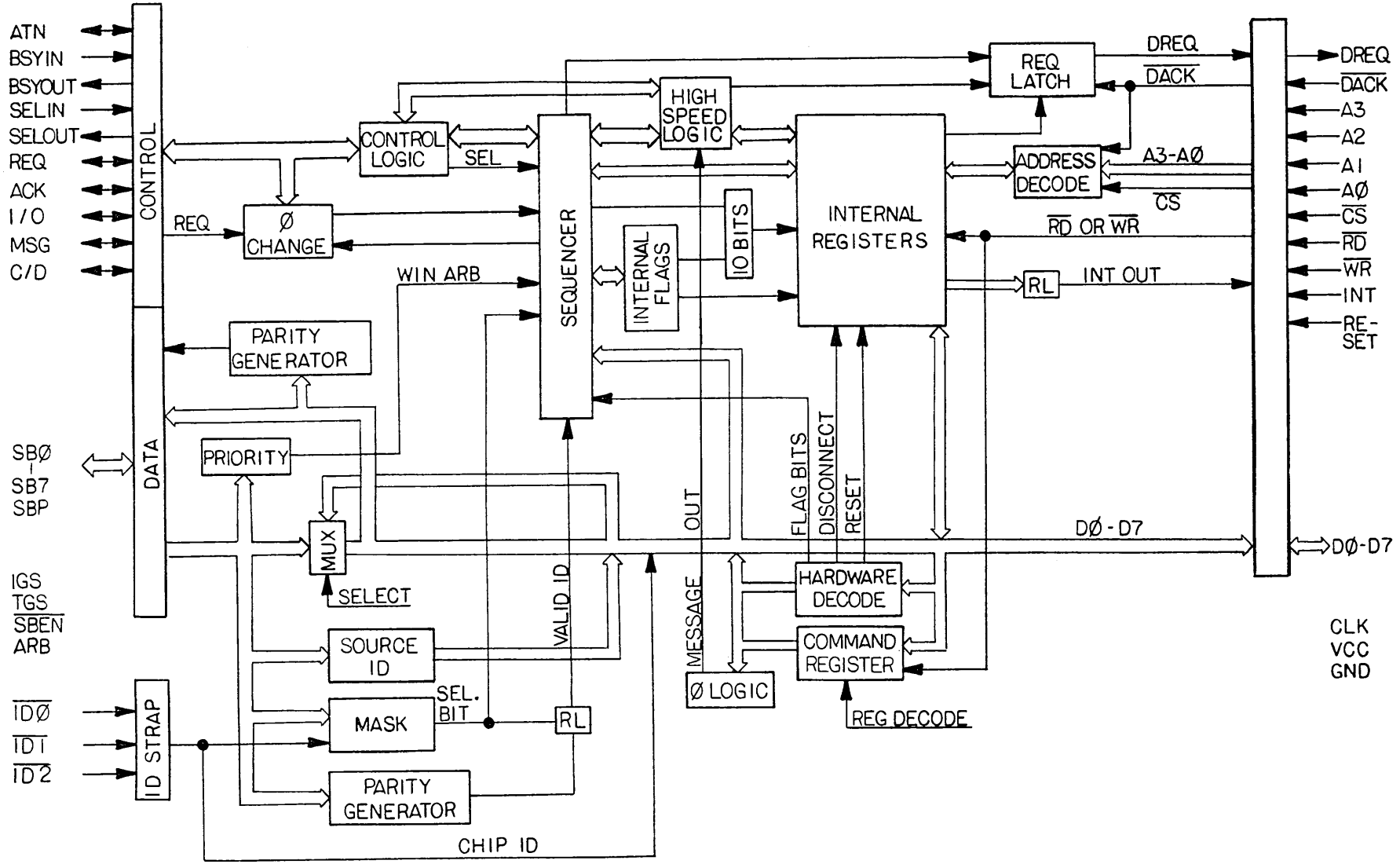


FIG. 1.3 NCR 5385E BLOCK DIAGRAM

SECTION 2
PIN DESCRIPTION

2.1 MICROPROCESSOR INTERFACE SIGNALS

CLK	16	Symmetrical square wave signal which generates internal chip timing. Maximum frequency is 10 MHz.
RESET	4	When high (1), this signal forces the chip into a reset state. All current operations are terminated. Internal storage elements are cleared and self-diagnostics are performed.
D0-D7	3-1 47-43	These signals comprise an active high data bus. It is intended that these signals be connected to the microprocessor data bus.
INT	19	This signal is used to interrupt the microprocessor for various bus conditions that require service. INT is set high for request and cleared when the chip is reset or the Interrupt Register is read.
$\overline{WR}$	30	Write pulse (active low) is used to strobe data from the data bus into an internal register which has been selected.
$\overline{RD}$	31	Read pulse (active low) is used to read data from an internal register that has been selected. The contents of the register is strobed onto the data bus.
$\overline{CS}$	21	When low (0), this signal enables reading from or writing to the internal register which has been selected.
A0-A3	22, 23, 25, 26	These signals are used in conjunction with $\overline{CS}$, to address all the internal registers.
DREQ	29	Data request. When high (1), this signal indicates that the internal Data Register has a byte to transfer (inputting from the SCSI bus) or needs a byte to transfer (outputting to the SCSI bus). This signal becomes active only if the DMA mode bit in the Command Register is on. It is cleared when $\overline{DACK}$ becomes active.
$\overline{DACK}$	27	Data acknowledge. When low (0), this signal resets DREQ and selects the Data Register for input or output. $\overline{DACK}$ acts as a chip select for the Data Register when in the DMA mode. $\overline{DACK}$ and $\overline{CS}$ must never be active at the same time.

2.2 SCSI INTERFACE SIGNALS

$\overline{ID0}$ -$\overline{ID2}$	14-12	These active low signals determine the three-bit code of the SCSI bus ID assigned to the chip. External pullup resistors are required only if tied to switches or straps.
SB0-SB7, SBP	34-41, 33	Active high data bus. These signals comprise the SCSI data bus and are intended to be connected to the external SCSI bus transceivers.
BSYIN	17	When high (1), this signal indicates to the chip that the SCSI BSY signal is active.
BSYOUT	42	When high (1), the chip is asserting the BSY signal to the SCSI bus.
SELIN	18	When high (1), this signal indicates to the chip that the SCSI SEL signal is active.
SELOUT	32	When high (1), the chip is asserting the SEL signal to the SCSI bus.
ATN	5	INITIATOR ROLE: The chip asserts this signal when the microprocessor requests the attention condition or a parity error has been detected in a byte received from the SCSI bus. TARGET ROLE: This signal is an input which indicates the state of the ATN signal on the SCSI bus.
ACK	10	INITIATOR ROLE: The chip asserts this signal in response to REQ for a byte transfer on the SCSI bus. TARGET ROLE: This signal is an input which, when active, indicates a response to the REQ signal.
REQ	11	INITIATOR ROLE: This signal is an input which, when active, indicates that the Target is requesting a byte transfer on the SCSI bus. TARGET ROLE: Asserted by the chip to request a byte transfer on the SCSI bus.
MSG, C/D, I/O	9, 8, 7	INITIATOR ROLE: These signals are inputs which indicate the current SCSI bus phase. TARGET ROLE: The chip drives these signals to indicate the current bus phase.
IGS	6	Initiator Group Select. When high (1), this signal indicates to the external SCSI drivers that the chip is controlling in the Initiator role. Its purpose is to enable the external drivers for ATN and ACK.
TGS	28	Target Group Select. When high (1), this signal indicates to the external SCSI drivers that the chip is controlling in the Target role. Its purpose is to enable the external drivers for REQ, MSG, C/D, and I/O.

$\overline{\text{SBEN}}$	20	SCSI data Bus Enable. When low (0), this signal directly enables the external SCSI data bus drivers.
ARB	15	Arbitration phase. When high (1), this signal enables the external circuitry to place the ID bit on the SCSI bus for the Arbitration phase.

POWER SIGNALS

VCC	48	+5 V input
GND	24	Signal reference input

SECTION 3
ELECTRICAL CHARACTERISTICS

OPERATING CONDITIONS

PARAMETER	SYMBOL	MIN	MAX	UNITS
Supply Voltage	V_{DD}	4.75	5.25	VDC
Supply Current	I_{DD}		300	mA
Ambient Temp.	T_A	0	70	°C

INPUT SIGNAL REQUIREMENTS

PARAMETER	CONDITIONS	MIN	MAX	UNITS
High-level Input, V_{IH}		2.0	5.25	VDC
Low-level Input, V_{IL}		-0.3	0.8	VDC
High-level Input Current, I_{IH}	$V_{IH} = 5.25V$		10	μA
Low-level Input Current, I_{IL}	$V_{IL} = 0V$		-10	μA

OUTPUT SIGNAL REQUIREMENTS (Except $\overline{SBEN}$, IGS, and TGS)

PARAMETER	CONDITIONS	MIN	MAX	
High-level Output Voltage, V_{OH}	$V_{DD} = 4.75V$ @ $I_{OH} = -400\mu A$	2.4	—	VDC
Low-level Output Voltage, V_{OL}	$V_{DD} = 4.75V$ @ $I_{OL} = 2.0mA$	—	0.4	VDC

$\overline{SBEN}$, IGS, and TGS SIGNALS

PARAMETER	CONDITIONS	MIN	MAX	UNITS
High-level Output Voltage, V_{OH}	$V_{DD} = 4.75V$ @ $I_{OH} = -400\mu A$	2.4	—	VDC
Low-level Output Voltage, V_{OL}	$V_{DD} = 4.75V$ @ $I_{OL} = 4.0mA$	—	0.4	VDC

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

SECTION 4
INTERNAL REGISTERS

4.0 GENERAL

The NCR SCSI Protocol Controller has a set of internal registers which are used by the microprocessor to direct the operation of the SCSI bus. These registers are read (written) by activating $\overline{CS}$ with an address on A3-A0 and then issuing a $\overline{RD}$/($\overline{WR}$) pulse. They can be made to appear to a microprocessor as standard I/O ports or as memory-mapped I/O ports depending on the external circuitry that controls $\overline{CS}$. The following sections describe the operation of these internal registers.

REGISTER SUMMARY

A3	A2	A1	A0	R/W	REGISTER NAME
0	0	0	0	R/W	Data Register
0	0	0	1	R/W	Command Register
o	0	1	0	R/W	Control Register
0	0	1	1	R/W	Destination ID
0	1	0	0	R	Auxiliary Status
0	1	0	1	R	ID Register
0	1	1	0	R	Interrupt Register
0	1	1	1	R	Source ID
1	0	0	1	R	Diagnostic Status
1	1	0	0	R/W	Transfer Counter (MSB)
1	1	0	1	R/W	Transfer Counter (2nd BYTE)
1	1	1	0	R/W	Transfer Counter (LSB)
1	1	1	1	R/W	Reserved for Testability

4.1 DATA REGISTER

The Data Register is used to transfer SCSI commands, data, status and message bytes between the microprocessor data bus and the SCSI bus. This is an eight-bit register which is doubly-buffered in order to support maximum throughput. In the non-DMA mode, the microprocessor reads from (writes to) the Data Register by activating $\overline{CS}$ with A3-A0 = 0000 and issuing a $\overline{RD}$/($\overline{WR}$) pulse. A bit has been included in the Auxiliary Status Register to indicate when the Data Register is full. In the DMA mode, the DMA logic reads from (writes to) the Data Register by responding to DREQ with $\overline{DACK}$ and issuing a $\overline{RD}$/($\overline{WR}$) pulse. The SCSI bus reads from or writes to the Data Register when the chip is connected as an Initiator or Target and the bus is in one of the Information Transfer Phases.

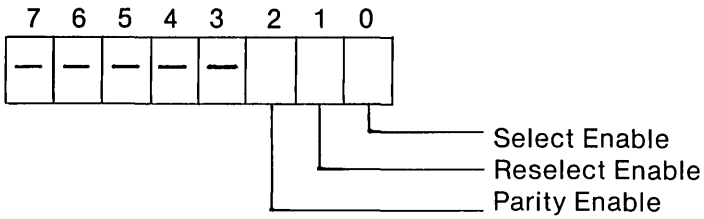
4.2 COMMAND REGISTER

The Command Register is an eight-bit register used to give commands to the SCSI chip. The microprocessor can write to (read from) the Command Register by activating $\overline{CS}$ with A3-A0 = 0001 and issuing a $\overline{WR}$/($\overline{RD}$) pulse. Writing to the Command Register causes the chip to execute the command that is written. The Command Register can be read; however, the chip resets the Command Register when it sets an Interrupt. Therefore, one cannot guarantee that the data in the register will be correct after loading an interrupting command or enabling selection or reselection. To be safe, a copy of the last command issued should be stored in the microprocessor's memory. Immediate commands are not stored.

The contents of the Command Register are described in a later section (See page 19, COMMANDS).

4.3 CONTROL REGISTER

This eight-bit read/write register is used for enabling certain modes of operation for the SCSI Protocol Controller. The microprocessor reads from (writes to) the Control Register by activating $\overline{CS}$ with A3-A0 = 0010 and issuing a $\overline{RD}$ ($\overline{WR}$) pulse.



BIT 7-3 Reserved

BIT 2 Parity Enable — When the parity enable bit is a "1", the chip generates and checks parity on all transfers on the SCSI bus. When the parity enable bit is a "0", the chip generates but does not check parity on bus transfers.

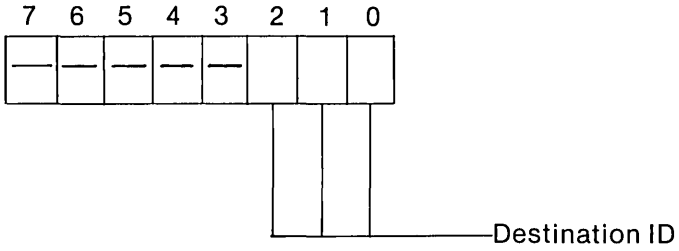
BIT 1 Reselect Enable — When this bit is a "1", the chip will respond to any attempt by a Target to reselect it. When the bit is a "0", the chip will ignore all attempts to reselect it.

BIT 0 Select Enable — When this bit is a "1", the chip will respond to attempt to select it as a Target. When it is a "0", the chip will ignore all selections.

NOTE: After being reset and completing self-diagnostics, the control register will contain all zeros.

4.4 DESTINATION ID REGISTER

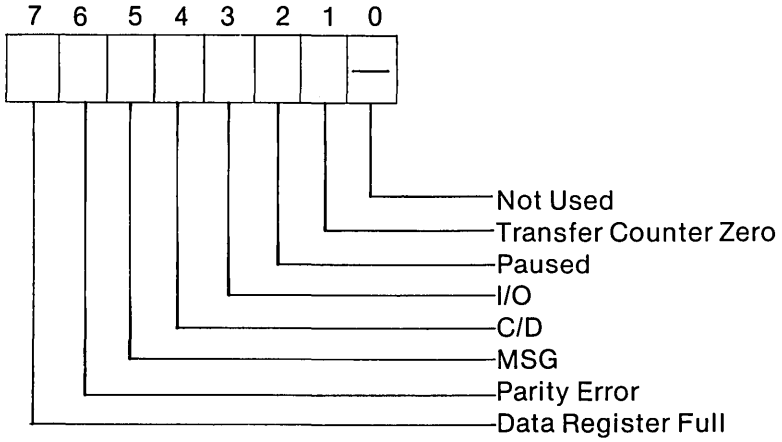
The Destination ID Register is an eight-bit register that is used to program the SCSI bus address of the destination device prior to issuing a Select or Reselect command to the chip. Bits 0-2 specify the address and bits 3-7 are always zeroes. The ID register is written (read) by activating $\overline{CS}$ with A3-A0 equal to "0011" and then pulsing $\overline{WR}$ ($\overline{RD}$).



4.5 AUXILIARY STATUS REGISTER

The Auxiliary Status Register is an eight-bit read-only register. It contains bits which indicate the status of the chip's operational condition. Some of these bits are used to determine the reason for interrupts. Therefore, the Auxiliary Status Register should always be read prior to reading the Interrupt Register when servicing interrupts. After the Interrupt Register is read, the Auxiliary Status Register bits needed to service the interrupt may change.

The Auxiliary Status Register is read by activating $\overline{CS}$ with A3-A0 = 0100 and then pulsing $\overline{RD}$. The individual bits of the Auxiliary Status Register are defined below.



BIT 7 Data Register Full

This bit indicates the status of the Data Register and must be monitored by the microprocessor during non-DMA mode commands that use the Data Register. When the DMA mode bit in the Command Register is off (0) and the command being executed is one of Send, Receive or Transfer Info commands (refer to Section 5.0 page 19, COMMANDS), data is transferred to (from) the chip by writing (reading) the Data Register. Data Register Full is set on (1) when data is written and turned off (0) when data is read. Therefore, Data Register Full should be on before taking data from the chip, and off when sending data to the chip.

The Data Register Full bit is always reset (to 0) at the time an interrupting type command is loaded into the Command Register. Therefore, when issuing such commands, the Command Register should be loaded prior to loading the Data Register and monitoring the Data Register Full flag.

BIT 6 Parity Error

When this bit is one, it indicates that the chip has detected a parity error on a byte of data received across the SCSI bus. It can be set when the chip is executing one of the Receive commands or the Transfer Info command (when the transfer is an input). This bit is reset after the Interrupt Register is read.

BIT 3-5 I/O, C/D, MSG

These bits indicate the status of the SCSI I/O, C/D, and MSG signals at all times. They define the Information Phase type being requested by the Target. These signals are significant when servicing interrupts and the chip is logically connected to the bus in the Initiator role. An interrupt will occur with any phase change. This allows the Initiator to prepare for the next phase of data transfer. These bits are only held while INT is active. The bits are coded as follows:

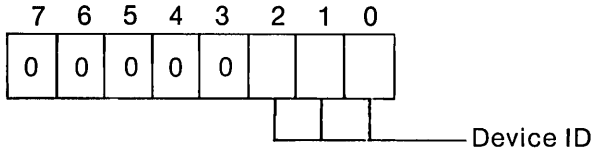
I/O	C/D	MSG	BUS PHASE
0	0	0	Data Out
0	0	1	Unspecified Info Out
0	1	0	Command
0	1	1	Message Out
1	0	0	Data In
1	0	1	Unspecified Info In
1	1	0	Status
1	1	1	Message In

BIT 2	Paused	When on (1), this bit indicates that the chip has aborted the command being executed in response to the Pause command. It is turned off when the interrupting type command code is loaded into the Command Register.
BIT 1	Transfer Counter Zero	This bit is provided to indicate the status of the 24-bit Transfer Counter. When on (1), it indicates that the Transfer Counter is equal to zero. It is intended to facilitate interrupt servicing.
BIT 0	Not Used	

NOTE: The Auxiliary Status Register will contain the following pattern after a Reset and self-diagnostics: 00xxx010.

4.6 ID REGISTER

The ID Register is an eight-bit read-only register that indicates the logical SCSI bus address occupied by the chip. Bit 0-2 directly reflect the logical inversion of the chip ID input signals $\overline{ID0}$-$\overline{ID2}$. The ID Register is active high whereas the ID input signals are active low. The ID Register allows the microprocessor to read the chip's SCSI bus address which would normally be strapped in hardware. Bits 3-7 of the ID Register will always be zeroes. The ID Register is read by activating $\overline{CS}$ with A3-A0 = 0101 and then pulsing $\overline{RD}$.

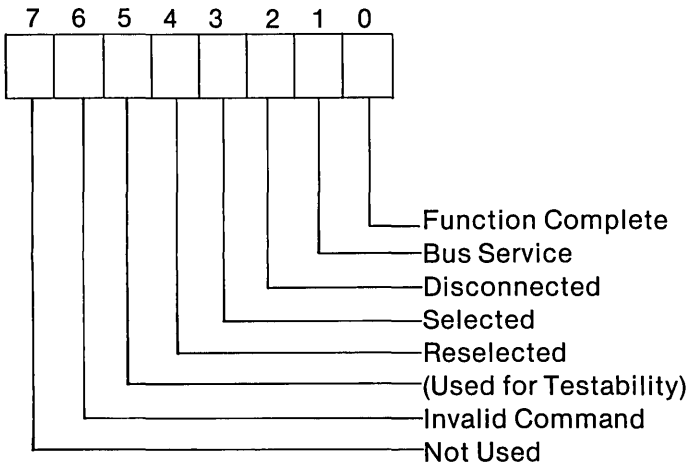


4.7 INTERRUPT REGISTER

The Interrupt Register is an eight-bit read-only register. It is used in conjunction with the Auxiliary Status Register to determine the reason for an interrupt condition. This register is read by activating $\overline{CS}$ with A3-A0 = 0110 and then pulsing $\overline{RD}$. When the Interrupt Register is read, it automatically resets itself (after the read is complete) and enables the chip for a new interrupt condition. Since the Parity Error bit in the Auxiliary Status Register is reset after a read of the Interrupt Register, and since I/O, C/D, and MSG are only held while INT is active, the Auxiliary Status Register should always be read prior to reading the Interrupt Register.

If a Selected or Reselected interrupt occurs after issuing a command that would normally cause an interrupt, the chip will ignore the last command issued. This allows the microprocessor to service the Selected or Reselected interrupt prior to proceeding with the other operation. An example of this situation is when the microprocessor issues a command to select a Target at about the same time another Target reselects the chip. If the chip sees the reselection first, the microprocessor will receive an interrupt for the reselection, and the chip will ignore the Select command, which would now be invalid since the chip is now logically connected on the SCSI bus to another device.

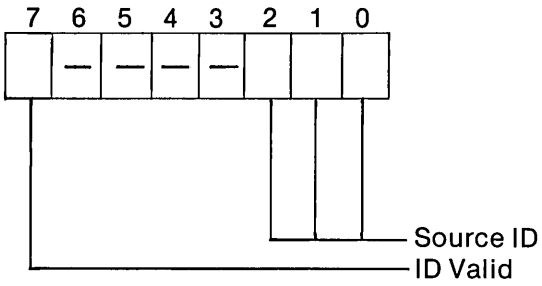
Individual interrupt conditions are described below. (Note: that for all cases, an interrupt condition is on, when the corresponding bit is a one (1), and off when zero (0).)



BIT 7	Not Used	May be either (1) or (0).
BIT 6	Invalid Command	When on (1), this bit indicates that the last command loaded into the Command Register is not valid.
BIT 5	Not Used	(Reserved for testability)
BIT 4	Reselected *	This interrupt will be on (1) when the chip has been reselected by another SCSI device. After setting this interrupt, the chip is logically connected to the bus in an Initiator role and is waiting for the Target to send REQ or disconnect from the bus.
BIT 3	Selected *	This interrupt will be on (1) whenever the chip has been selected by another SCSI device. After setting this interrupt, the chip is logically connected to the bus in the Target role and is waiting for a command to be loaded into the Command Register. * The chip will become selected (reselected) only if the ID data byte put on the SCSI bus during the Selection (Reselection) Phase has good parity and not more than one ID other than the chip's own ID is on.
BIT 2	Disconnected	This interrupt will be set on (1) when the chip is connected to the bus in the Initiator role and the Target disconnects or when the chip is executing a Select or Reselect command and the destination device does not respond before the Transfer Counter times out.
BIT 1	Bus Service	When the chip is logically connected to the bus in the Initiator role, this bit will be set on (1) whenever the Target sends a REQ which the chip cannot automatically handle. This happens when the first REQ for connection is received or when the chip is executing a Transfer Info or Transfer Pad command and either the Transfer Counter is zero or the Target changes the Information Phase type. A Bus Service interrupt may also be set if a phase change occurs before REQ is seen. This early notification will allow the Initiator extra time to prepare for a phase change in some unbuffered systems. (Note: that the chip may generate Bus Service Interrupts for phases that never request transfers. This is not an error condition, merely transitional status of I/O, C/D, and MSG.) If the chip is logically connected in the Target role, this bit will be set on (1) whenever the Initiator asserts ATN. When indicating ATN the Bus Service interrupt may occur by itself, with a Selected interrupt, or with a Function Complete interrupt.
BIT 0	Function Complete	When this bit is on (1), it indicates that the last interrupting command has completed. It is the normal successful completion interrupt for Select, Reselect, Send and Receive commands (Refer to Section 5.0 page 19, COMMANDS). During any of the Receive commands, it is set on (1) along with the parity error bit as soon as a parity error is detected. A Bus Service Interrupt may also occur simultaneously with the Function Complete if an ATN signal was activated during a Send or a Receive command. The Function Complete interrupt is also generated at the end of a Message In phase for a Transfer Info command. (See TRANSFER INFO command, page 29 for details.)

4.8 SOURCE ID REGISTER

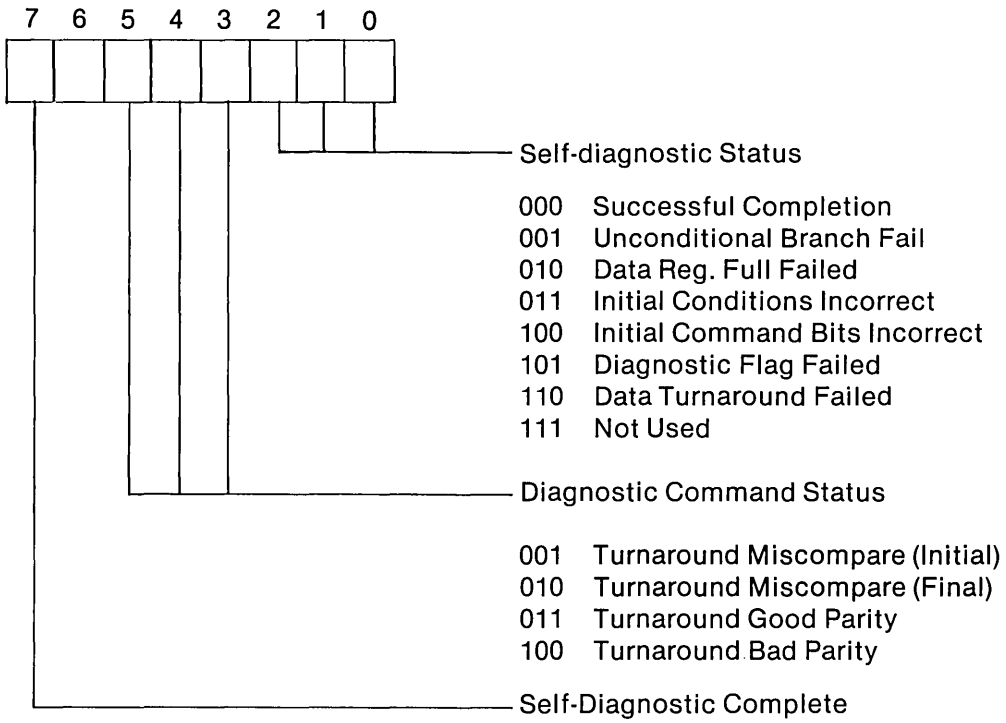
The Source ID Register is an eight-bit read-only register which contains the three-bit encoded ID of the last device which Selected or Reselected the chip. The following is the format of the Source ID Register.



The ID Valid bit indicates that the source device placed its own ID bit on the SCSI bus during the Selection Phase. The SPC chip has encoded the source ID and placed it in bits 2-0. This information remains valid until the chip disconnects from the SCSI bus, at this time the ID Valid bit is reset.

4.9 DIAGNOSTIC STATUS REGISTER

The Diagnostic Status Register is an eight-bit read-only register which indicates the result of self-diagnostics and the last diagnostic command issued to the chip. The format of the Diagnostic Status Register is shown below.



Bit 7 = 1 indicates that self-diagnostics have been completed. (NOTE: A reset will clear bits 6-3 if possible). After a reset to the chip, the microprocessor should make sure that the Diagnostic Status Register contains the following pattern before attempting any commands: 10000000. This code indicates self-diagnostics are complete and no errors were detected. After a diagnostic command has been executed, bits 6-3 will contain the resulting status, but bit 7 and bits 2-0 are not affected.

The microprocessor may read the Diagnostic Status Register by activating $\overline{CS}$ with A3-A0 = 1001 and issuing a $\overline{RD}$ pulse.

If an error is detected during self-diagnostics, the proper status is loaded into the Diagnostic Status Register and the chip halts until a Reset command or a Reset signal is asserted. Refer to the Self-Diagnostic Status Code Summary for an explanation of the individual codes.

When a diagnostic command is issued to the chip, the chip will attempt to perform the function, load a status into bits 6-3, and initiate a Function Complete Interrupt.

4.9.1 SELF-DIAGNOSTIC STATUS CODE SUMMARY

000 - Successful Completion. The chip executed all self-diagnostics following a reset and detected no errors.

001 - Unconditional Branch Failed. The chip's internal sequencer attempted an unconditional branch and failed to reach the desired location.

010 - Data Register Full Failed. The chip attempted to set and reset the Data Register Full status bit in the Interrupt Register and failed.

011 - Initial Conditions Incorrect. The chip detected one of its internal initial conditions in the wrong state.

100 - Initial Command Bits Incorrect. The chip tested bits 6,4,2,1 and 0 of the Command Register and found at least one was not zero.

101 - Diagnostic Flag Failed. The chip failed in its attempt to set and reset its internal diagnostic flag.

110 - Data Turnaround Failed. During self-diagnostics the chip attempts to flush several bytes of data through its internal data paths. It also attempts to set and reset the Parity Error bit in the Interrupt Status Register. This status indicates that one of these operations failed.

4.10 TRANSFER COUNTER (THREE EIGHT-BIT COUNTERS)

The Transfer Counter is comprised of three, eight-bit register/counters. It is used by the chip for Send, Receive and Transfer commands that require more than a single byte of data to be transferred. It may also be used with Select and Reselect commands to set a timeout for no response. To write to (read from) the Transfer Counter, $\overline{CS}$ is activated with A3-A0 selecting a byte and then pulsing $\overline{WR}$ ($\overline{RD}$). The Transfer Counter is addressed as shown below.

A3	A2	A1	A0	SELECTED BYTE
1	1	0	0	Most Significant Byte
1	1	0	1	Middle Byte
1	1	1	0	Least Significant Byte

For Send, Receive and Transfer commands with single-byte not specified, the Transfer Counter specifies to the chip the maximum number of bytes to be sent or received before interrupting. The Transfer Counter must be loaded prior to issuing the command. When single-byte is specified, the chip neither uses nor alters the Transfer Counter. To facilitate servicing interrupts for commands that use the Transfer Counter, a bit is provided in the Auxiliary Status Register to indicate when the Transfer Counter is zero.

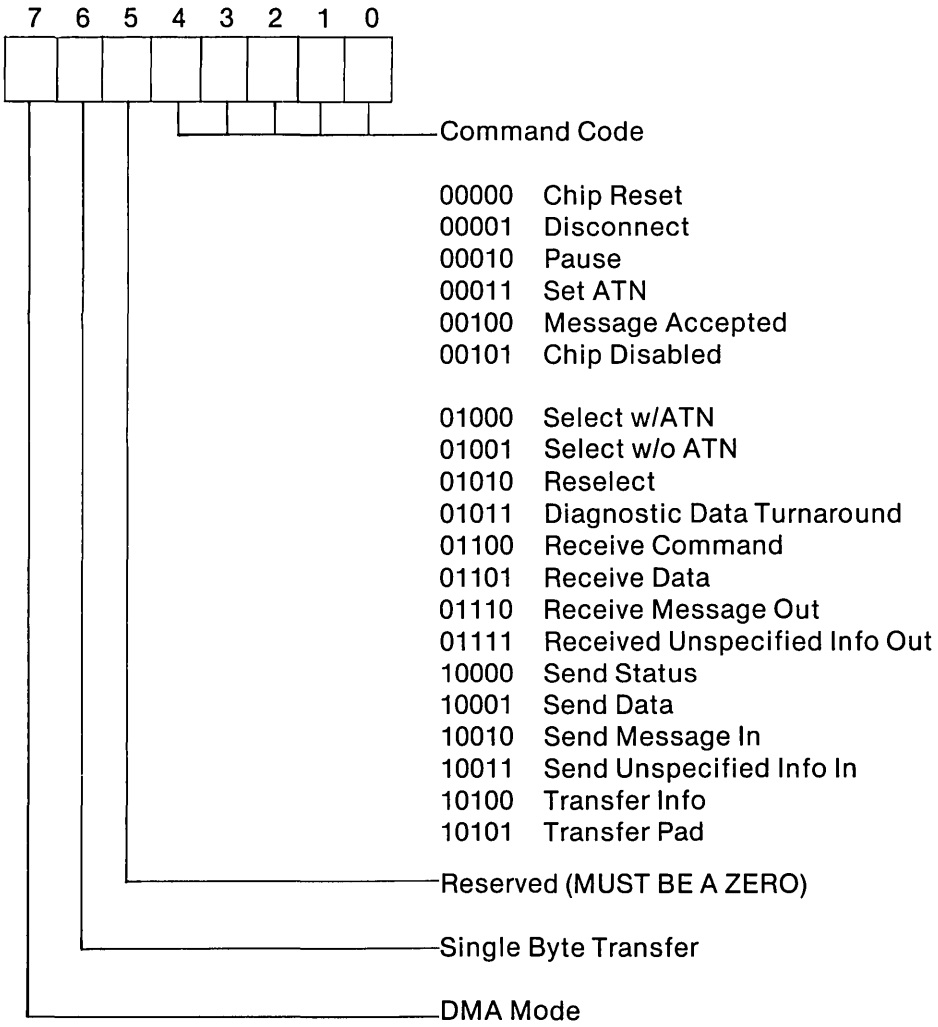
For Select and Reselect commands, the Transfer Counter specifies the number of time intervals (1024 CLK periods) that the chip will wait before automatically aborting the command due to no response (BSY) from the destination device. The Transfer Counter must be loaded prior to issuing the command. If the Transfer Counter is loaded with all zeroes, the timeout logic in the chip will be disabled, and the chip will not automatically abort the command due to no response.

SECTION 5
COMMANDS

This section defines command format, types, codes and operation. Commands are given to the chip by loading the Command Register.

5.1 COMMAND FORMAT

The bits in the Command Register are defined as follows.



BIT 7	DMA Mode	This bit is applicable only for commands that use the Data Register. When this bit is on (1), it indicates that data will be transferred to (from) the Data Register using the DMA signals DREQ and $\overline{\text{DACK}}$. When it is off (0), the microprocessor must monitor the state of the Data Register Full flag in the Auxiliary Status Register. Data is then transferred by using the appropriate input/output command.
BIT 6	Single Byte Transfer	When on (1), this bit indicates that only one byte of data is to be transferred for this command. The Transfer Counter will not be used or altered by the chip. Therefore, for common single byte message and status transfers, the Transfer Counter does not need to be loaded prior to issuing a command with this bit set. When this bit is off (0). the Transfer Counter is used by the chip to determine the length of the transfer for the command.
BIT 5	Reserved	This bit is not used and should always be programmed off (0).
BIT 4-0	Command Code	These bits are used to specify the command to be executed.

5.2 COMMAND TYPES

There are two types of commands; Immediate and Interrupting. All of the Immediate commands, except for Pause, cause immediate results within three clock cycles from the time the Command Register is loaded. The Pause command is explained in a later section (See page 22, PAUSE). Interrupting commands do not result in immediate action. Their completion is always flagged by an interrupt.

Command codes 00000-00111 specify Immediate commands. Immediate commands that are listed as reserved, will be ignored if issued to the SPC chip. Command codes 01000-10101 specify Interrupting commands. When one of these codes is loaded into the Command Register, a second Interrupting command code should not be loaded until after the interrupt has occurred for the first command. However, an Immediate type command may be loaded before the interrupt for an Interrupting command occurs. If a reserved Interrupting command code is issued, the chip will respond with an Invalid Command interrupt.

5.3 INVALID COMMANDS

The user of the chip can be in one of three states at any particular time: Disconnected, connected as an Initiator, or connected as a Target. Commands are valid only in specified states. If an invalid Immediate command is issued, the chip will ignore the command. If an Interrupting command is issued in an invalid state, or a reserved Interrupting command code is issued, an Invalid Command interrupt will result. The exceptions are described below:

The microprocessor must never issue any interrupting type command when the chip is not expecting such a command. Unpredictable results will occur in this case. The following is a list of user states in which the chip is not expecting an interrupting command:

1. The chip is currently processing an Interrupting type command and has not yet set the interrupt to signal the completion.
2. The chip is currently processing an Interrupting type command, a Pause command has been issued but the Paused bit in the Auxiliary Status Register has not been set.
3. The chip is connected as an Initiator, but the Target has not yet requested an Information Transfer.
4. The chip has completed a Transfer Info or Transfer Pad command and the Target has not requested additional information or has not changed the Information Phase.

In user states three and four, described above, the microprocessor must wait for a Bus Service, Disconnected, or Function Complete interrupt.

If an interrupting command is illegitimately issued in these states, no interrupt will occur for it, and it is likely that the current function will be altered.

5.4 COMMAND SUMMARY

Below is a summary that lists all commands. In the table the following abbreviations are used.

INT = INTERRUPTING D = DISCONNECTED I = CONNECTED AS AN INITIATOR
IMM = IMMEDIATE T = CONNECTED AS A TARGET

COMMAND CODE	COMMAND	TYPE	VALID STATES
00000	Chip Reset	IMM	D,I,T
00001	Disconnect	IMM	I,T
00010	Paused	IMM	D,T
00011	Set ATN	IMM	I
00100	Message Accepted	IMM	I
00101	Chip Disable	IMM	D,I,T
00110-00111	Reserved	IMM	
01000	Select w/ATN	INT	D
01001	Select w/o ATN	INT	D
01010	Reselect	INT	D
01011	Diagnostic	INT	D
01100	Receive Command	INT	T
01101	Receive Data	INT	T
01110	Receive Message Out	INT	T
01111	Receive Unspecified Info Out	INT	T
10000	Send Status	INT	T
10001	Send Data	INT	T
10010	Send Message In	INT	T
10011	Send Unspecified Info In	INT	T
10100	Transfer Info	INT	I
10101	Transfer Pad	INT	I
10110-11111	Reserved	INT	

5.5 COMMAND DEFINITIONS

5.5.1 CHIP RESET

Chip Reset immediately stops any chip operation and resets all registers, counters, etc. on the chip. It performs the same operation as the hardware "reset" input.

5.5.2 DISCONNECT

Upon receipt of this command, the chip immediately releases all SCSI bus signals and returns to a Disconnected idle state. For the Target role, this is the normal method of disconnecting from the bus when a transfer is complete. For the Initiator role, Disconnect may be used to release the bus signals as a result of a timeout condition. In this case, the chip ignores the Target and is left in the Disconnected state. For the Disconnected state, it is not valid to issue a Disconnect command. If issued, the chip will ignore this command.

5.5.3 PAUSE

Pause is an Immediate command that is valid in the Disconnected state or when logically connected to the bus as a Target device. Pause is not valid when connected as an Initiator.

When connected as a Target, the Pause command provides a means of halting a Send or Receive command without having to wait for the transfer to complete. When Pause is issued, it immediately sets a flag in the chip. Within one byte transfer cycle, the chip recognizes the flag, aborts the Send or Receive operation, and then sets the Paused status bit in the Auxiliary Status Register. At this time, the chip is still connected to the bus in the Target role, and it is waiting for another command.

The Pause command stops the Send or Receive command in an orderly manner leaving the Transfer Counter in a valid state that indicates the remaining number of bytes to be transferred. Also no REQ or ACK is asserted on the bus and no data is left in the chip waiting to be transferred. An operation that is paused may be resumed, if desired, simply by reloading the original command into the Command Register. (Note: after issuing the Pause while executing Send or Receive, it is necessary to continue transferring data with the chip (due to double-buffering) until the Paused status bit is set or an interrupt occurs.)

When in the disconnected state, Pause may be issued to abort a Select or Reselect command. After a Select or Reselect command is issued and before an interrupt occurs, a Pause command may be issued to abort the operation. The Pause command immediately sets an internal flag. If the chip has not yet won arbitration, it sets the Paused bit in the Auxiliary Status Register and waits in the disconnected state for another command. If the chip has won arbitration, it releases the bus by dropping the two ID bits with SELOUT on for a minimum of 100μs, checks for no BSYIN, and then releases the bus. After this procedure, it sets the Paused bit in the Auxiliary Status Register and waits for another command in the Disconnected state.

Since Pause is an Immediate command, it does not cause an interrupt. As previously noted, the chip sets the Paused status bit to indicate that is has been executed. If an interrupt-causing event occurs before the chip sees the pause flag set, the chip will set the interrupt. In this case, the Paused status bit is not set by the chip either before or after the interrupt. In all cases, an interrupt-causing event will take precedence over Pause. For example, in the Target role if ATN is on when Pause is issued, a Bus Service interrupt will occur and the Paused status bit will not be set.

If the Pause command is issued when the chip is Disconnected, the Paused status bit will be set by the chip, provided it has not already detected a Selection or Reselection.

5.5.4 SET ATN

The Set ATN command causes ATN to be asserted immediately if the chip is connected as an Initiator. This command is invalid and ignored if issued when the chip is Disconnected or is operating in a Target role. The ATN signal is de-asserted in a Message Out phase when the transfer count becomes zero or one byte has been transferred (in a one-byte transfer command) during the execution of a Transfer Info command.

The chip automatically sets ATN in two cases:

1. If a Select w/ATN command is issued and arbitration is won.
2. If a parity error is detected on an input byte during execution of a Transfer Info command.

5.5.5 MESSAGE ACCEPTED

The Message Accepted command is an Immediate command that is valid only when connected as an Initiator. It is used after a Transfer Info or Pad command (See pages 29,30 TRANSFER INFO and TRANSFER PAD) to indicate to the chip that ACK can be de-asserted for the last byte.

When an Initiator receives a message, a Transfer command is used. If the transfer is an input (I/O = 1) and the information is a message (MSG = 1, C/D = 1), the chip interrupts after receiving the last byte with a Function Complete interrupt. For this one special case, the chip also leaves ACK asserted on the bus. By interrupting and leaving ACK asserted, the chip gives the microprocessor a chance to interpret the message and set ATN, prior to ACK being de-asserted. This allows the chip to properly request a Message Out phase if the Initiator wants to send a "Reject Message" to the Target.

Message Accepted must always be issued after a Transfer Info for a Message In phase, whether or not Set ATN is issued, in order to have the chip de-assert ACK. If the Initiator wants to reject the message, Set ATN would be issued first followed by Message Accepted. If the message is not to be rejected, only Message Accepted is issued. (Note: until Message Accepted is issued, the Target will not send another REQ since ACK is still asserted.)

5.5.6 CHIP DISABLE

Chip Disable immediately stops all chip operations and logically disconnects it from the circuit. All outputs will be placed in a high impedance state and the chip will not respond to any commands (other than chip reset). The chip will also not respond to any activity on the SCSI bus. The only way to exit this condition is to activate the "reset" input or issue a Reset command.

5.5.7 SELECT w/ATN

This command causes the chip to attempt to select a Target. It may only be used if the microprocessor is in the Disconnected state. Any attempt to issue this command in another state will result in an Invalid Command interrupt. Before issuing this command, the microprocessor must load the Transfer Counter for a timeout on the Target's response. This value is computed according to the following formula:

Transfer Counter = Desired Timeout/ (1024 x Clock Period)

If the Transfer Counter is loaded with the value zero, the chip will wait indefinitely for a response from the Target being selected.

The microprocessor must also load the Destination ID Register with the three-bit code of the Target to be selected before issuing the Select w/ATN command.

When the chip detects the Select w/ATN command, it begins by attempting to arbitrate for control of the SCSI bus. If, at any time during arbitration the chip becomes selected or reselected, the Select w/ATN is aborted and forgotten and the chip will interrupt with one of the following conditions:

1. Selected
2. Selected and Bus Service
3. Reselected

If arbitration is won, the chip places the SCSI bus in the Selection phase with ATN asserted, and uses the Destination ID Register to identify the desired Target. At the same time, the chip begins a timer based on the value computed above. If the Target does not respond within the timeout period, the chip will disconnect from the bus and interrupt with the Disconnected flag set in the Interrupt Register. (Note: The microprocessor should never monitor the Transfer Counter Zero flag in the Auxiliary Status Register to determine when a timeout has occurred.) If the Target responds within the allotted time, the chip will interrupt with a Function Complete status. Control of the SCSI bus then belongs to the selected Target and after the interrupt status has been read, another interrupt may occur indicating either that the Target has disconnected or is requesting a transfer.

If the timeout is disabled and the Target does not respond, or if arbitration is not won, the only way to abort the Select w/ATN command is to issue the Pause command. After the Pause command is issued, it is still possible that the Function Complete or Disconnect interrupts may occur. This happens if one of the interrupts get set before the chip detects the Pause command, or if the Target responds while the chip is sequencing off the SCSI bus in a timeout condition. If the chip does not set either interrupt, it will set the Paused bit in the Auxiliary Status Register. If the microprocessor detects this bit after issuing the Pause command, then it is assured that the chip aborted the Select w/ATN command and no connection exists.

5.5.8 SELECT w/0 ATN

The Select w/o ATN is identical to the Select w/ATN command except that the ATN signal is not asserted during the Selection phase.

5.5.9 RESELECT

This command causes the chip to attempt to reselect an Initiator. It may only be used if the microprocessor is in the Disconnected state. Any attempt to issue this command in another state will result in an Invalid Command interrupt. Before issuing this command, the microprocessor must load the Transfer Counter for a timeout on the Initiator's response. This value is computed according to the following formula:

Transfer Counter = Desired Timeout/ (1024 x Clock Period)

If the Transfer Counter is loaded with the value zero, the chip will wait indefinitely for a response from the Initiator being reselected.

The microprocessor must also load the Destination ID Register with the three-bit code of the Initiator to be reselected before issuing the Reselect command.

When the chip detects the Reselect command, it begins by attempting to arbitrate for control of the SCSI bus. If, at any time during arbitration, the chip becomes selected or reselected, the Reselect is aborted and forgotten and the chip will interrupt with one of the following conditions:

1. Selected
2. Selected and Bus Service
3. Reselected

If arbitration is won, the chip places the SCSI bus in the Reselection phase using the Destination ID Register to identify the desired Initiator. At the same time, the chip begins a timer based on the value computed above. If the Initiator does not respond within the timeout period, the chip will disconnect from the bus and interrupt with the Disconnected flag set in the Interrupt Register. (Note: The microprocessor should never monitor the Transfer Counter Zero flag in the Auxiliary Status Register to determine when a timeout has occurred.) If the Initiator responds within the allotted time, the chip will interrupt with a Function Complete status. The chip (acting as the Target) is then in control of the SCSI bus, and waits for the Interrupt Register to be read by the microprocessor. After it has been read, the chip waits for a command from the microprocessor or ATN from the Initiator. If the ATN occurs, the chip will set the Bus Service interrupt. This interrupt may happen immediately after a command has been issued due to internal timing. In this case, the chip waits for the Interrupt Register to be read and the command is ignored. The chip then waits for a new command.

If the timeout is disabled and the Initiator does not respond, or if arbitration is not won, the only way to abort the Reselect command is to issue the Pause command. After the Pause command is issued, it is still possible that the Function Complete or Disconnected interrupts may occur. This happens if one of the interrupts get set before the chip detects the Pause command, or if the Initiator responds while the chip is sequencing off the SCSI bus in a timeout condition. If the chip does not set either interrupt, it will set the Paused bit in the Auxiliary Status Register. If the microprocessor detects this bit after issuing the Pause command, then it is assured that the chip aborted the Reselect command and no connection exists.

5.5.10 DIAGNOSTIC (DATA TURNAROUND)

This Interrupting command causes the chip to attempt to turn a data byte around through its internal data paths. When the command is loaded into the Command Register the Data Register Full bit is reset. The microprocessor then writes one byte into the Data Register. The chip moves the byte to another register and compares the contents of the Data Register. The byte is then moved to a third register (the SCSI output register) and good parity is generated if bit 6 of the command is off (0); bad parity is generated if bit 6 is on (1). Finally, the chip moves the byte back to the Data Register and compares it with the contents of the second register. Based on these comparisons and parity checking, the chip stores a result into the Diagnostic Status Register and sets the Function Complete interrupt. After reading the Interrupt Register, the microprocessor should make sure the Data Register Full bit is on (1) and read the contents of the Data Register. If the Data Register Full bit is not on (0), then an error has occurred. The following is a list of codes which are loaded into bits 6-3 of the Diagnostic Status Register as a result of this command.

BIT 6543	RESULT
0001	Data Miscompare (INITIAL)
0010	Data Miscompare (FINAL)
0011	Good Parity Detected
0100	Bad Parity Detected

5.5.11 RECEIVE COMMANDS

The Receive commands are Interrupting commands that are valid only when connected as a Target device. They are used by the Target to receive commands, data, and message information from an Initiator.

The Receive commands transfer data; therefore, the Single Byte Transfer and DMA mode bits in the Command Register are valid for these commands. If the Single Byte Transfer bit is off (0), the Transfer Counter must be loaded before a Receive command is issued to the chip. In this case, the chip uses the Transfer Counter to determine the number of bytes to receive.

When a Receive command is issued, the chip immediately resets the Data Register Full bit in the Auxiliary Status Register. The chip then drives the I/O, C/D, and MSG outputs for the proper information phase as follows.

COMMAND NAME	I/O	C/D	MSG
Receive Command	0	1	0
Receive Data	0	0	0
Receive Message Out	0	1	1
Receive Unspecified Info Out	0	0	1

The chip then proceeds to request and receive the specified number of information bytes. The DMA mode bit in the Command Register determines how the chip transfers these bytes from its Data Register to the microprocessor.

When a Receive command is terminated, the chip generates an interrupt. The following two events can cause termination:

1. The operation completes successfully; the Transfer Counter is zero. This event results in a Function Complete interrupt with the Parity Error bit in the Auxiliary Status Register off (0). If the initiator activated ATN during the operation, the Bus Service bit will also be on.
2. A Parity Error occurs. The last byte transferred is the byte that caused the error. This event causes a Function Complete interrupt with the Parity Error bit in the Auxiliary Status Register on (1). If the Initiator activated ATN during the operation, the Bus Service bit will also be on.

After any of the interrupts, the chip is always left in the connected Target state. The Transfer Counter indicates the number of bytes remaining to be transferred (zero if completed successfully, and the Data Register is empty (the last byte received is sent to the microprocessor). Also, ACK and REQ are inactive on the bus.

(Note: if a Bus Service interrupt alone occurs after issuing a Receive command, the Initiator activated ATN before the chip began executing the command. In this case, the command is ignored by the chip.)

A Receive command may be stopped prior to an interrupt causing event by issuing a Pause command. Operation of the Pause command is explained in an earlier section (See page 22, PAUSE). In the event the Initiator does not respond, or stops responding, the chip is left in a state where it cannot respond to a Pause command. For this case, a Disconnect command can be used to abort the command and the connection. The Disconnect command is explained in an earlier section (See page 22, DISCONNECT).

5.5.12 SEND COMMANDS

The Send commands are Interrupting commands that are valid only when connected to the bus in the Target role. They are used by a Target to send status, data, and message information to an Initiator.

The Send commands transfer data, and therefore, the Single Byte Transfer and DMA mode bits in the Command Register are valid for these commands. If the Single Byte Transfer bit is off (0), the Transfer Counter must be loaded before a Send command is issued to the chip. In this case, the chip uses the Transfer Counter to determine the number of bytes to send.

When a Send command is issued, the chip immediately resets the Data Register Full bit in the Auxiliary Status Register. Therefore, the first byte of data for the transfer cannot be put into the Data Register until after a Send command is loaded into the Command Register.

In executing a Send command, the chip drives the I/O, C/D, and MSG outputs for the proper information phase. These lines are logically driven for each Send command as shown below.

COMMAND NAME	I/O	C/D	MSG
Send Status	1	1	0
Send Data	1	0	0
Send Message In	1	1	1
Send Unspecified Info In	1	0	1

After resetting Data Register Full and driving I/O, C/D, and MSG, the chip then proceeds to monitor Data Register Full, take the data from the Data Register, and send it to the Initiator. The DMA mode bit in the Command Register specifies how the data is loaded into the chip.

After interrupting, the chip is left in the connected Target state, and ACK and REQ are inactive on the bus. When the transfer is complete, the chip interrupts with a Function Complete Interrupt. If the Initiator activated ATN during the transfer, a Bus Service bit will also be set by the chip.

(Note: if a Bus Service interrupt alone occurs after issuing a Send command, the Initiator activated ATN before the chip began executing the command. In this case, the command is ignored by the chip.)

A Send command may be stopped prior to an interrupt causing event by issuing a Pause command. Operation of the Pause command is explained in an earlier section (See page 22, PAUSE). In the event the Initiator does not, or stops responding, the chip is left in a state where it cannot respond to a Pause command. For this case, a Disconnect command can be used to abort the command and the connection. The Disconnect command is explained in an earlier section (See page 22, DISCONNECT).

5.5.13 TRANSFER INFO

The Transfer Info command is an Interrupting command that is valid only when connected to the bus in the Initiator role. It is used by the Initiator for all information transfers across the SCSI bus.

A Transfer Info command is issued by an Initiator in response to a Bus Service interrupt. The Bus Service interrupt, as explained in a previous section (See page 14, INTERRUPT REGISTER), is received by the connected Initiator upon the following conditions: receiving the first REQ from a Target, a previous command has completed and the Target changes phases, the Target changes phases before termination, or when a previous command has completed and the Target is requesting more information. It is not valid to issue a Transfer Info command without having a Bus Service interrupt, because the Target requests and controls all transfers. The chip will only permit one Transfer Info or Transfer Pad per Bus Service interrupt.

After an Initiator receives a Bus Service interrupt, and prior to issuing a Transfer Info command, the I/O, C/D, and MSG bits from the Auxiliary Status Register (read prior to reading the Interrupt) should be examined to determine the type of information phase and the direction of transfer requested by the Target. The Initiator then prepares for the transfer. If the Single Byte Transfer bit is not going to be set in the Command Register, the Transfer Counter must be loaded prior to issuing the Transfer Info command. This is done in order to specify to the chip the maximum number of bytes to be transferred.

When a Transfer Info is issued, the chip immediately resets the Data Register Full bit in the Auxiliary Status Register. For this reason, the first byte of data for an output operation cannot be loaded into the Data Register until after the command is loaded into the Command Register. The chip then proceeds with the transfer, expecting data to be read from (input), or written to (output), its Data Register as indicated by the DMA Mode bit in the Command Register. The chip automatically detects the direction of the transfer from the I/O bit which is stored in the Auxiliary Status Register.

The chip continues a transfer until an interrupt causing event occurs. The following four events will cause the chip to terminate and interrupt.

1. The maximum number of bytes specified have been transferred and the Target activated REQ or the Information Phase changed. This event results in a Bus Service Interrupt. Either single byte transfer was specified or the Transfer Counter is zero as indicated by a bit in the Auxiliary Status Register. The Target may or may not have changed the information phase type. The I/O, C/D, and MSG bits in the Auxiliary Status Register need to be examined at the time of the interrupt to determine what phase the Target is requesting.

   (Note: due to early notification of the phase change, a phase may be selected spuriously and not transfer any data. The microprocessor should not consider this an error condition.)

2. The Target changes the information phase type before the maximum number of bytes are transferred. This event also causes a Bus Service interrupt. The new information phase may be determined by examining the I/O, C/D, and MSG bits in the Auxiliary Status Register. The Transfer Counter may be read at the time of the interrupt to determine the number of bytes remaining to be transferred. When this interrupt occurs for an output transfer, the chip may take one more byte from the microprocessor than it transfers, because of pre-fetching. However, the Transfer Counter still reflects the number of bytes remaining to be transferred.

3. The Target releases the bus by dropping BSY. This event results in a Disconnected interrupt. Following this interrupt, the chip is no longer in the Initiator role. It now remains in the Disconnected state.

4. The last byte of a Message Input phase has been received. This event results in a Function Complete interrupt. For this case, ACK is left active on the bus to allow the microprocessor to Set ATN for the purpose of rejecting the message. After this interrupt is received and a Set ATN is issued (if desired), a Message Accepted must be issued to turn off ACK for the last byte of the Message In phase.

For input transfers (I/O = 1), the chip checks parity for each byte received if the Parity Enable bit in the Control Register is on. When checking parity and the parity error occurs, the chip activates ATN prior to deactivating ACK for the byte that causes the error. It also turns on the Parity Error bit in the Auxiliary Status Register. The parity error, however, does not result in an interrupt. The chip waits for one of the four events listed above before interrupting. Therefore, the Parity Error bit should be examined when servicing any interrupt after issuing Transfer Info command for an input transfer.

If ATN is asserted by the chip, either because of a parity error or because a SET ATN command is issued, the ATN will remain asserted until the end of the connection, or until a Message Out is transferred. Therefore, during each cycle of a Transfer Info operation for output, the chip checks for a message phase (C/D = 1, MSG = 1) and also either a single byte transfer or the Transfer Counter set at zero. If these conditions exist, the chip turns off ATN prior to activating ACK for the last byte of the message.

As previously stated, a Transfer Info normally terminates with an interrupt. If a Transfer Info command must be aborted, possibly because of a timeout violation, either a Chip Reset or a Disconnect command can be used. It is noted, however, that although these commands will force the chip into a disconnected state, the Target device is left on the bus. A SCSI bus reset, which is not a chip function, is the only way an Initiator can force a Target to disconnect.

5.5.14 TRANSFER PAD

The Transfer Pad command is an Interrupting command that is valid only when connected to the bus as an Initiator. It is similiar to the Transfer Info command except that the data transfer between the chip and the microprocessor bus will be different.

Transfer Pad can be used by an Initiator to continue handshaking with a Target without giving data to, or taking data from, the chip. This may be useful if the Target requests an invalid Information Transfer Phase. The chip operates in the same manner as it does for a Transfer Info command, except that for output transfers it takes only one byte of data from the microprocessor and sends the same byte repeatedly until the transfer terminates. For input transfers, it accepts data from the SCSI bus but does not check parity or send it to the microprocessor. Though data is not exchanged with the microprocessor bus, the Transfer Counter is still used by the chip so that a maximum number of pad bytes can be specified.

Protocol for using a Transfer Pad command is the same as the Transfer Info except that the DMA Mode bit has significance only for output transfers. The Transfer Pad terminates because of the same four events that cause a Transfer Info command to terminate. Also, similar to the Transfer Info command, Chip Reset and Disconnect can be used to abort the command.

SECTION 6
BUS INITIATED FUNCTIONS

6.1 SELECTION

If the Select Enable bit in the Control Register is on, the chip may be selected by another SCSI device to be a TARGET for an I/O operation. Selection occurs in the chip only if all the following conditions exist: SELOUT = 0, BSYIN = 0, SELIN = 1, I/O = 0, the chip's ID bit is asserted by the selecting device on the data bus, no more than one other ID bit (the Initiator's) is asserted on the data bus and data bus parity is good.

When all of these conditions exist, the chip is selected. It then encodes the Initiator's ID and loads it into bits 2-0 of the Source ID Register. The chip also detects whether or not the Initiator asserted its ID during selection, and either sets or resets the ID Valid bit in the Source ID Register.

The chip then asserts BSYOUT, waits for SELIN to turn off, and proceeds to take one of the following actions as a result of being selected:

1. If ATN is not asserted by the Initiator during selection, the chip generates a Selected interrupt indicating that the chip is connected as a Target.
2. If ATN is asserted, the chip simultaneously generates Selected, and Bus Service interrupts, indicating that the chip is connected as a Target and ATN is asserted.

6.2 RESELECTION

If the Reselect Enable bit in the Control Register is on, the chip may be reselected by a SCSI Target device. Reselection occurs only if SELOUT = 0, SELIN = 1, BSYIN = 0, I/O = 1, the chip's ID bit and the Target's ID bit are asserted on the data bus, no other ID bits are asserted, and data bus parity is good.

When all of these conditions exist, the chip is reselected. It then encodes the Target's ID and loads it into the Source ID Register. The chip also sets the ID Valid bit in the Source ID Register.

The chip then asserts BSYOUT and waits for SELIN to be released by the Target. When the chip detects SELIN = 0, it de-asserts BSYOUT and then generates a Reselected interrupt.

Reselection is now complete and the chip is in the connected Initiator state.

SECTION 7
INITIALIZATION

The SCSI device may be initialized by asserting RST for a period of at least 100ns, or by issuing a Chip Reset command to the device. The NCR 5385E will respond to the RST pulse or the Chip Reset command, by immediately disconnecting from the SCSI bus, initializing all storage elements and executing an internal self-diagnostic program. The self-diagnostic is explained in a previous section (See page 17, Diagnostic Status Register). The following table lists the status of all registers after the initialization procedure.

	7	6	5	4	3	2	1	0
Data Register	x	x	x	x	x	x	x	x
Command Register	0	0	0	0	0	0	0	0
Control Register	0	0	0	0	0	0	0	0
Destination ID Register	0	0	0	0	0	0	0	0
Auxiliary Status Register	0	0	x	x	x	0	1	0
ID Register	0	0	0	0	0	x	x	x
Interrupt Register	0	0	0	0	0	0	0	0
Source Register	0	0	0	0	0	1	1	1
Diagnostic Status Register	1	x	x	x	x	x	x	x
Transfer Counter (MSB)	0	0	0	0	0	0	0	0
Transfer Counter (2nd)	0	0	0	0	0	0	0	0
Transfer Counter (LSB)	0	0	0	0	0	0	0	0

x = Unknown

TABLE 7.1 REGISTER INITIALIZATION

The controlling processor should loop on reading the Diagnostic Status Register until the Self-Diagnostic Complete bit (bit 7) is on (1). This should take approximately 350 clock cycles after reset occurs. The processor should then check the remaining bits in this register for all zeroes (no errors), and then load the Control Register enabling the proper bits to begin operation. The SCSI Protocol Controller is now connected to the SCSI bus in a disconnected state. It is ready to receive commands from the controlling processor or respond to (re) selection attempts.

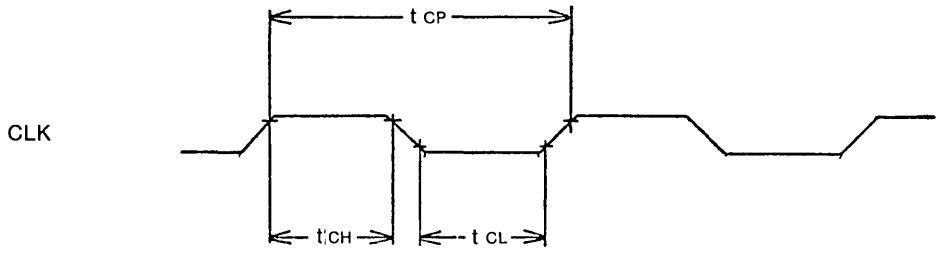
SECTION 8
EXTERNAL CHIP TIMING

Timing requirements must be over the operating temperature (0-70°C) and voltage (4.75 to 5.25V) ranges. Loading for all output signals, except $\overline{SBEN}$, is assumed to be four low-power Schottky inputs, including 50 pF capacitance. Loading for $\overline{SBEN}$ is assumed to be ten low-power Schottky inputs, including 100 pF capacitance.

8.1 MICROPROCESSOR INTERFACE

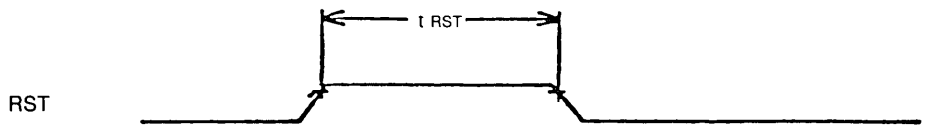
8.1.1 CLK

NAME	DESCRIPTION	MIN	MAX	UNITS
t_{CP}	Clock Period	100	200	ns
t_{CH}	Clock High	.45 t_{CP}	.55 t_{CP}	
t_{CL}	Clock Low	.45 t_{CP}	.55 t_{CP}	



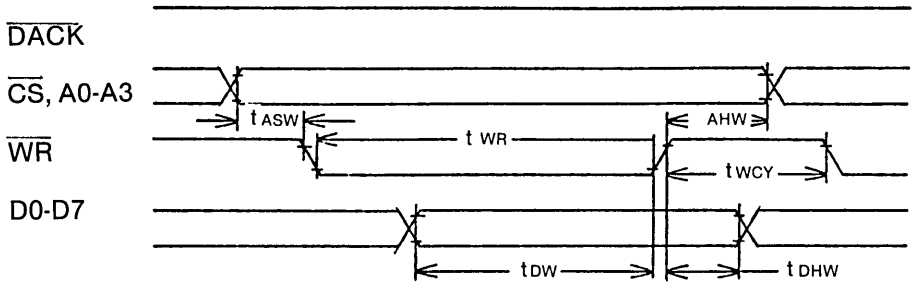
8.1.2 RESET

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{RST}	Reset Pulse Width	100			ns



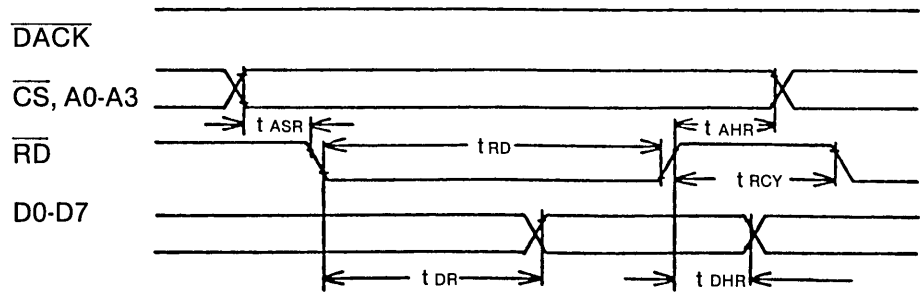
8.1.3 MPU WRITE

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{ASW}	Address Set-up Time	0			ns
t_{WR}	$\overline{WR}$ Pulse Width	95			ns
t_{DW}	Data-to $\overline{WR}$ High	50			ns
t_{AHW}	Address Hold Time	0			ns
t_{DHW}	Data Hold Time	20			ns
t_{WCY}	$\overline{WR}$ Off to $\overline{WR}$ or $\overline{RD}$ On	125			ns



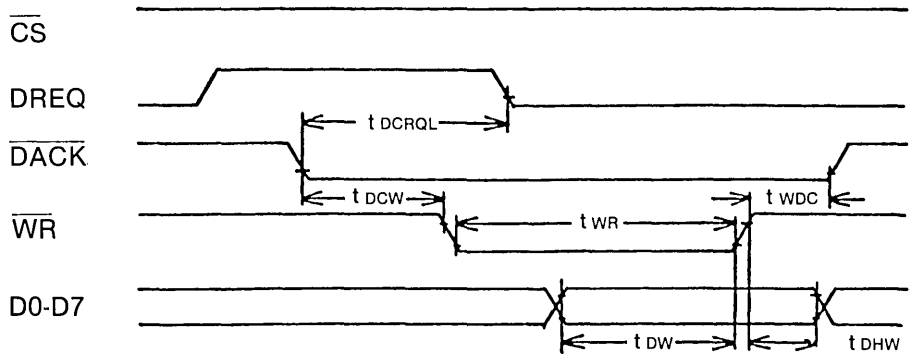
8.1.4 MPU READ

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{ASR}	Address Set-up Time to $\overline{RD}$	0			ns
t_{RD}	$\overline{RD}$ Pulse Width	125			ns
t_{DR}	$\overline{RD}$ to Data			90	ns
t_{AHR}	Address Hold Time	0			ns
t_{DHR}	Data Hold Time	10			ns
t_{RCY}	$\overline{RD}$ Off to $\overline{WR}$ or $\overline{RD}$ On	125			ns



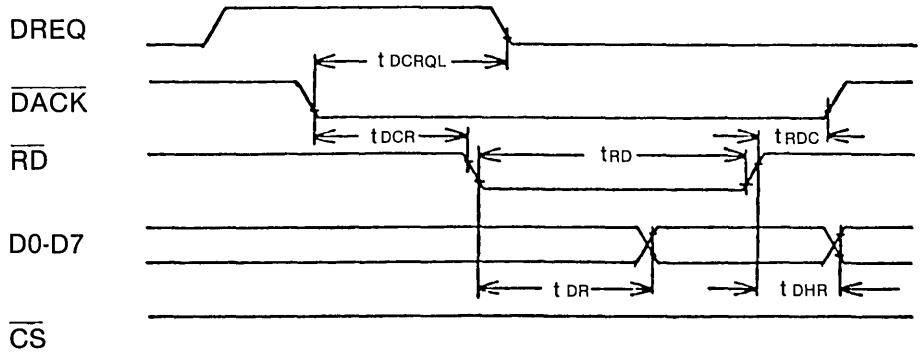
8.1.5 DMA WRITE

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{DCRQL}	$\overline{DACK}$ to DREQ Low	0		40	ns
t_{DCW}	$\overline{DACK}$ to $\overline{WR}$	0			ns
t_{WR}	$\overline{WR}$ Pulse Width	95			ns
t_{WDC}	$\overline{WR}$ High to $\overline{DACK}$ High	0			ns
t_{DHW}	Data Hold Time	20			ns
t_{DW}	Data to $\overline{WR}$ High	50			ns



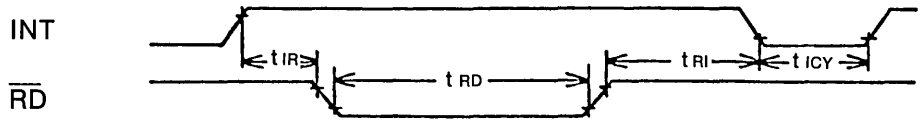
8.1.6 DMA READ

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{DCRQL}	$\overline{DACK}$ to DREQ Low	0		40	ns
t_{DCR}	$\overline{DACK}$ to $\overline{RD}$	0			ns
t_{RD}	$\overline{RD}$ Pulse Width	95			ns
t_{RDC}	$\overline{RD}$ High to $\overline{DACK}$ High	0			ns
t_{DHR}	Data Hold Time	10			ns
t_{DR}	$\overline{RD}$ to Data			80	ns



8.1.7 INTERRUPT

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{IR}	INT to $\overline{RD}$	0			ns
t_{RD}	$\overline{RD}$ Pulse Width	95			ns
t_{RI}	$\overline{RD}$ High to INT Low			125	ns
t_{ICY}	INT Off to INT On	125			ns



8.2 SCSI INTERFACE

8.2.1 SELECTION (INITIATOR)

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
^tBF	Bus Free	385			ns
$^tBIA(5)$	BSYIN low to ARB high	1.2		2.6	us
^tSLA	SELOUT high to ARB low & ID bit Disabled	3.2			us
^tBIBO (5)	BSYIN low to BSYOUT high	1.2		2.8	us
^tBCD	Bus Clear Delay			225	ns
^tAD	Arbitration Delay	3.0			us
^tPC	Priority check to SELOUT	0			ns
^tBID (5)	BSYIN low to ID bit high	1.2		2.9	us
^tADV	Arbitration Data Valid to Priority Check	0			ns
^tSI	SELOUT to IGS	2.0			us
^tIDBL	Target ID high to BSYOUT low	1.1			us
^tBOBI	BSYOUT low to BSYIN low	0		400	ns
^tBSL	BSYIN high to SELOUT low	800			ns
^tDID	SBEN active to Bus enabled	150			ns

NOTES:

1. The chip ensures that the bus remains free (BSYIN and SELIN inactive) for ^tBF before attempting arbitration.

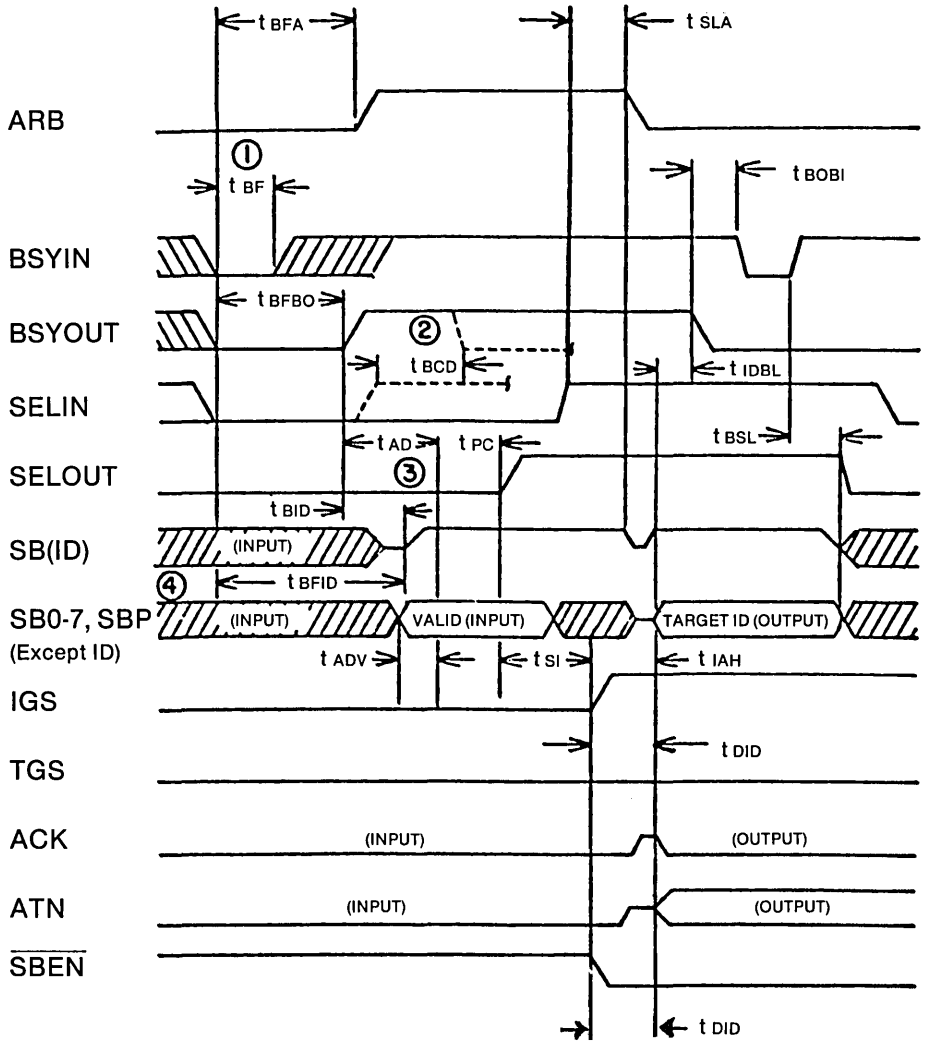
2. If SELIN becomes active at any time during arbitration, the chip must deassert BSYOUT within ^tBCD.

3. The chip waits (^tAD), and then checks to see if arbitration is won (^tPC). The chip then asserts SELOUT if arbitration is won.

4. One of the data bits is assigned as an ID bit by the IDO-ID2 signals. During Bus Free, the chip places all of the data bits, including ID, in a high impedance state. During arbitration the chip enables its ID bit and drives it high, but the remainder of the data bits remain in the high impedance state for reading.

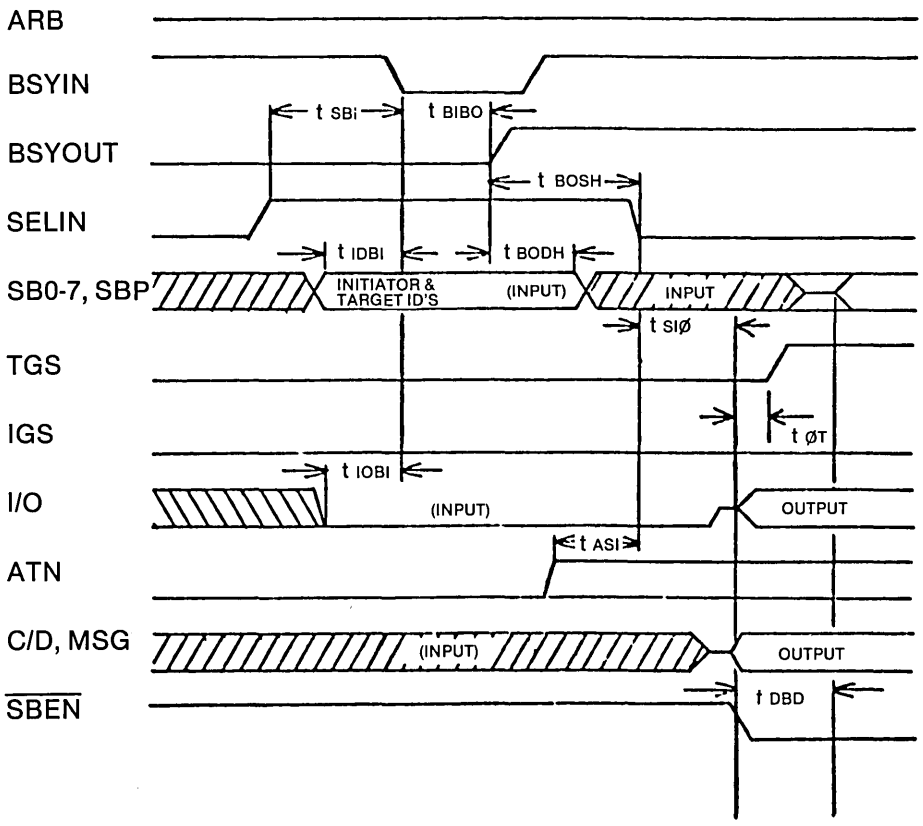
5. To verify these timings in a test environment, the user must allow a minimum of 45 clock cycles after the select command has been issued before the device begins to check for BSYIN low.

8.2.1 SELECTION (INITIATOR)



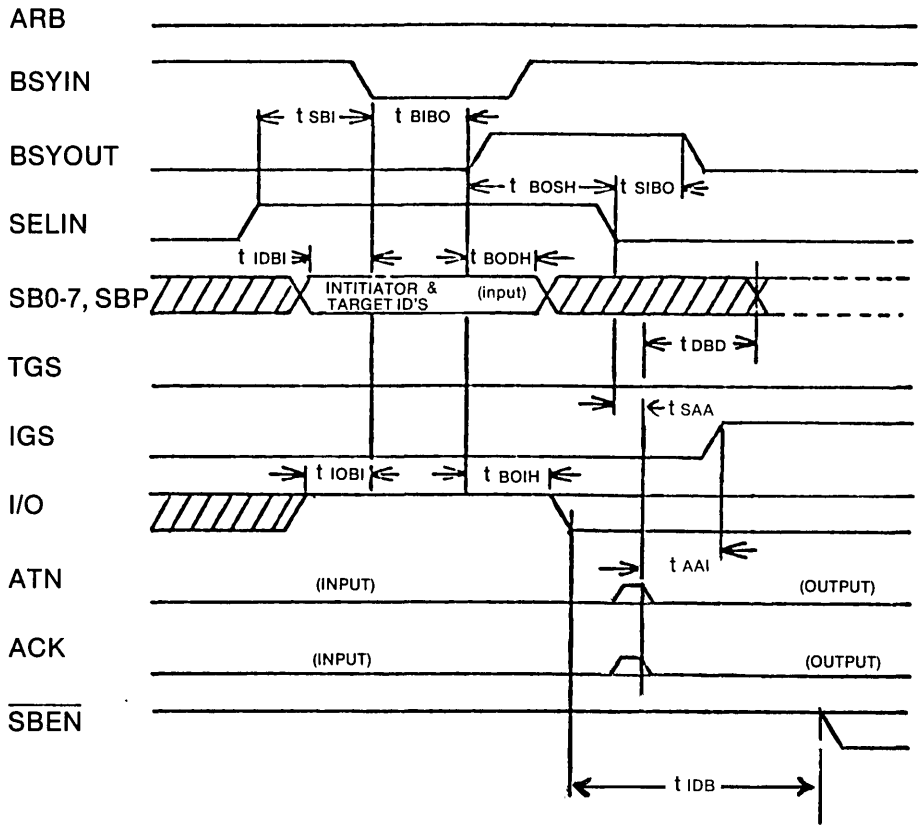
8.2.2 SELECTION (TARGET)

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
T_{SBI}	SELIN high to BSYIN low	50			ns
t_{IDBI}	ID's valid to BSYIN low	0			ns
t_{IOBI}	I/O low to BSYIN low	0			ns
t_{BIBO}	BSYIN low to BSYOUT high	0		2.0	μs
t_{BODH}	BSYOUT high Data Hold	0			ns
t_{BOSH}	BSYOUT high SELIN Hold	0			ns
t_{ASI}	ATN high to SELIN low	0			ns
t_{SIO}	SELIN low to Phase signals Enabled	150			ns
t_{OT}	Phase signals enabled to TGS High	150			ns
t_{DBD}	SBEN low to Data Bus Enabled	150			ns



8.2.3 RESELECTION (INITIATOR)

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{SBI}	SELIN high to BSYIN low	50			ns
t_{IDBI}	ID's valid to BSYIN low	0			ns
t_{IOBI}	I/O high to BSYIN low	0			ns
t_{BIBO}	BSYIN low to BSYOUT high	0		2.0	μs
t_{BODH}	BSYOUT high Data Hold	0			ns
t_{BOSH}	BSYOUT high SELIN Hold	0			ns
t_{BOIH}	BSYOUT high I/O hold	0			ns
t_{SIBO}	SELIN low to BSYOUT low	0			ns
t_{SAA}	SELIN low to ACK & ATN enabled	750			ns
t_{AAI}	ACK & ATN enabled to IGS high	150			ns
t_{IDB}	I/O low to SBEN low	0			ns
t_{DBD}	SBEN low to Data Bus Enabled	150			ns



8.2.4 RESELECTION (TARGET)

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{BF}	Bus Free	385			ns
t_{BIA} (5)	BSYIN low to ARB high	1.2		2.6	us
t_{SLA}	SELOUT high to ARB low & ID bit Disabled	3.2			us
t_{BIBO} (5)	BSYIN low to BSYOUT high	1.2		2.8	us
t_{BCD}	Bus Clear Delay			225	ns
t_{AD}	Arbitration Delay	3.0			us
t_{PC}	Priority check to SELOUT	0			ns
t_{BID} (5)	BSYIN low to ID bit high	1.2		2.9	us
t_{ADV}	Arbitration Data Valid to Priority Check	0			ns
t_{SO}	SELOUT Phase signals Enabled & SBEN Low	2.4			us
t_{OT}	Phase Signals Enabled to TGS High	150			ns
t_{DID}	SBEN low to Bus Enabled	150			ns
t_{IDBL}	INITIATOR ID high to BSYOUT low	2.7			us
t_{BOBI}	BSYOUT low to BSYIN	0		400	ns
t_{BIBO}	BSYIN high to BSYOUT high	0.7		2.0	us
t_{BSL}	BSYOUT high to SELOUT low	450			ns

NOTES:

1. The chip ensures that the bus remains free (BSYIN and SELIN inactive) for TBF before attempting arbitration.

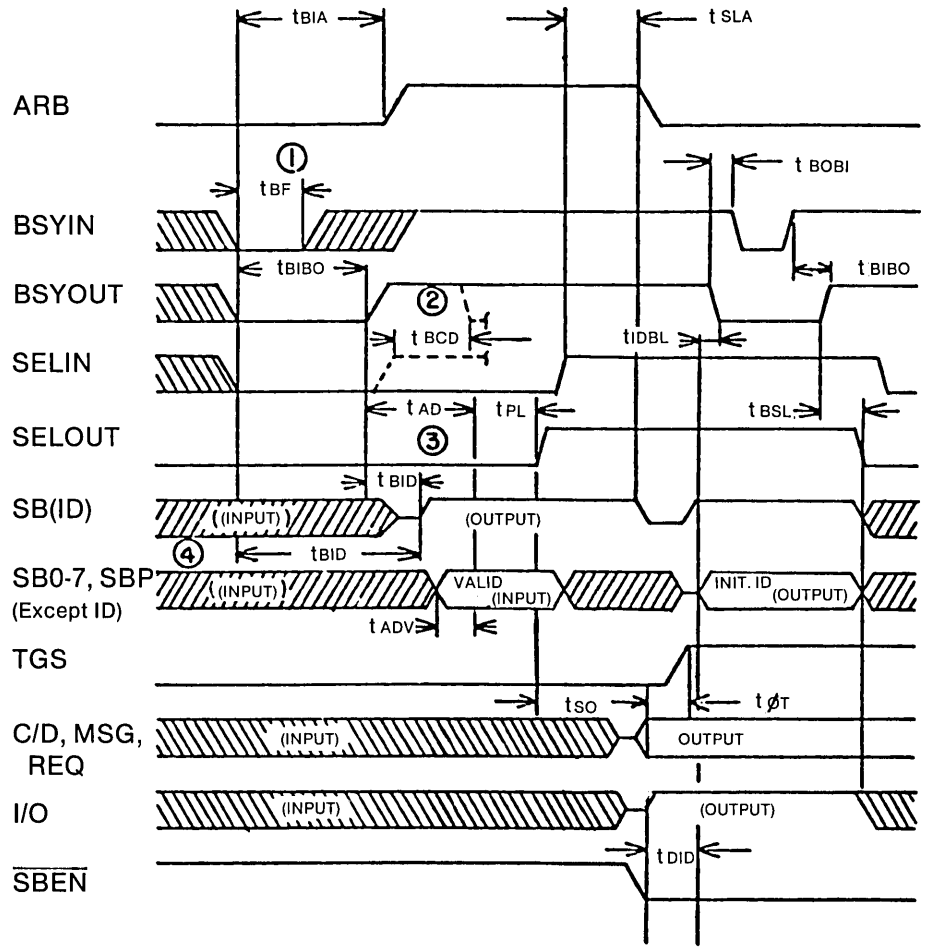
2. If SELIN becomes active at any time during arbitration, the chip must deassert BSYOUT within t_{BCD}.

3. The chip waits (t_{AD}), and then checks to see if arbitration is won (t_{PC}). The chip then asserts SELOUT if arbitration is won.

4. One of the data bits is assigned as an ID bit by the IDO-ID2 signals. During Bus Free, the chip places all of the data bits, including ID, in a high impedance state. During arbitration the chip enables its ID bit and drives it high, but the remainder of the data bits remain in the high impedance state for reading.

5. To verify these timings in a test environment, the user must allow a minimum of 45 clock cycles after the select command has been issued before the device begins to check for BSYIN low.

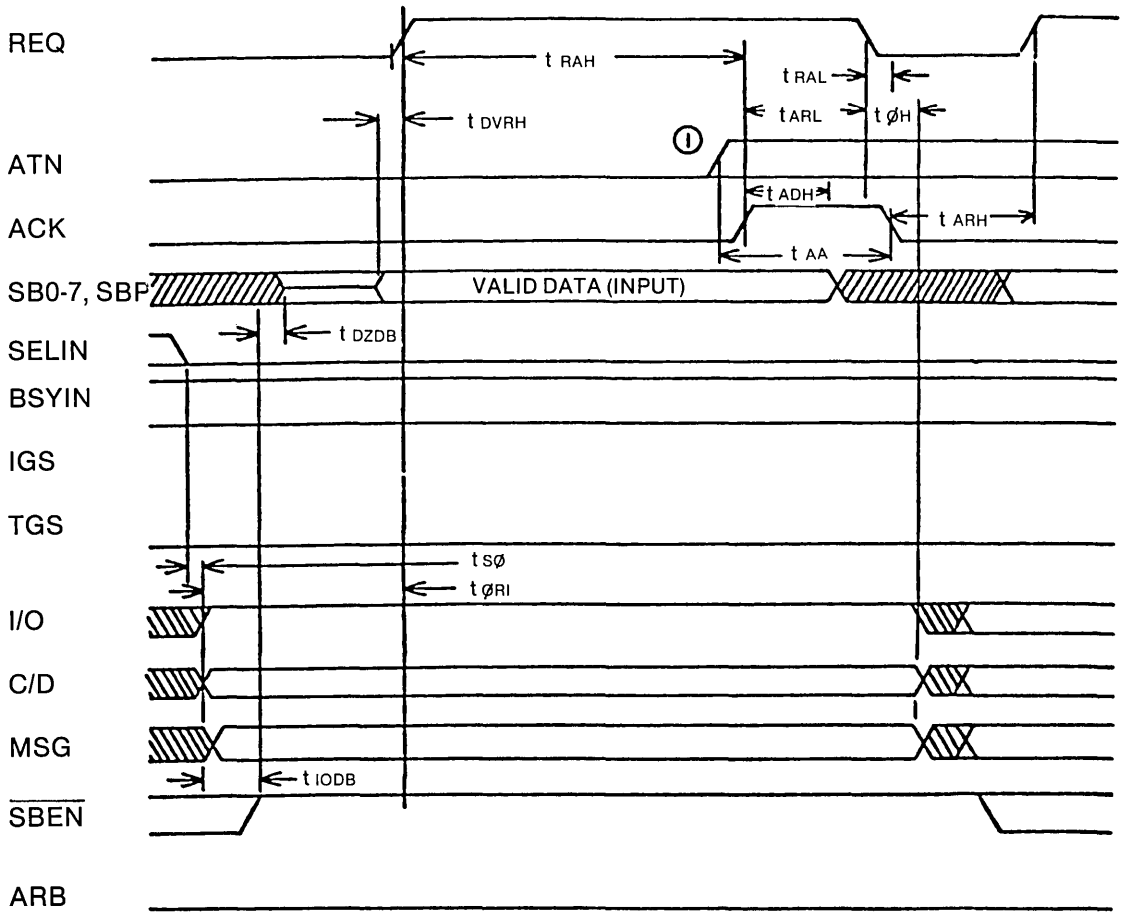
8.2.4 RESELECTION (TARGET)



8.2.5 INFORMATION TRANSFER PHASE INPUT (INITIATOR)

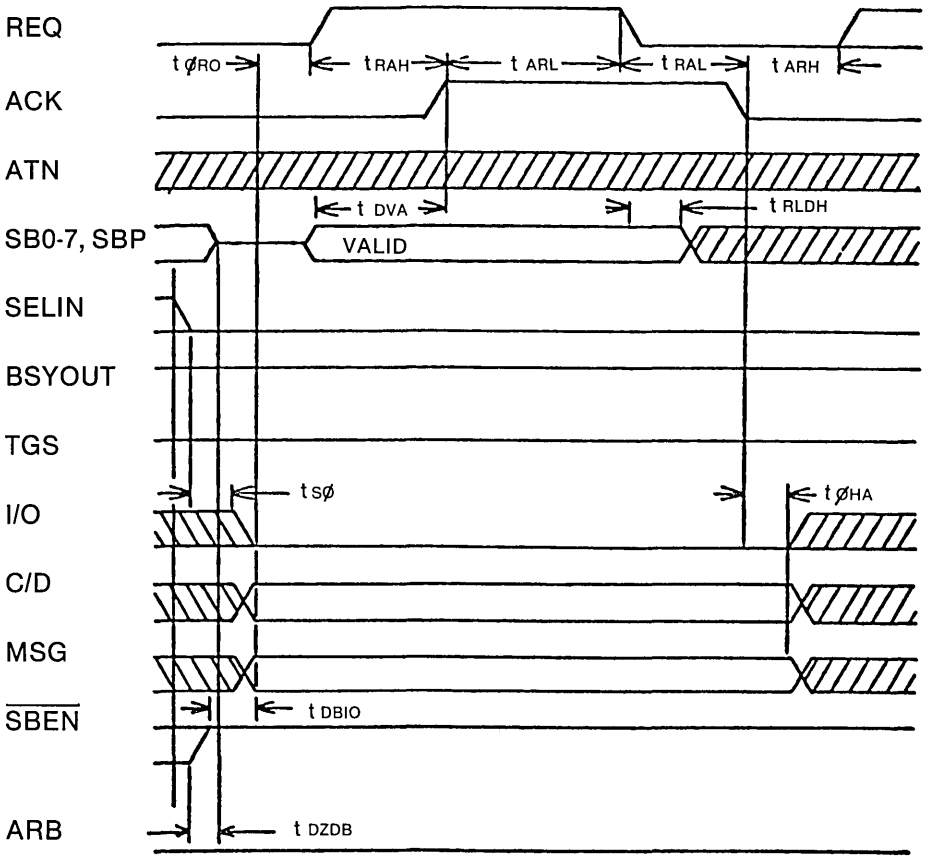
NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{DVRH}	Data Valid to REQ high	0			ns
$t_{\emptyset RI}$	Phase Valid to REQ high	100			ns
t_{RAH}	REQ high to ACK high	0			ns
t_{RAL}	REQ low to ACK low	0			ns
t_{AA}	ATN high to ACK low	100			ns
$t_{S\emptyset}$	SELIN low to Phase change	0			ns
$t_{\emptyset H}$	Phase hold from ACK low	20			ns
t_{ADH}	Data hold from ACK high	0			ns
t_{ARL}	ACK high to REQ low	35			ns
t_{IODB}	I/O high to SBEN high			50	ns
t_{DZDB}	Data Bus disable from SBEN high			10	ns
t_{ARH}	ACK Low to REQ High	35			ns

NOTE 1: If the chip detects a parity error it must assert ATN at least t_{AA} before it de-asserts ACK.



8.2.6 INFORMATION TRANSFER PHASE INPUT (TARGET)

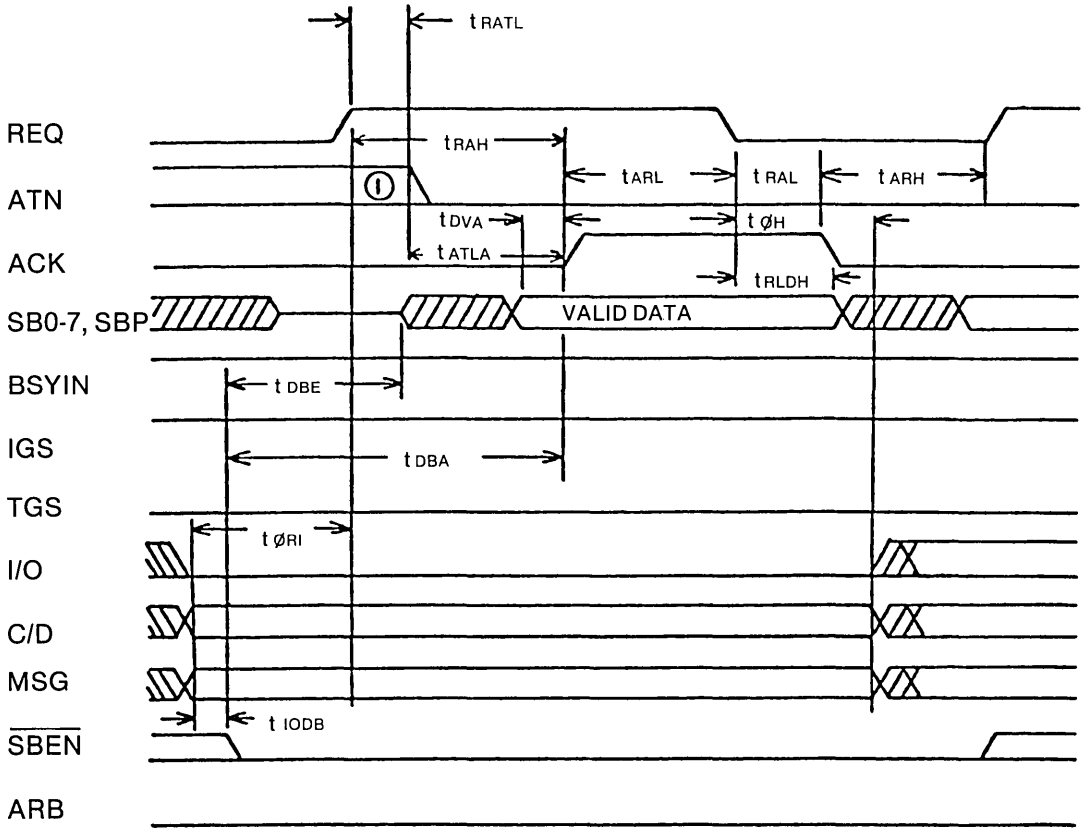
NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
$t_{S\emptyset}$	SELIN low to Phase Change	0			ns
$t_{\emptyset RO}$	Phase Change to REQ out	500			ns
t_{RAH}	REQ high to ACK high	35			ns
t_{ARL}	ACK high to REQ low	0			ns
t_{DVA}	Data Valid to ACK high	0			ns
t_{RAL}	REQ low to ACK low	35			ns
t_{ARH}	ACK low to REQ high	0			ns
t_{RLDH}	REQ low Data Hold	0			ns
$t_{\emptyset HA}$	Phase Hold from ACK low	0			ns
t_{DBIO}	SBEN high to I/O low	0			ns
t_{DZDB}	Data Bus disable to SBEN high	0			ns



8.2.7 INFORMATION TRANSFER PHASE OUTPUT (INITIATOR)

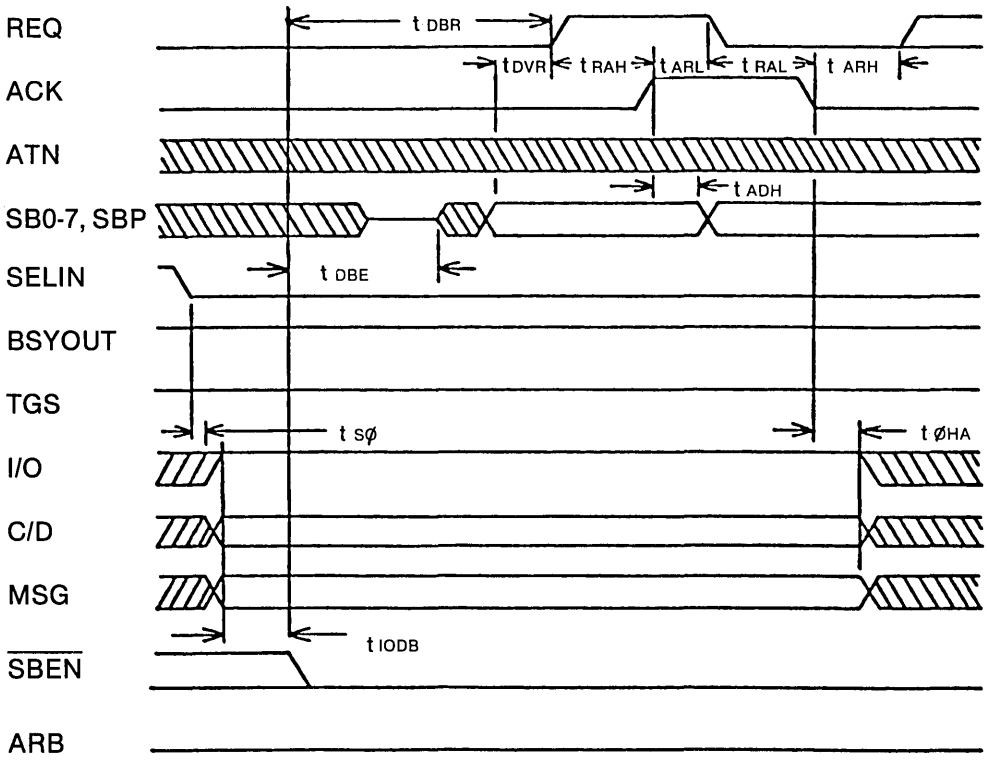
NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
$t_{\phi RI}$	Phase Valid to REQ high	100			ns
t_{RAH}	REQ high to ACK high	35			ns
t_{RAL}	REQ low to ACK low	0			ns
t_{DVA}	Data Valid to ACK high	100			ns
t_{RLDH}	REQ low Data hold	0			ns
$t_{\phi H}$	Phase hold from ACK low	20			ns
t_{ARL}	ACK high to REQ low	0			ns
t_{IODB}	I/O low to $\overline{SBEN}$ low	0			ns
t_{DBE}	$\overline{SBEN}$ low to Data Bus Enable	85			ns
t_{DBA}	$\overline{SBEN}$ low to ACK high	185			ns
t_{RATL}	REQ High to ATN low	0			ns
t_{ATLA}	ATN Low to ACK High	25			ns
t_{ARH}	ACK Low to REQ High	35			ns

NOTE 1: ATN is only de-asserted in this manner during the last byte of a Message Out Phase.



8.2.8 INFORMATION TRANSFER PHASE OUTPUT (TARGET)

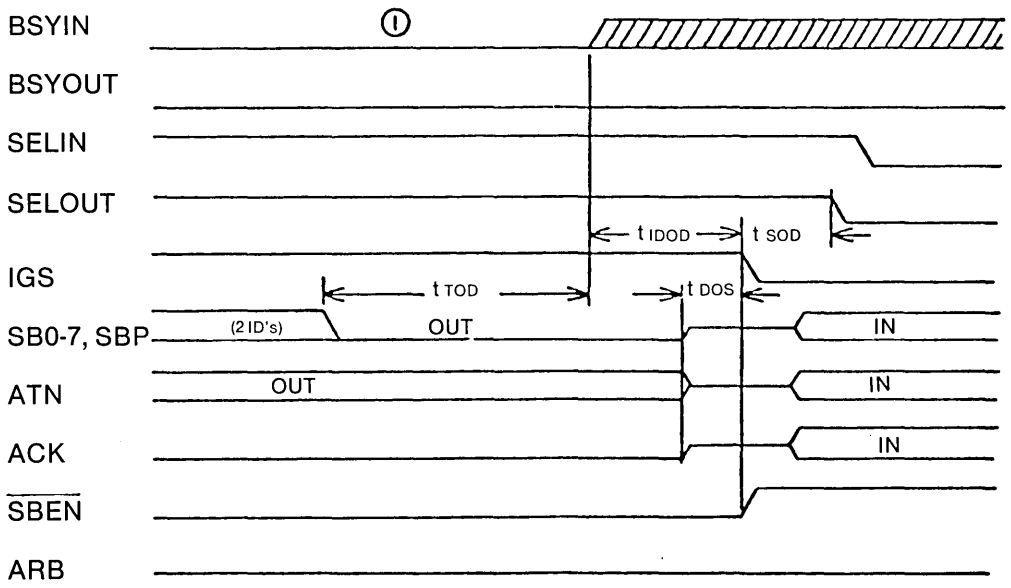
NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
$t_{S\emptyset}$	SELIN low to Phase Change	0			ns
t_{IODB}	I/O high to $\overline{SBEN}$ low	500			ns
t_{DBR}	$\overline{SBEN}$ low to REQ out	185			ns
t_{DVA}	Data Valid to REQ high	100			ns
t_{RAH}	REQ high to ACK high	0			ns
t_{ARL}	ACK high to REQ low	0			ns
t_{RAL}	REQ low to ACK low	0			ns
t_{ARH}	ACK low to REQ high	0			ns
$t_{\emptyset HA}$	Phase hold from ACK low	0			ns
t_{ADH}	Data hold from ACK low	0			ns
t_{DBE}	$\overline{SBEN}$ low to Data Bus Enabled	85			ns



8.2.9 BUS RELEASE FROM SELECTION (INITIATOR)

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{TOD}	Bus Release Timeout Delay	100			μs
t_{IDOD}	IGS & $\overline{SBEN}$ Turn-off Delay	0			ns
t_{SOD}	SELOUT Turn-off Delay	0			ns
t_{DOS}	Driver Turn-off set-up to IGS & $\overline{SBEN}$ off	0			ns

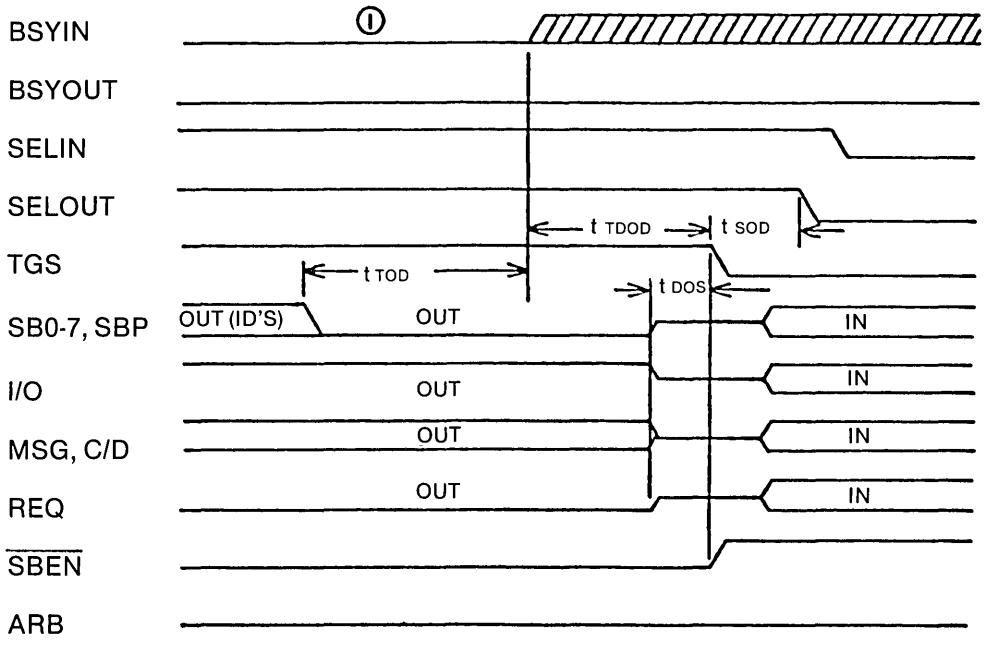
NOTE 1: If the chip detects BSYIN active by the end of the timeout delay, the bus release sequence shall be aborted since selection has been successful.



8.2.10 BUS RELEASE FROM RESELECTION (TARGET)

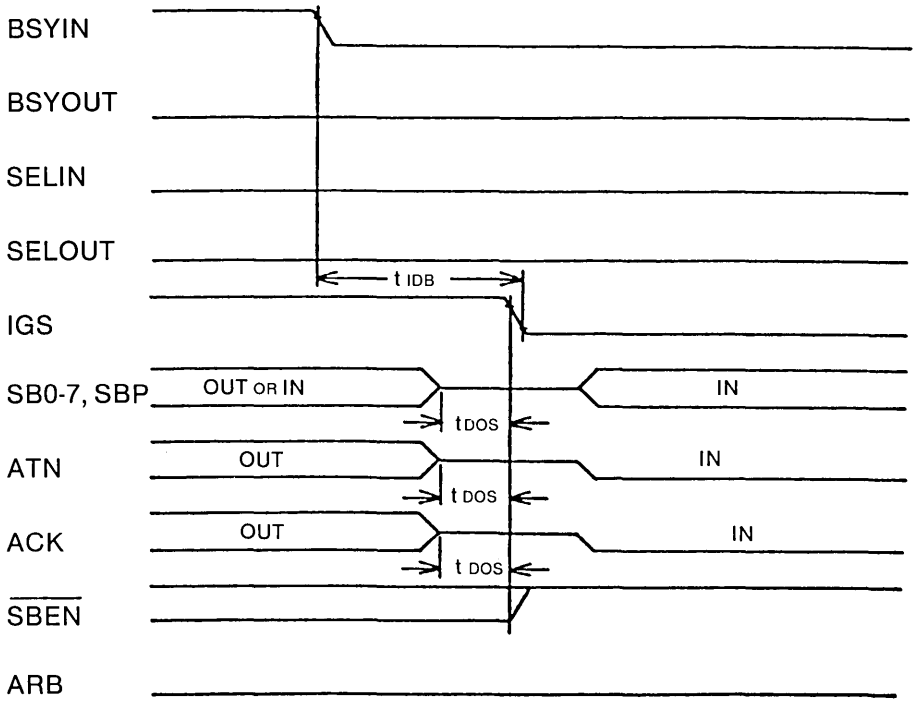
NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{TOD}	Bus Release Timeout Delay	100			*u*s
t_{DOD}	TGS & $\overline{SBEN}$ Turn-off Delay	0			ns
t_{SOD}	SELOUT Turn-off Delay	0			ns
t_{DOS}	Driver Turn-off set-up to TGS & $\overline{SBEN}$ off	0			ns

NOTE 1: If the chip detects BSYIN active by the end of the timeout delay, the bus release sequence shall be aborted since reselection has been successful.



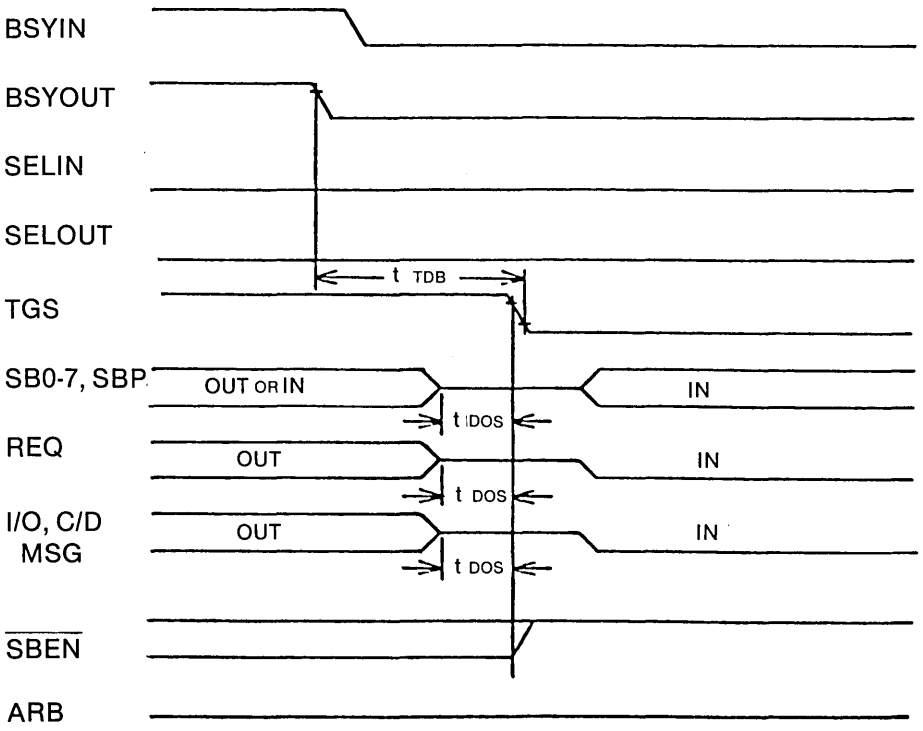
8.2.11 BUS RELEASE FROM INFORMATION PHASE (INITIATOR)

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{IDB}	IGS & $\overline{SBEN}$ Turn-off Delay from BSYIN off			225	ns
t_{DOS}	Driver Turn-off set-up to IGS off	0			ns



8.2.12 BUS RELEASE FROM INFORMATION PHASE (TARGET)

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{TDB}	TGS & $\overline{SBEN}$ Turn-off from BSYOUT off			225	ns
t_{DOS}	Driver Turn-off set-up to TGS off	0			ns



NCR Microelectronics Division
1635 Aeroplaza Drive
Logic Products Marketing
Colorado Springs, CO 80916
(303) 596-5612 or (800) 525-2252
TELEX 45-2457

MC-704 5/85