SN75C091A
SCSI Bus Controller

Data Manual

TEXAS INSTRUMENTS

1 Introduction

1.1 Description

The SN75C091A provides the capability to interface a microprocessor subsystem to the ANSI Small Computer Systems Interface (SCSI). This single-ended SCSI implementation acts as one of up to eight nodes communicating over a maximum six-meter bus as detailed in the SCSI specification (ANSI X3.131-1986). Its microprocessor and DMA interfaces allow the SN75C091A SCSI Bus Controller (SBC) to be used in a variety of host or peripheral applications.

This document is not intended to serve as a tutorial on the Small Computer Systems Interface bus; users should refer to ANSI X3.131-1986 for detailed information regarding the SCSI bus.

1.2 Features

1.2.1 SCSI Bus Interface

- Complies with ANSI X3.131-1986 SCSI standard
- Performs INITIATOR and TARGET functions
- Supports arbitration, selection, and reselection
- Performs asynchronous data transfers of up to 5 Megabytes/second (MBps)
- Performs synchronous data transfers of up to 5 Megabytes/second (MBps) with programmable offset up to 15
- Has on-chip 48-mA transceivers
- Provides optional parity generation, checking, and pass-through
- Reduces overhead associated with initiator multi-threading by automatically handling save-data-pointer messages, disconnects, and reconnects
- Performs automatic message and command-length decoding
- Has two 32-byte FIFOs for command and message preloading

1.2.2 Microprocessor Interface

- Provides chip control via directly-addressable registers
- Has optional address latch line for multiplexed address/data buses
- Allows DMA- or programmed-I/O data transfers
- Is interrupt-driven to minimize host polling
- Can execute multi-phase commands to minimize interrupts
- Has 24-bit transfer counter
- Provides byte-stacking control to accommodate 8-, 16-, and 32-bit systems
- Offers optional parity generation and checking
- Is equipped with separate ports for DMA and microprocessor interfacing

1.2.3 General

- Requires a single 5 V ± 5% power supply
- Low-power CMOS technology
- 68-pin PLCC package

2 Architecture

2.1 Block Diagram

The functional block architecture of the SN75C091A is as shown in Figure 2–1 below:

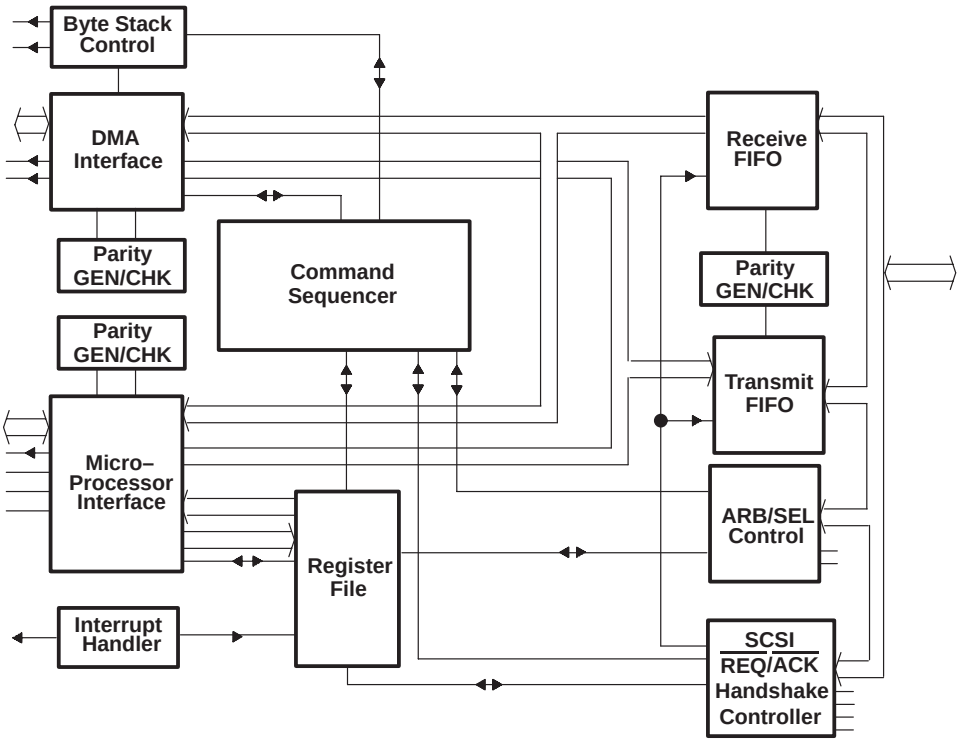


Figure 2–1. SN75C091A Functional Block Architecture

2.1.1 Command Sequencer

The heart of the SN75C091A is a large state machine called the command sequencer. Microprocessor commands to the SN75C091A are interpreted by the command sequencer, which then activates subordinate state machines (e.g. the arbitration controller, $\overline{\text{REQ}}/\overline{\text{ACK}}$ handshake controller, and DMA interface controller) to perform the functions necessary to carry out each command. The command sequencer enables the SN75C091A to execute powerful multiphase SCSI sequences with very few interrupts.

2.1.2 SCSI $\overline{\text{REQ}}/\overline{\text{ACK}}$ Handshake Controller

The $\overline{\text{REQ}}/\overline{\text{ACK}}$ handshake controller handles requests from the command sequencer to perform single SCSI phase transfers such as Message, Command, Status, and Data. $\overline{\text{REQ}}$, $\overline{\text{ACK}}$, $\overline{\text{ATN}}$, $\overline{\text{MSG}}$, $\overline{\text{C/D}}$ and $\overline{\text{I/O}}$ are controlled and/or monitored (depending on the mode, target or initiator) to transfer information between the transmit and receive FIFOs and the SCSI bus. The $\overline{\text{REQ}}/\overline{\text{ACK}}$ handshake controller supports both synchronous and asynchronous data transfers.

2.1.3 Arbitration and Selection Controller

The arbitration and selection controller handles requests from the command sequencer to establish a connection with another device on the SCSI bus. SCSI bus signals $\overline{BSY}$, $\overline{SEL}$, $\overline{I/O}$ and $\overline{ATN}$ are controlled and/or monitored to effect automatic completion of SCSI arbitration, selection, and reselection phases. This state machine also concurrently monitors the SCSI bus and alerts the command sequencer and interrupt logic if the SN75C091A has been selected or reselected by another device on the SCSI bus.

2.1.4 Register File

The register file consists of 32 registers which allow the local microprocessor to initiate, control, and monitor SCSI transfer operations performed by the SN75C091A. The 32-byte transfer and receive FIFOs can also be accessed through the register file.

2.1.5 Microprocessor Interface

The microprocessor interface provides the logic necessary for a microprocessor or other host computer system to access and store information in the register file. Both multiplexed and nonmultiplexed address/data buses are supported through this interface.

2.1.6 Receive and Transmit FIFOs

These 32-byte by 9-bit FIFOs provide a buffer between the SCSI bus and memory to improve transfer efficiency and minimize microprocessor overhead. These FIFOs are accessed in the same manner as a register in the register file: through the microprocessor interface for programmed I/O or through the DMA interface for SCSI data transfers performed via an intermediate DMA controller. The 32-byte FIFOs allow SCSI commands and messages to be preloaded or fully received, thus minimizing microprocessor intervention. The ninth bit allows parity pass-through mode for high-reliability systems.

2.1.7 Interrupt Control

Interrupt control logic monitors the various state machines to determine when microprocessor intervention is required. Interrupt status information is maintained in the register file and the interrupt is reported either by microprocessor polling or via the INTRQ signal (if external interrupts are enabled).

2.1.8 DMA Interface

The DMA interface provides the control logic necessary to interface the SN75C091A with an external DMA controller. DMA handshake signals DREQ and $\overline{DACK}$ and a separate 9-bit data port form the data path used to transfer SCSI data between external memory and the SN75C091A transmit and receive FIFOs without microprocessor intervention.

2.1.9 Byte Stack Control

The byte stack control is used in conjunction with the DMA interface to allow easy interfacing of the 8-bit SCSI bus to 16-, 24-, and 32-bit systems. This control logic facilitates loading and unloading of external bidirectional registers (byte stack registers) so that 16-, 24-, and 32-bit words can be broken down into their constituent bytes. No external logic is required to interface the SN75C091A to a 16-bit bytestack register; a decoder is necessary for interfacing to 24- and 32-bit systems.

2.1.10 Parity Generators/Checkers

Parity generation and checking is provided for all three SN75C091A ports (SCSI, DMA, and microprocessor). Versatile parity control via the register file allows the SN75C091A to adapt easily to any system. Parity generation control allows memory and SCSI parity information to be passed through the FIFOs. User-selectable parity sense (odd or even) provides error generation capabilities to assist in system checkout.

2.2 Data Path Examples

The SN75C091A provides a microprocessor port for information transfer and chip control. A separate DMA port is also provided for SCSI data transfers between memory and the SCSI bus. The DMA port may be connected directly to an 8-bit system or through byte stack registers to 16-, 24-, and 32-bit systems.

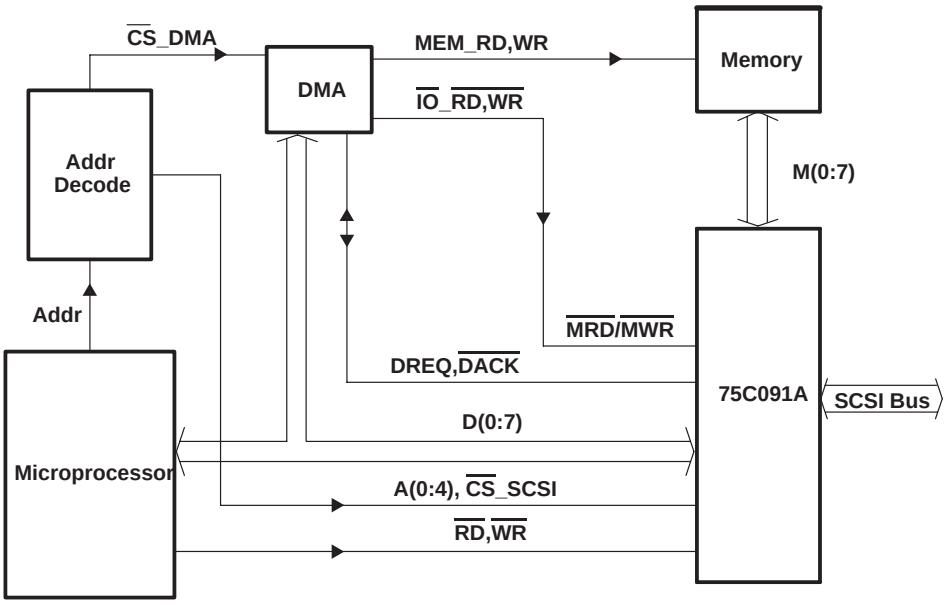
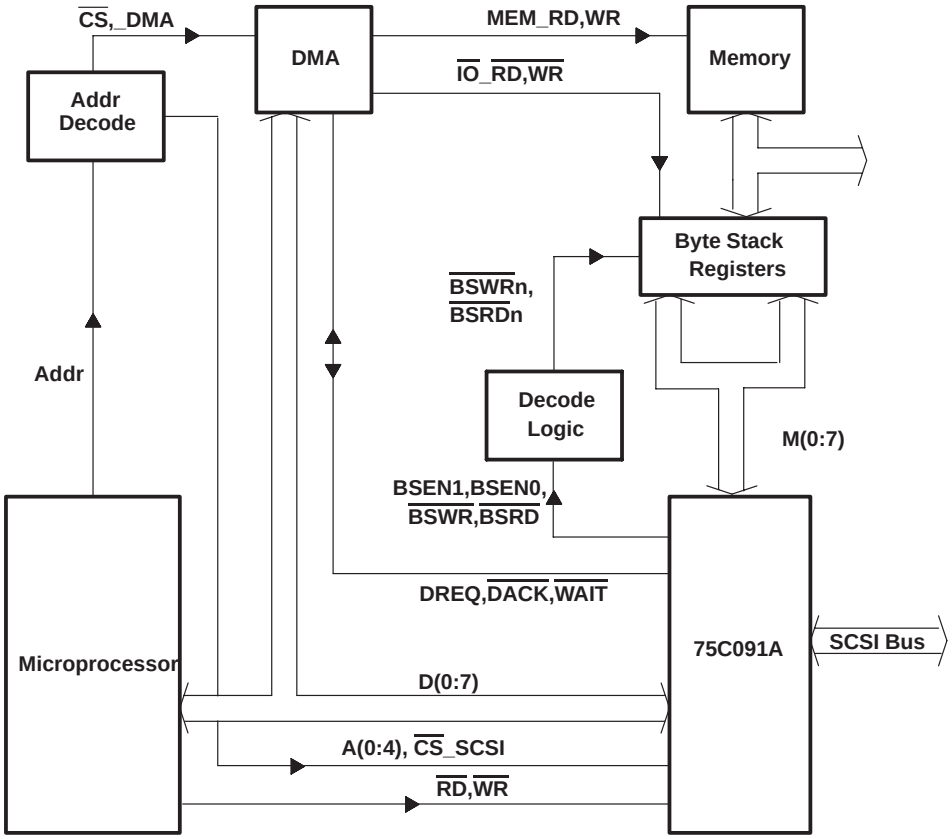


Figure 2–2. Data Paths (No Byte Stacking)

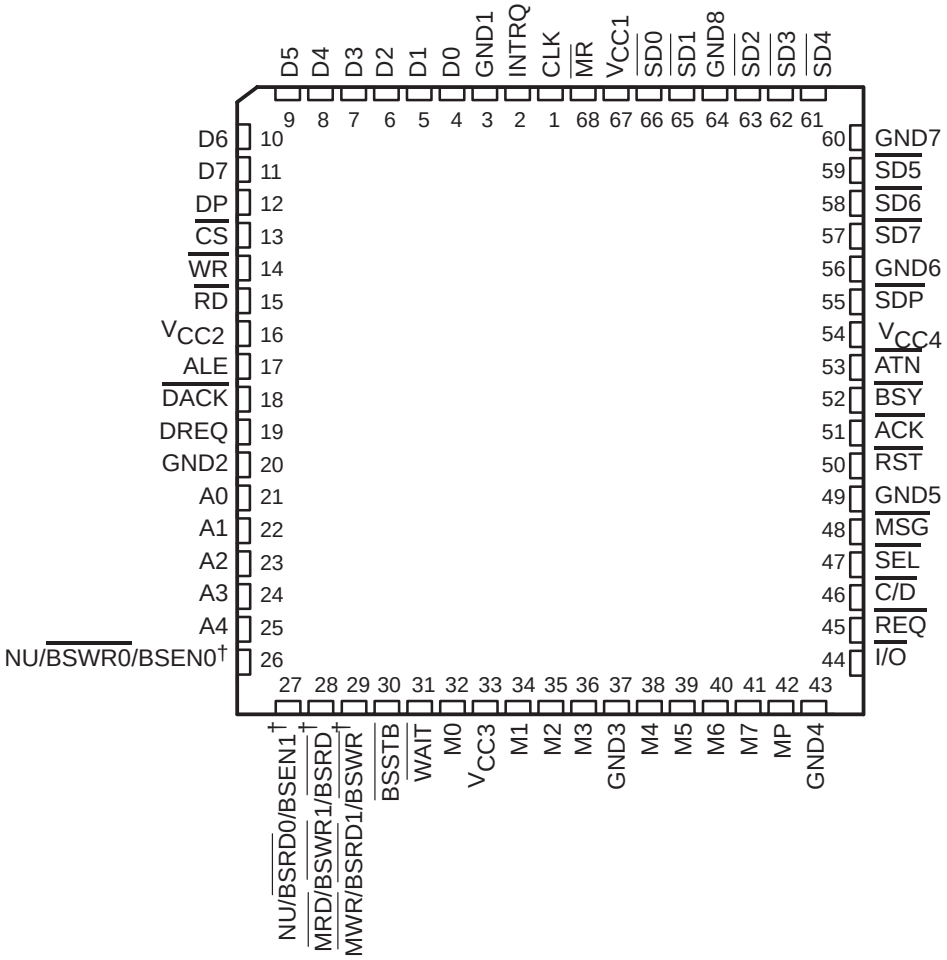


NOTE: For 16-bit byte stack operation, no decode logic is required to produce $\overline{\text{BSWRn}}$ and $\overline{\text{BSRDn}}$. These signals are decoded internally to minimize the external logic required for this application.

Figure 2–3. Data Paths (With Byte Stacking)

2.3 Pin Assignments and Functions

2.3.1 Pin Assignments

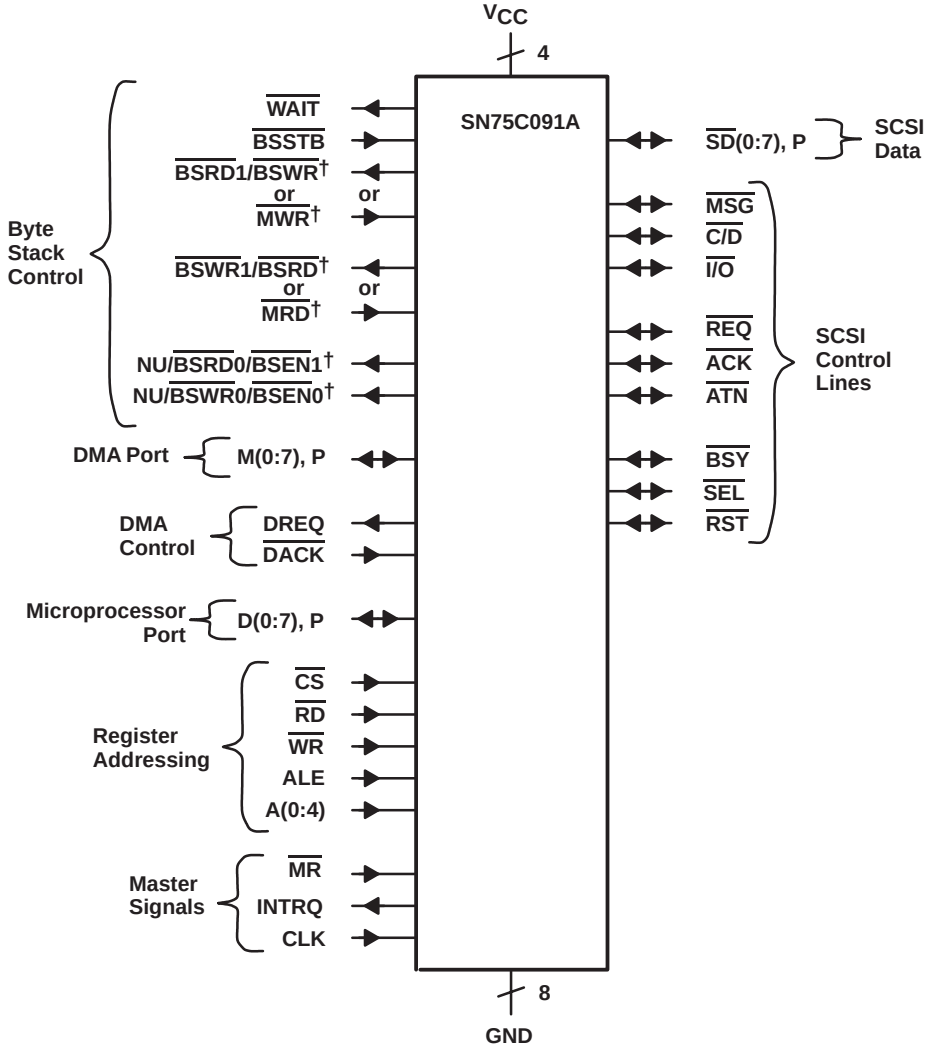


NU – make no external connection

† Use of these pins varies depending on whether non-byte-stack, 16-bit byte stack, or 24-/32-bit byte stack mode is being used. See Section 2.4.2, DMA Interface Signals, for details.

Figure 2–4. SN75C091A Pin Assignments

2.3.2 Pin Functions



NU – Make no external connection

† Use of these pins varies depending on whether non-byte-stack, 16-bit byte stack, or 24-/32-bit byte stack mode is being used. See Section 2.4.2, DMA Interface Signals, for details.

Figure 2–5. SN75C091A Pin Functions

2.4 Pin Functional Descriptions

The following tables describe the function of the SN75C091A signals. Local processor and DMA signals are presented first, followed by SCSI bus control and miscellaneous signals.

2.4.1 Microprocessor Interface Signals

SN75C091A SIGNALS MICROPROCESSOR INTERFACE				
PIN NO.	**MNEMONIC**	**SIGNAL NAME**	**I/O**	**FUNCTION**
1	CLK	Clock	I	Clock used for timing and internal control. This clock must meet chip specifications in order to produce proper timing to meet the SCSI specification. Nominal frequency is 20 MHz.
68	$\overline{MR}$	Mister Reset	I	$\overline{MR}$ places the SN75C091A into an idle state with all signals in the passive mode.
2	INTRQ	Interrupt Request	O	Indicates to the host or local processor that the interrupt registers should be read.
4, 5, 6, 7, 8, 9, 10, 11	D(0:7)	Data 0 through Data 7	I/O	Local processor data bus bits 0 through 7.
12	DP	Parity	I/O	Parity bit for the local processor data bus.
13	$\overline{CS}$	Chip Select	I	Asserted by the processor to enable access to the register file or FIFOs.
14	$\overline{WR}$	Write Enable	I	When asserted in conjunction with $\overline{CS}$, latches data into the register file on the rising edge.
15	RD	Read Emab;e	I	Used in conjunction with $\overline{CS}$ to read from the register file.
17	ALE	Address Latch Enable	I	On the falling edge of ALE, address on A(0:4) is latched into address register for multiplexed address/data buses. For non-multiplexed address/data buses, ALE should be tied high.
21, 22, 23, 24, 25	A(0:4)	Register Address 0 through Register Address 4	I	Address bit 0 (least significant bit) of the register file through address bit 4 (most significant bit) of the register file.

2.4.2 DMA Interface Signals

SN75C091A SIGNALS DMA INTERFACE				
PIN NO.	**MNEMONIC**	**SIGNAL NAME**	**I/O**	**FUNCTION**
19	DREQ	DMA Request	O	DREQ interfaces with an external DMA controller and forms the handshake for data transfers.
18	$\overline{\text{DACK}}$	DMA Acknowledge	I	$\overline{\text{DACK}}$ interfaces with an external DMA controller and is the response to DREQ. Data is read or written from/to the FIFOs while $\overline{\text{DACK}}$ is asserted.
32, 34, 35, 36, 38, 39, 40, 41	M(0:7)	DMA Port Data Bus	I/O	DMA port data bits 0 through 7. This bus is an alternate path into the transmit and receive FIFOs.
42	MP	DMA Port Parity	I/O	Parity for DMA port.
31	$\overline{\text{WAIT}}$	Processor Wait Enable	O	Signals the DMA to wait when the FIFO status is inappropriate for loading or unloading in DMA demand mode. Also used to suspend DMA activity while the byte stack is loaded or unloaded.
30	$\overline{\text{BSSTB}}$	Byte Stack Strobe		Informs the byte stack control logic that a read or write from/to the byte stack register has been completed. For 8-bit mode, $\overline{\text{BSSTB}}$ should be tied high.

The following DMA interface pins have different functions depending on whether the chip is in byte stack or non-byte-stack mode. Also, their byte stack mode functions vary with byte stack width (16, 24, or 32 bits).

SN75C091A SIGNALS DMA INTERFACE (8-BIT NON-BYTE-STACK MODE)				
PIN NO.	**MNEMONIC**	**SIGNAL NAME**	**I/O**	**FUNCTION**
29	$\overline{\text{MWR}}$	DMA Port Write Enable	I	$\overline{\text{MWR}}$ is used with $\overline{\text{DACK}}$ to write to the transmit $\overline{\text{FIFO}}$ through the DMA port. Data is latched on rising edge of $\overline{\text{MWR}}$.
28	$\overline{\text{MRD}}$	DMA Port Read Enable	I	$\overline{\text{MRD}}$ is used with $\overline{\text{DACK}}$ to read from the receive FIFO through the DMA port.
27, 26	n/a	n/a	O	Not used in 8-bit mode.

SN75C091A SIGNALS DMA INTERFACE (16-BIT MODE)				
PIN NO.	**MNEMONIC**	**SIGNAL NAME**	**I/O**	**FUNCTION**
29	$\overline{\text{BSRD1}}$	Byte Stack Read-Byte 1	O	Enables byte stack register byte 1 onto M(0:7), MP for loading into the transmit FIFO.
28	$\overline{\text{BSWR1}}$	Byte Stack Write-Byte 1		Loads receive FIFO byte into byte stack register byte 1 on rising edge.
27	$\overline{\text{BSRD0}}$	Byte Stack Read-Byte0	O	Enables byte stack register byte 0 onto M(0:7), MP for loading into the transmit FIFO.
26	$\overline{\text{BSWR0}}$	Byte Stack Write-Byte 0	O	Loads receive FIFO byte into byte stack register byte 0 on rising edge.

SN75C091A SIGNALS DMA INTERFACE (24- OR 32-BIT MODE)				
PIN NO.	**MNEMONIC**	**SIGNAL NAME**	**I/O**	**FUNCTION**
29	$\overline{\text{BSWR}}$	Byte Stack Write	O	Loads receive FIFO byte into selected byte stack register byte on rising edge.
28	$\overline{\text{BSRD}}$	Byte Stack Read	O	Enables selected byte stack register byte onto M(0:7), MP for loading into the transmit FIFO.
27, 26	BSEN1-BSEN0	Byte Stack Byte Enable	O	Encoded value of selected byte stack register byte to be enabled for read or write.

2.4.3 SCSI Bus Interface Signals

SN75C091A SIGNALS SCSI BUS INTERFACE				
PIN NO.	**MNEMONIC**	**SIGNAL NAME**	**I/O**	**FUNCTION**
48	$\overline{MSG}$	Message	I/O	MSG, C/D, and I/O are the SCSI bus phase signals used to determine the type and direction of a transfer. They are driven by a target and received by an initiator.
46	$\overline{C/D}$	Command/Data	I/O	
44	$\overline{I/O}$	Input/Output	I/O	
52	$\overline{BSY}$	Busy	I/O	Drives and monitors the $\overline{BSY}$ line of the SCSI bus.
47	$\overline{SEL}$	Se;ect	I/O	Drives and monitors the $\overline{SEL}$ line of the SCSI bus.
50	$\overline{RST}$	SCSI Reset	I/O	Drives and monitors the $\overline{RST}$ line of the SCSI bus.
45	$\overline{REQ}$	Request	I/O	$\overline{REQ}$ starts the $\overline{REQ}$/$\overline{ACK}$ handshake. It is driven by the target and received by the initiator.
51	$\overline{ACK}$	Acknowledge	I/O	$\overline{ACK}$ answers the $\overline{REQ}$/$\overline{ACK}$ information handshake. It is driven by the initiator and received by the target.
53	$\overline{ATN}$	Attention	I/O	$\overline{ATN}$ indicates to the target that the initiator has a message to send. $\overline{ATN}$ is driven by the initiator and received by the target.
55	$\overline{SDP}$	SCSI Data Parity	I/O	SCSI bus data parity line.
66, 65, 63, 62, 61, 59, 58, 57	$\overline{SD}$(0:7)	SCSI Data 0 through SCSI Data 7	I/O	SCSI bus data bit 0 through SCSI bus data bit 7.

2.4.4 Miscellaneous Signals

SN75C091A SIGNALS MISCELLANEOUS				
PIN #	**MNEMONIC**	**SIGNAL NAME**	**I/O**	**FUNCTION**
3, 20, 37, 43, 49, 56, 60, 64	GND	Ground		Ground reference
67, 16, 33, 54	V_{CC}	V_{CC}		5 V ±5% power supply

3 Internal Registers

3.1 General

The local microprocessor directs the operation of the SCSI bus controller (SBC) through a set of registers internal to the SBC. For nonmultiplexed address/data bus systems, these registers are read or written by asserting $\overline{CS}$ with the proper address on A(0:4) and then asserting $\overline{RD}$ or $\overline{WR}$. For multiplexed systems, the address and chip select are latched internally using ALE; the read or write strobes can then be applied. The following table lists the register addresses; subsequent paragraphs describe the functions of the various registers. (Note: many of the register function descriptions state that the register is set to all zeros by a master reset. “Master reset” in this context means either that the microprocessor has sent a Chip Reset command to the SBC or that the $\overline{MR}$ (master reset) line has been asserted.)

REGISTER ADDRESSES						
A4	**A3**	**A2**	**A1**	**A0**	**READ/WRITE**	**REGISTER**
0	0	0	0	0	R	Receive FIFO
0	0	0	0	0	W	Transmit FIFO
0	0	0	0	1	R/W	Command
0	0	0	1	0	R	Transfer status
0	0	0	1	1	R	Bus phase status
0	0	1	0	0	R	Function interrupt status
0	0	1	0	1	R	Error interrupt status
0	0	1	1	0	R/W	Interrupt enable
0	0	1	1	1		(Reserved)
0	1	0	0	0	R/W	Control
0	1	0	0	1	R/W	Byte stack control
0	1	0	1	0	R/W	Parity control
0	1	0	1	1	R/W	Synchronous transfer
0	1	1	0	0	R/W	Selection or Reselection timeout
0	1	1	0	1	R/W	Self-ID
0	1	1	1	0	R/W	Destination ID
0	1	1	1	1	R	Source ID
1	0	0	0	0	R/W	Target LUN
1	0	0	0	1	R/W	Command state
1	0	0	1	0	R/W	Transfer counter (least significant byte)
1	0	0	1	1	R/W	Transfer counter (middle byte)
1	0	1	0	0	R/W	Transfer counter (most significant byte)
1	0	1	0	1	R	Backup counter (least significant byte)
1	0	1	1	0	R	Backup counter (middle byte)
1	0	1	1	1	R	Backup counter (most significant byte)
1	1	0	0	0	R/W	Offset counter

REGISTER ADDRESSES						
A4	**A3**	**A2**	**A1**	**A0**	**READ/WRITE**	**REGISTER**
1	1	0	0	1		(Reserved)
1	1	0	1	0	R/W	Test control
1	1	0	1	1	R	Test points register 0
1	1	1	0	0		(Reserved)
1	1	1	0	1		(Reserved)
1	1	1	1	0		(Reserved)
1	1	1	1	1		(Reserved)

3.2 Transmit and Receive FIFOs

The SN75C091A uses two 32-byte by 9-bit FIFOs to buffer SCSI bus information transfers. The Receive and Transmit FIFOs are accessed through the microprocessor port at register file hexadecimal address 00000. Writing loads a byte into the transmit FIFO through the microprocessor port; reading enables the information onto the microprocessor port and unloads the byte from the receive FIFO. By polling the transfer status register FIFO status bits, the microprocessor can determine availability of space in the transmit FIFO or data in the receive FIFO. The microprocessor should never read the receive FIFO when it is empty or write the transmit FIFO when it is full, as loss of information integrity will result. Also, the FIFOs should not be acessed during execution of a command which uses the DMA interface. Note that 32-byte FIFOs are large enough to accommodate most SCSI messages or commands, so no polling is required for these types of transfers. Transmit and Receive FIFO pointers are reset by a master reset or by the appropriate FIFO clear command.

Parity	7	6	5	4	3	2	1	0

3.3 Command Register

The command register is an eight-bit read/write register that stores chip commands written by the microprocessor. Each command is executed immediately upon being sent to the chip. Generally, the microprocessor should not issue a new command to the SN75C091A while the previously issued command is still active (for exceptions, see Section 4, Commands). The command register is set to all zeros by a master reset.

7	6	5	4	3	2	1	0
DMA	**M/$\overline{\text{A}}$**	**DDIR**	**CC4**	**CC3**	**CC2**	**CC1**	**CC0**

DMA: Direct Memory Access. This bit controls the mode of data transfer from the SCSI bus to the microprocessor or DMA bus. When set low, the interface uses programmed I/O. When set high, DMA transfers are enabled. The microprocessor should not access the FIFOs until the command is complete if DMA transfers are enabled.

M/$\overline{\text{A}}$: Manual/$\overline{\text{Automatic}}$. This bit allows the microprocessor to manually control the number of bytes transferred during a command or message phase. When this bit is set to one, the count written to the offset counter by the microprocessor is used to determine the transfer length of the command or message phase. When this bit is set to zero, the group code of a command or the second byte of an extended message is decoded to determine a count value to be automatically loaded into the offset counter. Automatic mode eliminates the need for software to decode the command or message prior to completing the transfer. Manual mode may be used to complete a phase which was terminated prematurely.

DDIR: Data DIRection. This bit establishes the direction of data transfer during Select with- or Select without- $\overline{\text{ATN}}$ and Transfer commands. When DDIR is set low, a data-out phase is expected; when set high, a data-in phase is expected. If the data phase set up by the target does not match that expected by the initiator, a bus service interrupt is generated and the command stops.

CC4–CC0: Command codes (see Section 4, Commands).

3.4 Transfer Status Register

The transfer status register is an eight-bit read-only register that stores bits which reflect the operational state of the chip.

7	6	5	4	3	2	1	0
INT	**RFE**	**RFHF**	**TFF**	**TFHF**	**TC0**	**OC0**	**CDACT**

INT: INTerrupt pending. When set to 1, this bit indicates that an interrupt condition is pending (i.e., one or more bits in either the functional interrupt status register or error interrupt status register are set to 1). This bit is provided for systems which detect pending interrupt conditions through the use of a polling scheme rather than by monitoring the external INTRQ line (see description of MIE bit in the Interrupt Enable Register description). If INT is set to 1, the microprocessor must read the functional interrupt status register before issuing a command (the error interrupt status register may also need to be read). This bit is set to 0 when all interrupts in both interrupt registers have been cleared .

RFE: Receive Fifo Empty. This bit indicates the state of the receive FIFO during incoming information transfers. When this bit is set to 1, either no bytes have been received from the bus or all bytes that had been received have already been read. RFE is also set to 1 by a master reset. When RFE is set to 0, some bytes remain to be read from the FIFO.

RFHF: Receive Fifo Half Full. This bit is set to 1 if the receive FIFO contains sixteen or more bytes, and is set to 0 otherwise. It is also set to 0 by a master reset.

TFF: Transmit Fifo Full. This bit indicates the state of the transmit FIFO during outgoing information transfers. When this bit is set to 1, the FIFO is full and no more bytes may be written to it; when set to 0, the FIFO has room to accept more bytes.TFF is also set to 0 by a master reset.

TFHF: Transmit Fifo Half Full. This bit is set to 1 if the transmit FIFO contains sixteen or more bytes, and is set to 0 otherwise. It is also set to 0 by a master reset.

TC0: Transfer Counter Zero. This bit is set to1 whenever the transfer counter is zero.

OC0: Offset Counter Zero. This bit is set to 1 whenever the offset counter is zero.

CDACT: CommanD ACTive. This bit, when set to 1, indicates that an interrupting command is being executed. Only the command, transfer status, or bus phase status registers or the transmit/receive FIFOs should be accessed while this bit is set to 1.

3.5 Bus Phase Status Register

The bus phase status register is an eight-bit read-only register that stores bits which reflect the operational state of the chip and the present SCSI bus phase.

7	6	5	4	3	2	1	0
INIT	**TARG**	**0**	**ATN**	**MSG**	**C/D**	**I/O**	**SRST**

INIT: INITiator. This bit is set to 1 whenever the chip is logically connected as an initiator. It is set to 0 upon target disconnection, by a master reset, or by a SCSI reset.

TARG: TARGet. This bit is set to 1 whenever the chip is logically connected as a target. It is set to 0 at disconnection (disconnect command or multiphase command internal disconnect), by a master reset, or by a SCSI reset.

ATN: If the ATNDS (attention disable) bit in the control register is set to 0, this active-high bit represents the state of the SCSI bus line $\overline{ATN}$.

MSG, C/D, I/O: These three active-high bits represent the state of the SCSI bus phase lines $\overline{MSG}$, $\overline{C/D}$, and $\overline{I/O}$, respectively. They are used by a device connected in the initiator mode to determine which bus phase the target is requesting when a bus service interrupt is generated.

SRST: Scsi $\overline{RST}$. This active-high bit represents the state of the SCSI bus line $\overline{RST}$.

3.6 Functional Interrupt Status Register

The functional interrupt status register is an eight-bit read-only register that reflects SN75C091A functional interrupts. If no interrupting commands are active, this register reports an interrupt condition immediately; otherwise, the bits in this register are updated as the SN75C091A completes or aborts command execution. When this register is read, its bits are latched in order to provide stable data to the microprocessor. With the exception of the ABEND bit, any read-latched interrupt bit is cleared (set to 0) after the read is complete. A cleared interrupt bit is not be set to 1 again until the corresponding interrupt condition recurs. A persistent condition such as SCSI bus $\overline{ATN}$ only causes the $\overline{ATN}$ interrupt once. If new interrupts occur during a read of the functional interrupt status register, they are queued; when the register read is complete, these queued interrupts then cause the appropriate functional interrupt status register bits to be set to 1.

The functional interrupt status register is set to all zeros by a master reset. All bits except FC are also set to 0 by a SCSI reset.

7	6	5	4	3	2	1	0
SEL	**BUS**	**ATN**	**FC**	**DIS**	**0**	**RSL**	**ABEND**

SEL: SELected. When set to 1, this bit indicates that the chip has been selected as a target by another device on the bus. The chip is selected only if it detects its own ID with good parity on the data bus during the selection phase and if there is only one other ID on the bus. After setting this interrupt, the chip is connected as a target and waits for a command to be loaded.

BUS: BUs Service. When set to 1, this bit indicates to an initiator-connected device that an unanswered SCSI bus request is pending and that the microprocessor needs to issue an appropriate command based on the SCSI phase observed in the bus phase register. There are three occasions when this situation may occur:

When a $\overline{REQ}$ follows a reselection.

When a pending $\overline{REQ}$ follows an aborted command (for example, a command may be aborted due to a parity error halt condition or an unexpected phase change).

When a $\overline{REQ}$ follows a completed command.

ATN: ATtentioN. This interrupt indicates to a device connected as a target that the $\overline{ATN}$ bus line has been asserted by the initiator. This interrupt occurs only if the control register ATNDS (attention disable) bit is set to 0 (ATNDS inactive).

FC: Function Complete. When set to 1, this bit indicates that the previous interrupting command has fully completed (i.e., has not been halted).

DIS: DISconnected. This bit is set to 1 when a device connected in the initiator role detects that the target has legally released the $\overline{BSY}$ line (i.e., has disconnected).

RSL: ReSeLected. This bit is set to 1 when the SBC has been reselected by another device on the bus. The SBC is reselected only if it detects its own ID on the data bus with good parity and if there is only one other ID asserted on the data bus. After the interrupt, the SBC is logically connected as an initiator and waits for a $\overline{REQ}$ from the target.

ABEND: ABnormal ENDing. This bit, when set to 1, indicates that further interrupt information is available in the error interrupt status register (EISR). If set to 0, then all interrupt information can be obtained from the functional interrupt status register. This bit is the logical OR of all the bits in the EISR. It is set to 0 when the EISR is read.

3.7 Error Interrupt Status Register

The error interrupt status register is an eight-bit read-only register that reflects SN75C091A error condition interrupts. If any bit in this register is set to 1, the ABEND bit in the functional interrupt status register is also set to 1. All bits in this register are set to 0 by a master reset. This register operates in the same manner as the functional interrupt status register.

7	6	5	4	3	2	1	0
PE	**UMS**	**SRST**	**T-O**	**NVC**	**CNTL**	**NEWLN**	**HALT**

PE: Parity Error. This bit is set to 1 when a parity error is detected on a byte received from any port on the chip. Note that this assumes that parity checking is enabled.

UMS: Unexpected Message Sequence. This bit is set to 1 when a target executing a Wait for Select with $\overline{\text{ATN}}$ Command receives a message other than an ID message following the selection phase.

SRST: Scsi ReSeT. This bit is set to 1 when the chip detects assertion of the SCSI $\overline{\text{RST}}$ line. Release of the SCSI reset line may be detected by polling the SRST bit in the bus phase status register. SCSI reset has the same effect as a Chip Reset command except that only selected bits in the register file are reset and the FIFO pointers are not reset.

T-O: Time-Out. This bit is set to 1 when a selection or reselection timeout occurs. Following the timeout interrupt, the SCSI select line is held active until the microprocessor issues either a SCSI reset or disconnect command.

NVC: iNValid Command. This bit is set to 1 when an invalid command is written to the command register. An invalid command is a reserved command code or a command issued at an inappropriate time (e.g., a Send command issued by a device in initiator mode). See Appendix C for a listing of invalid command conditions.

CNTL: CoNTroL error. When set to 1, this bit indicates to an initiator that the target has unexpectedly and illegally disconnected while a chip command is active.

NEWLN: NEW LuN. When set to 1, this bit indicates that a new LUN (Logical Unit Number) reselected the chip during a Select and Transfer command. The new LUN is available in the LUN register. The microprocessor should also check the SDP bit in the command phase register to see if the data pointer for the previous LUN should be updated.

HALT: HALTed. This bit is set to 1 whenever a chip operation is halted by a Pause command or an error condition such as a parity error. See the individual command descriptions for further detail.

3.8 Interrupt Enable Register

Interrupt conditions are reported to the SBC through bits in the functional interrupt status register (FISR) and error interrupt status register (EISR). However, corresponding interrupts are not issued from the SBC to the microprocessor unless certain bits in the SBC interrupt enable register are set to 1. The interrupt enable register FCIE and AIE bits enable interrupt reporting to the microprocessor via the INT bit in the transfer status register. If the interrupt enable register MIE bit is set to 1, then interrupts can also be reported to the microprocessor via the SBC INTRQ line.

7	6	5	4	3	2	1	0
0	**0**	**0**	**0**	**0**	**FCIE**	**AIE**	**MIE**

FCIE: Function Complete Interrupt Enable. When set to 1, this bit enables the chip to report a function-complete interrupt to the microprocessor via the transfer status register INT bit upon command completion. When FCIE is set to 0, no interrupt is reported. For initiator mode transfers, this allows the interrupt to be held off until a subsequent bus service request is generated; for target commands, the interrupt is held off until a subsequent selection occurs.

AIE: $\overline{\text{ATN}}$ Interrupt Enable. This bit, when set to 1, enables an $\overline{\text{ATN}}$ interrupt to be generated for a target device when the SCSI $\overline{\text{ATN}}$ line is detected active. If AIE is set to 0, no interrupt is generated. In either case, the ATN bits in both the bus phase status register and the functional interrupt status register are set to 1.

MIE: Master Interrupt Enable. When set to 1, this bit enables the INTRQ line to reflect a pending interrupt. If MIE is set to 0, the INT bit in the transfer status register must be polled to determine when an interrupt is active.

3.9 Control Register

The control register is an eight-bit read/write register used to store bits which control various functions and operating modes within the SCSI interface of the SBC chip. This register is set to all zeros by a master reset.

7	6	5	4	3	2	1	0
SE	**RE**	**HA**	**HPE**	**AAPE**	**HD**	**HAAM**	**ATNDS**

SE: Selection Enable. This bit, when set to 1, allows the SBC device to be selected as a target. If this bit is set to 0, then the SBC ignores any selection attempt.

RE: Reselection Enable. This bit, when set to 1, allows the SBC to be reselected as an initiator. If this bit is set to 0, then the chip ignores all reselection attempts. Also, during a Select with $\overline{\text{ATN}}$ and Transfer command, RE is copied into bit 6 of the automatically assembled ID message sent during the message-out phase.

HA: Halt on $\overline{\text{ATN}}$. Setting this bit to 1 causes a device in target mode to halt a chip command during a data phase (whether invoked by low-level or multiphase command) when the $\overline{\text{ATN}}$ line is asserted. The HALT bit in the error interrupt status register and the ATN bit in the functional interrupt status register are both set to 1, and no further REQs are generated. For asynchronous data, interrupts are updated after $\overline{\text{ACK}}$. For synchronous data, interrupts are updated when the $\overline{\text{REQ}}$-$\overline{\text{ACK}}$ offset reaches zero. If HA is set to 0, the attention interrupt is generated when the chip command completes or is aborted for some other reason (i.e., halt on parity error).

HPE: Halt on Parity Error. If this bit is set to 1, detection of a parity error on an incoming data byte by any of the SCSI, microprocessor, or DMA interfaces during a data phase results in a halt of the current chip command and the setting to 1 of error interrupt status register bits HALT and PE. If HPE is set to 0, the parity error is reported when the chip command completes or is aborted for some other reason (i.e., a pause command is issued).

AAPE: Assert $\overline{\text{ATN}}$ on Parity Error. Setting this bit to 1 causes a chip in initiator mode to assert the $\overline{\text{ATN}}$ bus line when a parity error is detected on an incoming byte by any of the SCSI, microprocessor, or DMA interfaces. Note that preloading the FIFO via the microprocessor interface (prior to issuing a chip command) does not affect the $\overline{\text{ATN}}$ line if a parity error occurs because a chip command is not active. In this case, the parity error is reported to the microprocessor through an interrupt, allowing the microprocessor to clear the FIFO and start over. A parity error has no effect on the $\overline{\text{ATN}}$ line if the AAPE bit is set to 0.

HD: Halt on Disconnect. Setting this bit to 1 causes the chip to halt the Select with $\overline{\text{ATN}}$ and Transfer command and generate a disconnect interrupt if the target legally disconnects prior to a command-complete message. If HD is set to 0, the SBC waits indefinitely to be reselected.

HAAM: Hold $\overline{\text{ATN}}$ After Message. Setting this bit to 1 provides the special control needed for sending multiple messages during a message-out phase when the chip is acting as an initiator. When HAAM is set to 1, the $\overline{\text{ATN}}$ line is held low when the command terminates. In contrast, when HAAM is set to 0, the $\overline{\text{ATN}}$ line is released before $\overline{\text{ACK}}$ is asserted for the last byte of the message.

ATNDS: ATN DiSable. Setting this bit to 1 masks the SCSI $\overline{\text{ATN}}$ input so that $\overline{\text{ATN}}$ does not cause an interrupt, is not available in the bus phase or functional interrupt status registers, and is not recognized by any command. This feature is provided for target applications which do not support messages. For example, ATNDS can be used in conjunction with the Wait for Select without $\overline{\text{ATN}}$ command so that the command continues to execute even if the initiator asserts $\overline{\text{ATN}}$.

3.10 Byte Stack Control Register

The byte stack control register is an eight-bit read/write register used to store bits which control byte stack features of the chip. This register is set to all zeros by a master reset.

7	6	5	4	3	2	1	0
DMD	**0**	**0**	**0**	**WL1**	**WL0**	**BOF1**	**BOF0**

DMD: DeManD transfer DMA. This bit controls the DMA interface demand transfer mode. If set to 1, demand mode is enabled and DREQ (once asserted) is held asserted as long as there are enough bytes in the receive FIFO to read from or space in the transmit FIFO to write. In byte stack mode, the $\overline{\text{WAIT}}$ line is used to give the chip enough time to load or unload the required number of bytes from the byte stack register. The chip automatically switches in and out of demand mode as dictated by the FIFO status. If DMD is set to 0, then single transfer mode, in which a DREQ/$\overline{\text{DACK}}$ handshake is required for each transfer, is used. This feature is provided for systems in which the microprocessor needs to access memory for instructions on an interleaved basis.

WL0,1: Word Length 0,1. These bits determine the length of multi-byte words used in the byte stack interface to the DMA controller, as follows:

WL1	WL0	WORD LENGTH (BYTES)
0	0	1 (default – no byte stacking)
0	1	2
1	0	3
1	1	4

BOF0,1: Byte Offset 0,1. These bits determine the length of the byte offset used for the first word transferred to or from the DMA controller when using the byte stack control logic, as follows:

BOF1	BOF0	BYTE OFFSET (BYTES)
0	0	0 (default)
0	1	1
1	0	2
1	1	3

Further examples of byte offsets are shown in Section 6.3.3.

3.11 Parity Control Register

The parity control register is an eight-bit read/write register used to select the desired parity check/generate options for the SCSI, DMA, and microprocessor interfaces. This register is set to all zeros by a master reset.

7	6	5	4	3	2	1	0
PMPE	**MPCE**	**MPGE**	**PPCE**	**PPGE**	**SPE**	**SPCE**	**SPGE**

PMPE: Processor/Memory Parity Even. Setting this bit to 1 causes the microprocessor- and DMA-port parity check/generation to be even parity. If PMPE is set to 0, then parity is odd for both ports.

MPCE: Memory Parity Check Enable. Setting this bit to 1 forces the chip to check incoming bytes from the DMA memory port for correct parity as defined by the PMPE bit. If a parity error is detected, the proper error bit is set to 1 in the error interrupt status register. If MPCE is set to 0, no parity checking is done and no parity error flagged.

MPGE: Memory Parity Generation Enable. This bit, when set to 1, forces the chip to generate parity (type determined by the PMPE bit) on outgoing bytes to the DMA memory port. If MPGE is set to 0, then the parity sense for the DMA memory port is determined by the receive FIFO parity bit.

PPCE: Processor Parity Check Enable. When PPCE is set to 1, the chip checks incoming bytes from the local processor for correct parity as determined by PMPE bit. If a parity error is detected, the proper error bit is set to 1 in the error interrupt status register. If PPCE is set to 0, no parity checking is performed and no parity error flagged.

PPGE: Processor Parity Generation Enable. This bit, when set to 1, causes the chip to generate parity (type determined by the PMPE bit) for FIFO data accessed through the processor interface. If PPGE is set to 0, then FIFO data parity is obtained from the FIFO. (Note: parity is always generated for other registers in the register file.)

SPE: SCSI Parity Even. When set to 1, this bit causes the SCSI interface parity check/generation to be even parity. Odd parity is used if SPE is set to 0..

SPCE: SCSI Parity Check Enable. Setting this bit to 1 forces the chip to check incoming bytes from the SCSI bus for correct parity as determined by the SPE parity bit. If a parity error is detected, the proper error bit is set to 1 in the error interrupt status register. If SPCE is set to 0, then no parity checking is performed and no parity error flagged.

SPGE: SCSI Bus Parity Generation Enable. When SPGE is set to 1, the chip generates parity (type determined by the SPE bit) for data output to the SCSI bus from the FIFO. If SPGE is set to 0, parity information in the FIFO is used. Note that for automatically generated SCSI information such as the command-complete message or the selection ID, parity is always generated according to the SPE bit.

3.12 Synchronous Transfer Register

The synchronous transfer register is an eight-bit read/write register used to define the offset length and transfer period for synchronous data transfers over the SCSI bus. This register is set to all zeros by a master reset. The Offset Length bits (OL3–OL0) define the REQ-ACK offset; the maximum allowed offset is 15. This offset corresponds to the number of REQ pulses allowed to be outstanding before a corresponding ACK pulse is received by the target during synchronous data transfers. An offset length of zero implies that asynchronous mode transfers are to be used.

The Transfer Period bits (TP3–TP0) define the transfer period length in terms of internal clock cycles of the chip. The transfer period is the minimum time between leading edges of successive REQ pulses (target) or of successive ACK pulses (initiator). REQ is always active for two clock cycles, and bits TP3-TP0 contain the number of internal clock cycles for which the REQ line is held inactive prior to the next REQ pulse. Thus, the transfer period corresponds to the value in (TP3-TP0) + 2. For example, a value of 3 in bits TP3-TP0 indicates a transfer period of 5 clock cycles. The minimum allowed transfer period is four clock cycles, so

TP3-TP0 values of 0, 1, and 2 all correspond to a four-clock-cycle transfer period (REQ is always released for at least two clock cycles).

In multiple-thread Initiator SCSI applications, following reselection by a different target, the synchronous transfer register must be updated prior to negation of ACK for the ID message. This action guarantees that the correct mode is set up before data phase requests begin.

7	6	5	4	3	2	1	0
TP3	**TP2**	**TP1**	**TP0**	**OL3**	**OL2**	**OL1**	**OL0**

3.13 Selection/Reselection Time-Out Register

The selection/reselection time-out register programs the selection/reselection time-out period for a command that is selecting/reselecting another device. This time-out period begins when the chip releases BSY during the selection/reselection phase and ends when the target (initiator) responds by asserting BSY. If a time-out occurs, the chip waits for a length of time known as the selection abort time (after removing the selection/reselection ID from the bus) prior to halting the command and generating a time-out interrupt. The time-out values are multiples of 3.27 ms and the maximum programmable time-out period is 0.83 seconds. A value of zero in this register disables the time-out mechanism, allowing indefinite time-outs. The microprocessor must issue a pause command in order for a time-out to occur. Once the pause command is issued, the chip continues as if the time-out has expired. Note that this sequence assumes that the selection phase is in progress. If arbitration has not been won, a pause simply halts the command and generates a halted interrupt. This register is set to all zeros by a master reset.

7	6	5	4	3	2	1	0
TO7	**TO6**	**TO5**	**TO4**	**TO3**	**TO2**	**TO1**	**TO0**

3.14 Self ID Register

The self ID register is a read/write register that contains the encoded SCSI address of the device which is using the chip. This address is decoded and the corresponding data bus line is asserted during an arbitration phase and again during the selection or reselection phase after arbitration is won. The Self ID register is set to all zeros by a master reset.

7	6	5	4	3	2	1	0
0	**0**	**0**	**0**	**0**	**ID2**	**ID1**	**ID0**

3.15 Destination ID Register

The destination ID register contains the encoded address of the SCSI device which is to be selected or reselected. It must be loaded prior to any command which attempts a selection or reselection phase, such as the Select with ATN, Select without ATN, Reselect, Select and Transfer, or Reselect and Send/Receive Data. This address is decoded and then loaded onto the data bus during a selection or reselection phase. This read/write register is set to all zeros by a master reset.

7	6	5	4	3	2	1	0
0	**0**	**0**	**0**	**0**	**ID2**	**ID1**	**ID0**

3.16 Source ID Register

Bits 2-0 of the source ID register contain the encoded address of the initiator/target which last selected/reselected the SBC chip. The Source ID valid (SIV) bit is set to 1 if the selecting/reselecting device asserted its own ID on the bus during the selection/reselection phase. If SIV is set to 0, then the source ID is not valid. This distinction is necessary since an all-zero ID is valid and cannot otherwise be distinguished from a cleared register. If SIV is set to 0 in the source ID register after a selection, then the initiator is operating in the single-initiator mode and does not support reselection. A target has no option and must

place both its own ID and the ID of the initiator it wants to reselect on the bus when attempting a reselection phase. Thus, SIV is always set to 1 in the initiator source ID register after a successful reselection. This register is set to all zeros following a master reset.

7	6	5	4	3	2	1	0
0	**0**	**0**	**0**	**S/V**	**ID2**	**ID1**	**ID0**

3.17 Target Logical Unit Number (LUN) Register

The target LUN register is used to store received ID messages or ID messages to be sent during Select with $\overline{\text{ATN}}$ and Transfer, Reselect and Send or Reselect and Receive, or Wait for Select with $\overline{\text{ATN}}$ multiphase commands. This register must be loaded prior to commands which send an ID message. Although it is a full 8-bit register, bits 6 and 7 are not used to formulate outgoing ID messages (see bit descriptions below). Received ID messages stored in the target LUN register contain the entire 8-bit message from the SCSI bus. This register is set to all zeros by a master reset.

7	6	5	4	3	2	1	0
1	**DSCPRV**	**LUNTAR**	**Reserved**	**Reserved**	**TL2**	**TL1**	**TL0**

Bit 7: This bit is tied high internally for initiator or target multiphase commands sending an ID message (by definition of ID message, this bit value must be “1”).

DSCPRV: DiSConnect PRiVilege. This bit is used by the initiator to grant the target the privilege of disconnecting. It is internally connected to the RE (Reselection Enable) bit in the control register for initiator ID message out. This creates an interlock to prevent reselection from being granted to the target when it is disabled in the initiator’s control register. For a target sending the ID message, this bit is internally set to 0.

LUNTAR: For SCSI 1, this bit is reserved and should be set to 0. SCSI 2 protocols use this bit to associate the identify message with a target routine (bit value 1) or a logical unit (bit value 0).

Bits 4-3: These bits are reserved by the SCSI specification and should be set to 0.

TL2-TL0: Target Lun 2-0. For SCSI 1, these bits represent the logical unit number of the thread being established by the initiator or reestablished by the target. SCSI 2 uses them to identify a target routine or a logical unit, depending on the LUNTAR bit.

During an initiator Select with $\overline{\text{ATN}}$ and Transfer command, the TL2-TL0 bits of an incoming ID message are compared with the current TL2-TL0 bits in the LUN register before they are loaded. If the compare is unsuccessful, the “new LUN” interrupt is generated to indicate that a target is trying to reestablish a connection to a different logical unit. The incoming ID message is then loaded into the LUN register.

3.18 Command State Register

The command state register incrementally stores a code representing each successfully completed phase of a multiphase command. Each command has code definitions which are shown in the command description. The intent of this register is twofold:

1. To inform the microprocessor how far a multiphase command executed before some type of abnormal termination (something other than a function-complete interrupt) occurred so that the microprocessor can complete the transaction in low-level mode.
2. To begin execution of a Select with $\overline{\text{ATN}}$ and Transfer command at one of three possible entry points.

This register should only be accessed after a command has terminated due to completion, after an abnormal sequence, or after a pause command. The microprocessor can read the register after determining the type of interrupt that halted activity.

This register is set to all zeros by a master reset.

7	6	5	4	3	2	1	0
SDP	**0**	**0**	**0**	**CS3**	**CS2**	**CS1**	**CS0**

SDP: Save Data Pointer. This bit is set to 1 when the chip receives a save-data-pointer message while connected as an initiator and executing a Select with ATN and Transfer command. This bit is set to 0 by a master reset, by a function-complete interrupt, or by reading the command phase register. When the microprocessor detects that SDP is set to 1, the backup transfer counter should be read to determine the value of the data transfer counter at the time of the save-data-pointer message.

CS3-CS0: Command State 3-0. These bits represent an encoded phase of a multiphase command. See Section 4, Commands, for a description of the codes.

3.19 Transfer Counter Register

The transfer counter register is a 24-bit down-counter used to keep track of the data bytes traversing the chip-SCSI bus interface. It is composed of three 8-bit bytes, each of which may be addressed and read or written individually. The microprocessor loads the number of SCSI data bytes to be transferred into the transfer counter register prior to issuing a chip command to transfer data. When the counter decrements to zero, the data transfer is complete. If a data transfer is aborted, the transfer counter indicates how many bytes are left to transfer (see also the description of the backup transfer counter register).

The transfer counter register is loaded by writing to each of the three bytes. A write to the least-significant byte sets both of the higher-order bytes to all zeros. This allows the microprocessor to perform only one write to define the length for relatively short transfers (< 256 bytes). For longer transfers, the low-order byte should be written first, followed by writes to one or both of the higher-order registers. This register is set to all zeros by a master reset.

7	6	5	4	3	2	1	0
							lsb
msb							

3.20 Backup Counter Register

The backup counter register is used to save a copy of the contents of the transfer counter register whenever a save-data-pointer message is received by a chip in initiator mode during a Select with ATN and Transfer command. It may be read by the microprocessor to determine the amount of data successfully transferred over the SCSI bus in case an error occurs or a different target or LUN reconnects. The SDP (save data pointer) bit in the command phase register indicates whether a save-data-pointer message has been received and, thus, if the backup counter register has been updated. This register is set to all zeros by a master reset.

7	6	5	4	3	2	1	0
							lsb
msb							

3.21 Offset Count Register

The offset count register is used internally by the SBC to control the number of bytes transferred across the SCSI bus for synchronous data, extended messages, or command transfers. The SBC decodes the number of bytes to be transferred from the second byte of an extended message or the group code of a command. The offset counter can also be loaded by the microprocessor and used with manual transfer mode (selected by the control bit M/$\overline{A}$ in the command register) to override the automatic handling of these flows. The offset count register is an eight-bit register which allows a count of up to 255 to be stored. This register is set to all zeros by a master reset.

7	6	5	4	3	2	1	0
OC7	**OC6**	**OC5**	**OC4**	**OC3**	**OC2**	**OC1**	**OC0**

3.22 Supplemental Register Features

Registers 25 through 31 were used for SBC chip development and debug and are not intended for system use. However, the following specific features may be helpful to designers:

Register 26 (Test control register):

Bit 0 (FIFO): When set to 1, this bit enables an internal loop-back path from the Transmit FIFO to the Receive FIFO. Operation is as follows and can be monitored via the FIFO status bits in the Transfer status register. This bit is set to 0 by a master reset.

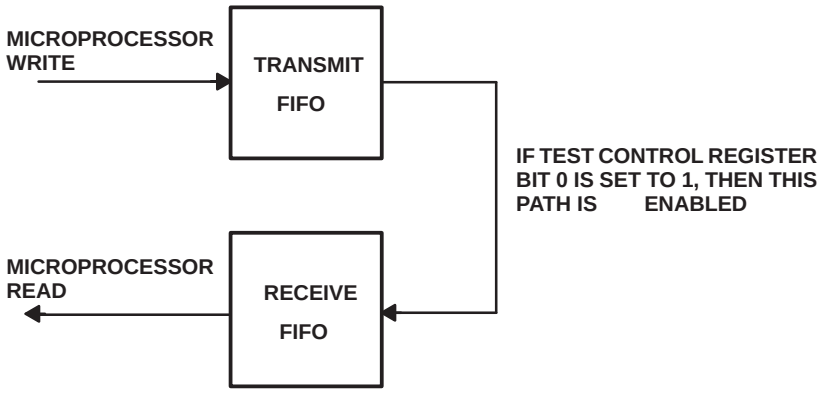


Figure 3–1. Internal Loop-Back Path for FIFO Test

Transmit FIFO Writes: A microprocessor-FIFO write causes a write to the transmit FIFO. If, prior to the microprocessor-FIFO write, the receive FIFO is not full and the transmit FIFO is not empty, then a write to the receive FIFO from the transmit FIFO and a read out from the transmit FIFO also occur. Thus, successive microprocessor-FIFO writes fill the receive FIFO first and then the transmit FIFO.

Receive FIFO Reads: A microprocessor-FIFO read causes a read from the receive FIFO. If, prior to the microprocessor-FIFO read, the receive FIFO is not full and the transmit FIFO is not empty, then a write to the receive FIFO from the transmit FIFO and a read out from the transmit FIFO also occur. Thus, successive microprocessor-FIFO reads empty the transmit FIFO first and then the receive FIFO.

If SCSI parity checking is enabled, data coming out of the transmit FIFO is checked for correct parity and an interrupt is generated if a parity error is detected.

Register 27 (Test points register 0):

Bit 0 (TFMTY): When set to 1, this bit indicates that the transfer FIFO is empty. This bit supplements the information provided by the TFF (transfer FIFO full) and TFHF (transfer FIFO half full) bits in the transfer status register. This bit is set to 0 by a master reset.

4 Commands

4.1 General

The SBC is driven by chip commands written by the local microprocessor to the SBC command register. The format and bit definitions of the command register are repeated below. The first three bits set modes for information transfer and are only valid for those types of commands (i.e., send, receive, and transfer).

7	6	5	4	3	2	1	0
DMA	**M/$\overline{\text{A}}$**	**DDIR**	**CC4**	**CC3**	**CC2**	**CC1**	**CC0**

DMA: Direct Memory Access. This bit controls the mode of data transfer from the SCSI bus to the microprocessor or DMA bus. When set to 0, the interface uses programmed I/O; when set to 1, DMA transfers are enabled. If DMA transfers are enabled, the microprocessor should not access the FIFOs until the command is complete.

M/$\overline{\text{A}}$: Manual/$\overline{\text{Automatic}}$. This bit allows the microprocessor to manually control the number of bytes transferred during a command or message phase. When this bit is set to 1, the count written to the offset counter by the microprocessor is used to determine the transfer length of the command or message phase. When this bit is set to 0, the group code of a command or the second byte of an extended message is decoded to determine a count value to be automatically loaded into the offset counter. Automatic mode eliminates the need for software to decode the command or message prior to completing the transfer. Manual mode may be used to complete a phase which was terminated prematurely.

DDIR: Data DIRection. This bit establishes the direction of data transfer during initiator-mode Select with- or Select without $\overline{\text{ATN}}$ and Transfer commands. When set to 0, the SBC expects a data-out phase; when set to 1, a data-in phase is expected. If the data phase set up by the target does not match that expected by the initiator, a bus service interrupt is generated and the command stops.

CC4–CC0: Command codes

4.2 Command Types

The chip commands are divided into two subsets: interrupting and noninterrupting. Both types of commands are executed immediately after they are written to the command register. As their names imply, they differ in whether or not an interrupt is generated to indicate command completion. The noninterrupting commands generally complete within a few clock cycles, while the interrupting commands can take from ten to several thousand clock cycles to complete.

The interrupting commands are subdivided into single-phase and multiphase commands. Phase refers to the SCSI bus phase encountered during command execution. The single-phase commands (also called low-level commands) execute bus management phases (e.g., arbitration, selection/reselection) or a single information transfer phase. The multiphase commands (also called high-level commands) combine information-transfer phases and bus management phases.

The SBC examines each command received from the microprocessor to determine if the command is valid (i.e., if it has been issued while the SBC is in an appropriate state to receive that command). A command issued at an inappropriate time causes the generation of an invalid command interrupt. The following table summarizes the chip commands, the valid states in which they can be issued, the resultant state if the command is successful, and whether they are noninterrupting (NI) or interrupting (I) commands. A description of each command follows the table.

4.3 Command Summary

4.3.1 Noninterrupting Commands

COMMAND CODE	COMMAND NAME	ISSUED STATE	RESULT STATE
00000	Chip Reset	ANY	D
00001	Disconnect	T, TO	D
00010	Pause	I, T	I, T
00011	Assert $\overline{ATN}$	I	I
00100	Negate $\overline{ACK}$	I	I
00101	Clear Receive FIFO	D, I, T	D, I, T
00110	Clear Transmit FIFO	D, I, T	D, I, T

STATE

D = Disconnected, I = Initiator, T = Target, TO = Time-Out

4.3.2 Single-Phase Interrupting Commands

COMMAND CODE	COMMAND NAME	ISSUED STATE	RESULT STATE
00111	SCSI Bus Reset	ANY	D
01000	Select with $\overline{ATN}$	D	I
01001	Select without $\overline{ATN}$	D	I
01010	Reselect	D	T
01011	(reserved)	–	–
01100	Receive Command	T	T
01101	Receive Data	T	T
01110	Receive Message Out	T	T
01111	Receive Unspecified Information Out	T	T
10000	Send Status	T	T
10001	Send Data	T	T
10010	Send Message In	T	T
10011	Send Unspecified Information In	T	T
10100	Transfer Information	I	I
10101	Transfer Pad	I	I
10110	(reserved)	–	–
10111	(reserved)	–	–

STATE

D = Disconnected, I = Initiator, T = Target, TO = Time-Out

4.3.3 Multiphase Interrupting Commands

COMMAND CODE	COMMAND NAME	ISSUED STATE	RESULT STATE
11000	Select with $\overline{\text{ATN}}$ and Transfer	D, I	D
11001	Select without $\overline{\text{ATN}}$ and Transfer	D	D
11010	Reselect and Receive Data	D	T
11011	Reselect and Send Data	D	T
11100	Wait for Select with $\overline{\text{ATN}}$ and Receive	D, T	T
11101	Wait for Select without $\overline{\text{ATN}}$ and Receive	D, T	T
11110	Conclude	T	D
11111	Link to Next Command	T	T

STATE

D = Disconnected, I = Initiator, T = Target, TO = Time-Out

4.4 Noninterrupting Commands

4.4.1 Chip Reset

Chip Reset halts all operations and returns the chip to a master reset state. The SBC releases all SCSI bus lines and the receive and transmit FIFO pointers are reset.

4.4.2 Disconnect

In response to a disconnect command, the SBC releases the SCSI $\overline{\text{BSY}}$ signal and exits target mode. The disconnect command is also used following a selection/reselection time-out interrupt to cause the SCSI $\overline{\text{SEL}}$ signal to be released, indicating the end of the selection/reselection attempt by freeing the SCSI bus.

4.4.3 Pause

The pause command is used to halt execution of certain chip commands. Pause capability gives the microprocessor additional control over commands with long execution times. In cases such as arbitration, the pause command may not immediately be recognized. Thus, once a pause command is issued, the command register should not be written again until an interrupt indicating command completion or termination is received. Case-by-case use of the pause command is as follows:

Arbitration Phase Pause: Issuing a pause command during any chip command performing arbitration and selection halts that command and generates a halted interrupt if arbitration is lost or if the SCSI bus is being used by another device. For example, if a bus-free phase has been detected and the chip is actively involved in the arbitration process, a pause does not occur unless and until arbitration is lost. If arbitration is won, the pause command has no effect on the arbitration process.

Selection/Reselection Time-Out: Issuing a pause command during any chip command performing selection/reselection with a selection/reselection time-out value of zero (see Section 3.13, Selection/Reselection time-out register) causes a time-out to occur when the selection/reselection time-out begins. The pause command causes the selection/reselection ID to be released on the SCSI bus and a selection/reselection abort time to be started. If no $\overline{\text{BSY}}$ response is detected within the abort time, the time-out interrupt is generated. This capability allows indefinite arbitration/selection time-outs to be used by the microprocessor for cases in which the time-out register does not provide a long enough delay.

Data Phase Pause: A pause command can be used to halt any data phase. Data phase pause capability allows the microprocessor to halt a long data transfer phase in case of unexpected error situations.

1. Initiator Asynchronous: Following receipt of the pause command, the SBC performs one $\overline{\text{REQ}}$/$\overline{\text{ACK}}$ handshake, then generates a halted interrupt. The transfer counter contains the number of bytes left to transfer. A transfer pad command can be used to finish the data phase.

2. Target Asynchronous: Following receipt of the pause command, the SBC performs one REQ/ACK handshake, then generates a halted interrupt. The target may reissue a data transfer command to continue the transfer or may issue a command to change phases. If the data phase is not completed, a FIFO clear command may be needed prior to performing another type of transfer.
3. Initiator Synchronous: Pauses by holding off ACK (even if the REQ/ACK offset is nonzero) and then generates a halted interrupt. The transfer counter contains the number of bytes left to transfer. A transfer pad command can be used to finish the data phase.
4. Target Synchronous: Pauses when the REQ/ACK offset decrements to zero and generates a halted interrupt. The target may reissue a data transfer command to continue the transfer or may issue a command to change phases. If the data phase is not completed, a FIFO clear command may be needed prior to performing another type of transfer.

4.4.4 Assert ATN

The Assert ATN command is used in the initiator state to inform the target that the initiator has a message pending. The ATN line is asserted immediately upon issuance of the command. If ATN is already asserted when the command is issued, no action results. This command is used when the initiator wishes to reject a message from the target. ACK is held active during the last message byte until a Negate ACK command is issued. The Assert ATN command may be issued prior to Negate ACK if the initiator wishes to reject the message. This allows the target to determine which message in a long string of messages is being refused.

ATN is released automatically before asserting ACK for the last byte of any outbound message (or messages) if the HAAM bit in the transfer command is set to 0. If the HAAM bit is set to 1, the ATN line is held active to request that the target perform another message-out phase.

4.4.5 Negate ACK

The Negate ACK command is used in the initiator state to complete a message-in transfer. ACK is held asserted following the last byte of a successfully completed transfer command during message-in phase. This allows the microprocessor to evaluate the message prior to releasing ACK. The Assert ATN command may be issued before the Negate ACK command in order to reject the message. The Negate ACK command must be issued to complete the REQ/ACK handshake.

4.4.6 Clear Receive FIFO

The Clear Receive FIFO command is used to reset the receive FIFO pointer.

4.4.7 Clear Transmit FIFO

The Clear Transmit FIFO command is used to reset the transmit FIFO pointer. For example, in multithread initiator applications, this command is used to clear unsent SCSI data or commands from the transmit FIFO when a connection to a different logical thread is established.

4.5 Single-Phase Interrupting Commands

4.5.1 SCSI Bus Reset

The SCSI Bus Reset command causes assertion of the SCSI bus RST signal for a period of 200 µs; a function-complete interrupt is then generated. The SCSI Bus Reset command has the same effect as a Chip Reset command except that the FIFO pointers are not reset and only selected bits in the register file are reset (see Section 3, Registers, for details).

4.5.2 Select with ATN

The Select With ATN command enables the SBC to start the arbitration and selection phases in order to establish a connection with a target and become an initiator. A target ID must be loaded into the destination ID register and the initiator ID loaded into the self ID register prior to command issuance by the

microprocessor. The chip then begins arbitration during the next bus-free phase. If arbitration is lost, the chip tries again to arbitrate when the next bus-free phase is detected. If arbitration is won, the selection phase proceeds with the chip asserting both SEL and ATN, placing both self (initiator) and destination (target) IDs onto the SCSI data bus, and releasing BSY. If the target responds within the selection time-out, the chip assumes the initiator role. The command terminates with a function-complete interrupt when the first REQ is received from the target. If the bus phase of the incoming REQ is not message-out, then ATN is released. If the target does not respond within a selection time-out, the self and destination IDs (known together as the "selection ID") are released and a selection abort time-out is started. If the target does not respond within the selection abort time, a time-out interrupt is generated (see disconnect command). The pause command can be issued to terminate this command prematurely (see pause command). Also, if the SBC is selected/reselected by another SCSI device, this command is terminated with a selected/reselected interrupt.

4.5.3 Select without ATN

Select without ATN is identical to Select with ATN except that ATN is not asserted during the selection phase, indicating that the initiator does not support the optional SCSI messages during the connection.

4.5.4 Reselect

The Reselect command is issued to place a disconnected device into the target mode by reselecting an initiator. The initiator ID must be loaded into the destination ID register and the target ID loaded into the self ID register prior to issuing the command. The chip begins arbitration at the next bus-free phase. If arbitration is lost, the chip tries again to arbitrate at the next bus-free phase. If arbitration is won, the chip begins the reselection phase by asserting SEL and I/O, placing both self (target) and destination (initiator) IDs on the bus, and then releasing BSY. If the initiator responds by asserting BSY within the selection time-out, the chip releases SEL and assumes the target role by asserting BSY. The command finishes with a function-complete interrupt. If the initiator does not respond within the selection time-out), the self and destination IDs (known together as the "selection ID") are released and a selection abort time-out is started. If the initiator does not respond within the selection abort time, a time-out interrupt is generated (see disconnect command). The pause command can be issued to terminate this command prematurely (see pause command). Also, selection/reselection of the SBC by another SCSI device terminates this command with a selected/reselected interrupt.

4.5.5 Receive

The Receive Data, Receive Command, Receive Unspecified Information, and Receive Message commands are used by devices in the target mode to set the bus phase lines for a transfer from the initiator to the target and then to carry out the REQ half of each transfer handshake. SCSI bus phase lines are held stable throughout the transfer and until a subsequent command modifies them as shown in the following table:

SCSI BUS PHASE DEFINITION			
COMMAND	**MSG**	**C/D**	**I/O**
Receive data	0	0	0
Receive command	0	1	0
Receive unspecified information	1	0	0
Receive message	1	1	0

The following events result in completion or termination of the Receive commands (note: a Chip Reset command or assertion of the SBC $\overline{MR}$ line causes termination of any command):

TYPE OF RECEIVE	INTERRUPT BITS		EVENT
	SET TO 1	REGISTER	
Receive data/ unspecified info	FC	FISR	The transfer counter decrements to zero.
	HALT, PE	EISR, EISR	A parity error is detected with the Halt On Parity bit set to 1 and the transfer counter is not zero.
	HALT, ATN	EISR, FISR	$\overline{ATN}$ condition occurs with the Halt On $\overline{ATN}$ Bit set.
	HALT	EISR	A Pause command is issued.
Receive command	FC	FISR	The offset counter decrements to zero.
	HALT, PE	EISR, EISR	A parity error is detected on the first command byte.
Receive message	FC	FISR	The offset counter decrements to zero.
	HALT, PE	EISR, EISR	A parity error is detected on the first or second message byte.
All transfers	SRST	EISR	A SCSI bus reset condition was detected.

Receive Data transfers can be asynchronous or synchronous and can be received through either the DMA or microprocessor interface. Synchronous mode is selected by storing a nonzero offset in the synchronous control register; a zero offset selects asynchronous mode. DMA mode is selected by setting the DMA bit in the command register to 1 when the command is issued; a DMA bit set to 0 selects programmed I/O mode. The transfer counter controls the amount of data received, and it must be loaded before the receive command is issued. Received data is stored in the receive FIFO. FIFO flags in the transfer status register can be monitored by the microprocessor to determine when data is available during programmed I/O mode.The DMA interface transfers the data from the receive FIFO to memory if DMA mode is requested. A pause command can be used to halt a Receive Data command. Alternately, the following control register bits are used to halt a Receive Data command conditionally:

HPE: When this bit is active, the Receive Data command is halted when a parity error is detected on a data byte received through the SCSI interface. The halt is implemented in the same manner as by a Pause command.

HA: When this bit is active, the Receive Data command is halted when an $\overline{ATN}$ condition is detected. The halt is implemented in the same manner as by a Pause command.

Receive Unspecified Information transfers are identical to data transfers except that synchronous transfer mode is not allowed.

Receive Command transfers always use programmed I/O. The source of the command length information depends on the value of the M/$\overline{A}$ (manual/automatic) bit in the command register. If M/$\overline{A}$ is set to 1, the microprocessor must manually load a command length value into the offset counter. If M/$\overline{A}$ is set to 0, a command length value based on the group code embedded in the first byte of the command (see table below) is automatically loaded into the offset counter. The offset counter is used as a transfer counter for nondata transfers and is decremented once for each byte received over the SCSI bus. In automatic mode, if a parity error is detected on the first byte of the SCSI command, the chip command is halted because the group code cannot reliably be determined. Parity errors detected on later automatic-mode bytes or in manual mode do not halt the chip command but are reported by a parity error interrupt when the command completes. Received bytes are stored in the receive FIFO.

AUTOMATIC COMMAND-TRANSFER-LENGTH DECODING	
GROUP CODE	**NO. OF BYTES AUTOMATICALL TRANSFERRED**
0	6 bytes
1	10 bytes
5	12 bytes
2, 3, 4, 6, 7	2 bytes

Receive Message transfers always use programmed I/O. The source of the message length information depends on the value of the M/$\overline{A}$ (manual/automatic) bit in the command register. If M/$\overline{A}$ is set to 1, the microprocessor must load a message length value into the offset counter prior to issuing the receive command. If M/$\overline{A}$ is set to 0, the SBC determines message length automatically from the message: either one for single-byte messages or as specified in the second byte for extended messages. For extended messages, the second message byte is loaded into the offset counter. The offset counter is used as a transfer counter for nondata transfers and is decremented once for each byte received over the SCSI bus. In automatic mode, if a parity error is detected during the first or second byte of the message, the message length cannot reliably be determined. In this case, the SBC performs dummy transfers (without regard for FIFO status) until $\overline{ATN}$ is released by the initiator. Clear Receive FIFO should be issued following this case to clear any dummy transfers out of the receive FIFO. Parity errors detected on later bytes or in manual mode do not affect the transfer but are reported by a parity error interrupt when the command completes the message transfer. All bytes received are stored in the receive FIFO.

4.5.6 Send

The Send Data, Send Status, Send Unspecified Information, and Send Message commands are used by devices in the target mode to set the bus phase lines for a transfer from the target to the initiator and then to carry out the $\overline{REQ}$ half of the transfer handshake. SCSI bus phase lines are held stable throughout the transfer and until a subsequent chip command modifies them as shown in the following table:

SCSI PHASE DEFINITION			
COMMAND	**MSG**	**C/D**	**I/O**
Send data	0	0	1
Send status	0	1	1
Send unspecified information out	1	0	1
Send message in	1	1	1

Any of the following events result in completion or termination of the Send commands (note: a Chip Reset command or assertion of the SBC $\overline{MR}$ line terminates any command):

TYPE OF SEND	INTERRUPT BITS		EVENT
	SET TO 1	**REGISTER**	
Send data/unspecified info	FC	FISR	The transfer counter decrements to zero.
	HALT, PE	EISR	A parity error is detected with the Halt On Parity
	HALT, ATN	EISR, FISR	An $\overline{ATN}$ condition occurs with the Halt On $\overline{ATN}$ Bit set to 1.
	HALT	EISR	A Pause command is issued.
Send status	FC	FISR	The status byte is sent.
Send message	FC	FISR	The offset counter decrements to zero.
All transfers	SRST	EISR	A SCSI bus reset condition is detected.

Send Data transfers can be asynchronous or synchronous and can be sent through either the DMA or microprocessor interface. Synchronous mode is selected by storing a nonzero offset in the synchronous control register; a zero offset selects asynchronous mode. DMA mode is selected by setting the DMA bit in the command register to 1 when the command is issued; a DMA bit set to 0 selects programmed I/O mode. The transfer counter controls the amount of data sent, and it must be loaded before the send command is issued. Data to be sent is stored in the transmit FIFO. FIFO flags in the transfer status register can be monitored by the microprocessor to determine when there is room in the transmit FIFO for more data during programmed I/O mode. The DMA interface transfers the data from memory to the transmit FIFO if DMA mode is requested. The following control register bits may be used to halt a Send Data command conditionally:

HPE: When this bit is active, the Receive Data command is halted when a parity error is detected on a data byte received through either the microprocessor or DMA interface. The halt is implemented in the same manner as by a Pause command.

HA: When this bit is active, the Receive Data command is halted when an $\overline{\text{ATN}}$ condition is detected. The halt is implemented in the same manner as by a Pause command.

Send Unspecified Information transfers are identical to data transfers except that synchronous transfer mode is not allowed.

Send Status transfers always use programmed I/O. The status length is always one byte, and this byte must be stored in the transmit FIFO prior to issuing the command.

Send Message transfers always use programmed I/O. The source of the message length information depends on the value of the M/$\overline{\text{A}}$ (manual/automatic) bit in the command register. If M/$\overline{\text{A}}$ is set to 1, the microprocessor must load a message length value into the offset counter prior to issuing the send command. If M/$\overline{\text{A}}$ is set to 0, the SBC determines message length automatically from the message: either one for single-byte messages or as specified in the second byte for extended messages. For extended messages, the second message byte is loaded into the offset counter. The offset counter is used as a transfer counter for nondata transfers and is decremented once for each byte sent over the SCSI bus. In automatic mode, the first byte of a single-byte message and the first three bytes of an extended message must be loaded prior to issuing the Send command. This assures that the bytes used to determine message length are valid. If a parity error is detected while preloading any message bytes, an interrupt is generated and the microprocessor must clear the transmit FIFO and reload the message bytes. Since parity errors do not halt a message transfer, preloading the transmit FIFO with the entire message prior to issuing the send command is recommended in order to insure message integrity.

4.5.7 Transfer Info

The Transfer Info command is used by a device in the initiator mode to send or receive data, commands, status, messages, and unspecified information. This command is issued in response to a SCSI bus phase change. The bus phase change is reported to the microprocessor as a bus service interrupt after the leading edge of $\overline{\text{REQ}}$. SCSI bus phase information is then available to the microprocessor through the bus phase register. The Transfer command causes the initiator to complete the $\overline{\text{ACK}}$ half of each $\overline{\text{REQ}}$–$\overline{\text{ACK}}$ handshake. A function-complete interrupt is generated when the transfer is complete. A halted interrupt is generated if the chip command is terminated prior to its completing the transfer (e.g., due to a parity error or a bus service interrupt). If the AAPE bit in the control register is set to 1, $\overline{\text{ATN}}$ is automatically asserted if a parity error is detected on data coming into the part through any of the SCSI, DMA, or microprocessor interfaces while a transfer command is active (this is true for transfers during any SCSI bus phase).

Any of the following events result in completion or termination of the Transfer Command and generation of the corresponding interrupt (note: a Chip Reset command or assertion of the SBC $\overline{MR}$ line terminates any command):

TYPE OF TRANSFER	INTERRUPT BITS		EVENT
	SET TO 1	REGISTER	
Data/unspecified info transfer	FC	FISR	The transfer counter decrements to zero.
	HALT, PE	EISR	A parity error is detected with the Halt On Parity bit set to 1 and the transfer counter is nonzero.
	HALT	EISR	A Pause command is issued and the transfer halts with a nonzero transfer counter.
Command out transfer	FC	FISR	The offset counter decrements to zero.
Message-in transfer	FC	FISR	The offset counter decrements to zero.
	PE, HATL, BS	EISR, EISR, FISR	A parity error was detected in the first or second byte of the message and the target has changed bus phase.
Message-out transfer	FC	FISR	The offset counter decrements to zero.
Status transfer	FC	FISR	One byte is received.
All transfers	BUS	FISR	The SCSI bus phase changes and a $\overline{REQ}$ is issued in the new bus phase.
	DIS	FISR	$\overline{BSY}$ is released by the target, resulting in a disconnect.
	SRST	FISR	A SCSI bus reset condition is detected.

Data Transfers may be asynchronous or synchronous and can use either the DMA or microprocessor interface. Synchronous mode is selected by storing a nonzero offset in the synchronous control register; a zero offset selects asynchronous mode. DMA mode is selected by setting the DMA bit in the command register to 1 when the command is issued. A DMA bit set to 0 selects programmed I/O mode.

The transfer counter controls the amount of data sent or received; it must be loaded with the number of data bytes to be transferred before the transfer command is issued. During programmed I/O mode, FIFO flags in the transfer status register can be monitored by the microprocessor to determine when the transmit FIFO can be loaded with more data for data-out transfers or when more data is available in the receive FIFO for data-in transfers. The DMA interface transfers the data between memory and the FIFOs if DMA mode is requested.

The HPE bit in the control register can be used to halt a data transfer conditionally if a parity error is detected on a data byte received through either the DMA or microprocessor interface during a data-out transfer or through the SCSI interface during a data-in transfer. Transfers are halted in the same manner as by a pause command. The transfer pad command can be issued to complete a halted transfer.

Unspecified Information transfers are identical to data transfers except that synchronous transfer mode is not allowed.

Command Out transfers always use programmed I/O. The source of the command length information depends on the value of the M/$\overline{A}$ bit in the command register. If M/$\overline{A}$ is set to 1, the microprocessor must manually load a command length value into the offset counter. If M/$\overline{A}$ is set to 0, a command length value based on the group code embedded in the first byte of the command (see table below) is automatically loaded into the offset counter. The offset counter is used as a transfer counter for nondata transfers and is decremented once for each command byte sent over the SCSI bus. In automatic mode, the first two bytes of the SCSI command must be loaded into the transmit FIFO prior to issuing the transfer command. If a parity error is detected while preloading any command bytes, a parity error interrupt is generated and the microprocessor must clear the transmit FIFO and reload the bytes. This action assures that the SBC has a valid first byte from which to determine the command length. Since parity errors do not halt a command-out

transfer, preloading the transmit FIFO with the entire command prior to issuing the Transfer command is recommended in order to insure command integrity. A bus-service interrupt is the only condition that halts a command-out transfer prematurely.

AUTOMATIC COMMAND-TRANSFER-LENGTH DECODING	
GROUP CODE	**NO. OF BYTES AUTOMATICALLY TRANSFERRED**
0	6 bytes
1	10 bytes
5	12 bytes
2, 3, 4, 6, 7	2 bytes

Status-In transfers always use programmed I/O. The status length is always one byte and this byte is stored in the receive FIFO after being received from the target.

Message-in transfers always use programmed I/O. The source of the message length information depends on the value of the M/A (manual/automatic) bit in the control register. If M/A is set to 1, the microprocessor must load a message length value into the offset counter prior to issuing the transfer command. If M/A is set to 0, the SBC determines message length automatically from the message: either one for single-byte messages or as specified in the second byte for extended messages. For extended messages, the second message byte is automatically loaded into the offset counter. The offset counter is used as a transfer counter for nondata transfers and is decremented once for each byte received over the SCSI bus. All received bytes are stored in the receive FIFO. Following a function-complete interrupt, the chip holds ACK asserted to allow the received message to be evaluated. The Negate ACK command must be issued to complete the message transfer. The Assert ATN command can be issued prior to the Negate ACK command if the message is to be rejected.

If a parity error is detected during the first or second byte of a message, the message length cannot reliably be determined. The opportunity to issue the Negate ACK command is not provided for this case. The SBC performs dummy transfers (without regard for receive FIFO status) until the target changes phases. The Clear Receive FIFO command should then be issued to clear any dummy transfers out of the receive FIFO. It is recommended that the AAPE bit in the control register be set to 1 to notify the target of an error before a phase change occurs. Parity errors detected on later bytes or in manual mode do not affect message length determination, so the transfer completes with ACK asserted. However, both the parity error and function-complete interrupts are generated.

Message-out transfers always use programmed I/O. The source of the message length information depends on the value of the M/A (manual/automatic) bit in the control register. If M/A is set to 1, the microprocessor must load a message length value into the offset counter prior to issuing the transfer command. If M/A is set to 0, the SBC determines message length automatically from the message: either one for single-byte messages or as specified in the second byte for extended messages. For extended messages, the second message byte is automatically loaded into the offset counter. The offset counter is used as a transfer counter for nondata transfers and is decremented once for each byte received over the SCSI bus. In automatic mode, the first byte of a single-byte message or the first three bytes of an extended message must be loaded prior to issuing the transfer command. This assures that the bytes used to determine message length are valid. If a parity error is detected while preloading any message bytes, an interrupt is generated and the microprocessor must clear the transmit FIFO and reload the message bytes. Parity errors do not halt a message transfer, so preloading the FIFO with the entire message prior to issuing the transfer command is recommended to insure message integrity. If ATN is asserted and the HAAM (hold ATN after message out) bit is set to 0 in the control register, ATN is released prior to assertion of SCSI ACK for the last message byte; otherwise, ATN remains asserted.

4.5.8 Transfer Pad

The Transfer Pad command works in the same manner as the Transfer Info command for data-in and data-out phases, with the following exceptions:

For data-out phase, the contents of the top of the transmit FIFO at the time the command is issued are transferred repeatedly. If a specific byte is to be transferred repeatedly, it must be written to the transmit FIFO (when the FIFO is empty) using programmed I/O before the command is issued. The DMA bit in the command register should be set to 0 for this command.

For data-in phase, the incoming bytes are not checked for parity, nor are they written to the receive FIFO.

4.6 Multiphase Interrupting Commands

These commands execute a prescribed sequence of SCSI bus activities. The SBC chip is said to be operating in high-level mode when executing a multiphase command. In this mode, low-level chip commands are chained together and interrupts are handled internally until the command completes or until a deviation from the expected flow occurs. Interrupts are generated to indicate command termination or completion, and the command state code (a code representing the last successfully executed command phase) is then available in bits 3:0 of the command state register. Low-level SBC commands can be used to complete a SCSI connection if a multiphase command is terminated prematurely. Multiphase commands greatly reduce the number of interrupts generated during a SCSI bus transaction. Flowcharts for each of these commands are included after the text description along with a command state code/interrupt table describing interrupts and their relation to the command state register codes.

4.6.1 Select with $\overline{\text{ATN}}$ and Transfer

This command automatically handles the common SCSI initiator sequence used to arbitrate for the bus, select a target, send an ID message, send a SCSI command, send or receive data, receive status, and receive a command-complete message. Multiple Save-Data-Pointer messages, disconnects, and reconnects are also handled automatically throughout the data transfer. Figure 4–1 is a flowchart of the Select with $\overline{\text{ATN}}$ and Transfer command. Successful command completion, a deviation from the expected flow, or error conditions result in an interrupt. The interrupt registers and the command state register can be used to determine the cause of the interrupt as shown in the command state/interrupt table at the end of this section. When the command is issued, the control register M/$\overline{\text{A}}$ bit must be set to 0 (automatic mode) and the data direction and DMA bits in the command register should be set appropriately.

4.6.1.1 Command Setup

Prior to issuing this command, the host must load the following registers:

REGISTER NAME	COMMAND PHASE AFFECTED
Command state	Command entry point
Control	Reselection (RE), Data (HPE, AAPE), Disconnect (HD)
Self ID	Arbitration/selection and reselection
Destination ID	Selection and reselection
Time-Out	Selection
Target LUN	ID message transfer, send and receive
Transfer counter	Data
Synchronous control	Data (only if synchronous transfers are used)
Byte stack control	Data (only if byte stack mode or demand mode is used)
Transmit FIFO	Command - the Command Description Block (CDB) must be written into the transmit FIFO prior to issuing the command.
Parity control	All

4.6.1.2 Command Entry Points

The command state register value determines the entry point into the Select with $\overline{ATN}$ command as shown in the following table. Upon command termination, the command state register value identifies the last successfully completed phase. It may be desirable to restart the Select with $\overline{ATN}$ and Transfer command after the termination condition has been handled in low-level mode.

COMMAND ENTRY POINT	COMMAND STATE CODE	COMMAND ENTRY DESCRIPTION
1	0-1, 3-7, A, D, F	Arbitration/selection (beginning of flow).
2	2	This entry point is used when the target requests a resend of the ID message instead of going to command phase. The processor handles the resend in low-level mode and then continues the Select with $\overline{ATN}$ command by reissuing the command after a bus service interrupt for command phase is received.
	E	This entry point also eases the use of linked commands. The command phase code is set to hexadecimal E at the completion of a linked command. The Select with $\overline{ATN}$ and Transfer command can be reissued to send the linked command when a bus service interrupt for the command phase is received.
3	8	This entry point can be used following command termination due to a disconnect with the halt–on-disconnect bit set to 1 in the control register or after the SBC is reselected by a different target. The command can be reissued in response to a bus-service interrupt following reselection, low-level mode ID message receipt, and update of the register file.
	9	This entry point can also be used following command termination due to a new LUN interrupt. The command can be reissued in response to a bus service interrupt after the register file has been updated to handle the transfer for the new LUN.

4.6.1.3 Arbitration/Selection

As in the Select with $\overline{ATN}$ command, the SBC arbitrates for the bus and attempts to select the target in the destination ID register. If successful, the command state code is set to 1 and the SBC continues command execution.

4.6.1.4 ID Message Transfer

Since $\overline{ATN}$ is asserted during the selection process, the first phase expected is a message-out phase, during which an identify message is sent. This message consists of the pattern “1rLUNREG”, where “r” is the value of the RE bit in the control register and LUNREG is the value of target LUN register bits 5-0. Parity is generated for the SCSI bus according to the SPE bit in the parity control register. Upon transfer of this message, the $\overline{ATN}$ line is released and the command state code is set to 2.

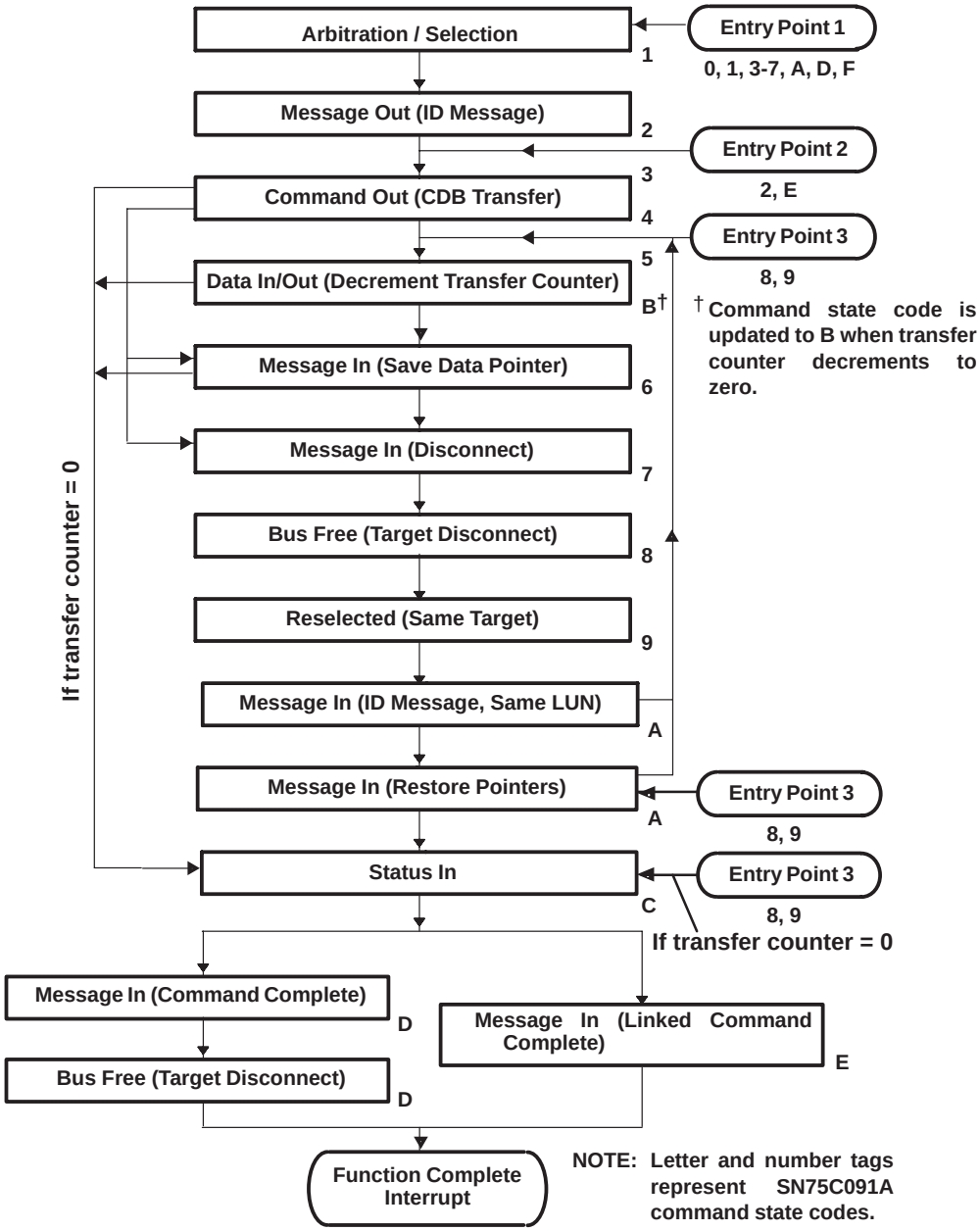


Figure 4–1. Select with $\overline{\text{ATN}}$ and Transfer Command

4.6.1.5 Command Data Block Transfer

The command state code is updated to 3 when the target asserts $\overline{\text{REQ}}$ for the SCSI command phase. The CDB that was preloaded into the transmit FIFO is transferred in the same manner as in low-level mode. The command state code is updated to 4 after the successful transfer of all the CDB bytes and the SBC continues command execution.

4.6.1.6 Data Transfer

The command state code is updated to 5 when the target asserts $\overline{\text{REQ}}$ to begin the data phase. Data transfers proceed as in the Transfer Info command; however, they may be interlaced with certain message-in phases used to save data pointers and to perform disconnect/reconnect cycles. If the proper flow is followed, any number of save-data-pointer messages or disconnect/reconnect cycles may occur without processor intervention. When the transfer counter decrements to zero, the data transfer is complete and the command state code is set to hexadecimal B.

HPE and AAPE control register bits affect data transfer as follows:

HPE (halt on parity error): This bit causes the data transfer to halt on a parity error as defined for the transfer command. The command is terminated with halted- and parity-error interrupts. A transfer pad command can be used to complete the data phase. Note that if a parity error occurs with the HPE bit set to 0, the Select with $\overline{\text{ATN}}$ and Transfer command is terminated when the data transfer is complete or when the target changes phase. A parity error and a halted interrupt are generated. This prevents the data pointer from being saved after a parity error has occurred.

AAPE (assert $\overline{\text{ATN}}$ on parity error): $\overline{\text{ATN}}$ is asserted if a parity error is detected as defined for the transfer command. For the Select with $\overline{\text{ATN}}$ and Transfer command, AAPE only affects data transfers because the command is terminated prior to acknowledging a message-in or status byte if a parity error condition is observed.

4.6.1.7 Disconnect/Reselection

A target disconnect causes the command state code to be set to 8. If the HD bit in the control register is set to 1, the command terminates with a disconnected interrupt and the command state code is set to 8. If the HD bit is set to 0 , the SBC waits indefinitely for a reconnect by the same target. If a different target reconnects, a reselected interrupt is generated. If the SBC is selected, a selected interrupt is generated. In either case, the command state code retains the value 8. Note that occurrence of a disconnect at any time other than following a disconnect message or command-complete message is a SCSI catastrophic error condition. In this case, the control error interrupt is generated; a master reset and/or SCSI reset is required for recovery.

4.6.1.8 Message-In Transfers

Only the single-byte messages shown below are handled by the Select with $\overline{\text{ATN}}$ and Transfer command. Each message is decoded and evaluated prior to acknowledgement. This allows the SBC to acknowledge only error-free messages appropriate for the command flow. An acceptable message is acknowledged and the appropriate action is taken. If the message is not acceptable, a bus service interrupt is generated and the processor must receive the message using the Transfer Info command. For example, a message with a parity error is not received via the Select with $\overline{\text{ATN}}$ and Transfer command and no parity error interrupt is generated until the message is received using the Transfer Info command.

Save Data Pointer Message (hexadecimal 02): A Save Data Pointer message is acknowledged but not stored into the receive FIFO. Its receipt causes loading of the current transfer counter value into the backup counter register. The command state code is updated to 6 and the SDP bit is set to 1. If a data error occurs, the processor uses the SDP bit to determine if the backup counter needs to be read in order to redo a data transfer using the most recently saved data pointer.

Disconnect Message (hexadecimal 04): Receipt of a Disconnect message precedes a legal disconnect. The Disconnect message is acknowledged but not stored in the receive FIFO, and the command state code is updated to 7. The command accepts the disconnect message only if it occurs immediately after the SCSI CDB transfer or after a save-data-pointer message. Reconnection implies restoration of pointer values; however, since receipt of the save-data-pointer message automatically saves the transfer counter, no microprocessor intervention is required.

ID Message (hexadecimal 80-FF): An ID message is received following reselection. If the ID message LUN matches the LUN register, the message is acknowledged but not loaded into the FIFO. The

command state code is updated to hexadecimal A. If the LUNs do not match, the ID message is loaded into the LUN register and acknowledged, the command state code is left at 9, and the command is terminated with a new LUN interrupt.

Restore Pointers Message (hexadecimal 03): A Restore Pointers message can be received following receipt of an ID Message. The Restore Pointers message is not stored in the receive FIFO. Since SCSI pointer restoration is implied by the reselection, this message is not required; however, support is provided for those targets that do send the Restore Pointers message.

Command Complete Message: The Command Complete message is received and stored in the receive FIFO following a status transfer. If the command complete message is hexadecimal 00 (i.e., command completed normally), the command state code is updated to hexadecimal D and the SBC waits for the target to disconnect before generating a function-complete interrupt. If a Linked Command Complete message is received, the command state code is updated to hexadecimal E and a function-complete interrupt is generated.

4.6.1.9 Status Transfer

SCSI status is stored in the receive FIFO and the command state code is updated to hexadecimal C. As with the message-in transfers, if a parity error is observed on the status byte, a bus service interrupt is generated and the status must be received using the Transfer Info command (no parity interrupt is generated until the status is received). The Select with $\overline{\text{ATN}}$ and Transfer command allows a status transfer for the four cases detailed below:

1. After the command phase, the target can send status if the transfer counter is zero. This allows handling of commands which do not require a data transfer.
2. After the ID message of a reconnect, the target can send status if the transfer counter is zero. This allows a target to disconnect while it stores a buffer to tape, e.g., before it has determined the transaction status.
3. A status phase can be entered after a data phase when the transfer counter reaches zero.
4. A status phase can be entered after a save data pointer message if the transfer counter is zero.

Any of the following events result in completion or termination of the Select with $\overline{\text{ATN}}$ and Transfer command and generation of the corresponding interrupt. SCSI bus reset and/or control error interrupts can occur during any command phase. (Note: a Chip Reset command or assertion of the SBC $\overline{\text{MR}}$ line terminates any command.)

SELECT-WITH-$\overline{\text{ATN}}$-AND-TRANSFER COMMAND STATE CODES AND INTERRUPTS

COMMAND STATE CODE (HEX)	LAST SUCCESSFUL PHASE	BIT(S) SET TO 1	REGISTER	CAUSE
0	Arbitration/selection unsuccessful	T-O	EISR	Target did not respond.†
		SEL	FISR	Chip selected as a target by an initiator.†
		RSL	FISR	Chip reselected as an initiator by a target.†
		HALT	EISR	Pause command halted arbitration.†
1	Selection successful	BUS	FISR	No message-out phase.
2	Identify message sent	BUS	FISR	No command-out phase.

† Following these events, a Clear Transmit FIFO command should be issued to clear the unsent SCSI command bytes from the transmit FIFO.

SELECT-WITH-$\overline{\text{ATN}}$-AND-TRANSFER COMMAND STATE CODES AND INTERRUPTS (Continued)

COMMAND STATE CODE (HEX)	LAST SUCCESSFUL PHASE	BIT(S) SET TO 1	REGISTER	CAUSE
3	Start of CDB transfer	BUS, HALT	FISR, EISR	Command phase wrong length.
4	CDB transfer complete	BUS	FOSR	No status, data, or message-in phases. If status, TC > 0. If data, I/O did not match DDIR or TC = 0. If message in, message was not SDP or disconnect.
		HALT	EISR	Pause command halted chip command.
5	Data tramsfer started or continued	BUS	FISR	No status, data, or message-in phases. If status, TC > 0. If data, I/O did not match DDIR or TC = 0. If message in, message was not SDP.
		PE, HALT	EISR, EISR	Parity error during data transfer.
		HALT	EISR	Pause command halted data transfer.
6	Save data pointer message received (SDP bit is set to 1 in bus phase register)	BUS	FISR	No data or message-in phases. If data, I/O did not match DDIR. If message in, message was not disconnect.
		HALT	EISR	Pause command halt.
7	Disconnect message received	BUS	FISR	Any new bus information phase.
8	Target disconnected	SEL	FISR	Chip selected as target.‡
		RSL	FISR	If DIS, then any target reselected. Otherwise, a new target reselected.‡
		DIS	FISR	Target disconnected with HD set to 1 in control register.‡
		HALT	EISR	Pause command halt.
9	Original target reselected	BUS	FISR	No message-in phase, or if message in, message was not ID.
		NEWLN	EISR	New LUN reconnected.‡
A	Correct ID message received	BUS	FISR	No status, data, or message-in phase. If status, TC > 0. If data, I/O did not match DDIR or TC = 0. If message in, message was not RP.
B	Data transfer completed (TC = 0) (subset of Code 5)	BUS	FISR	No status or message-in phase. If message in, message was not SDP.
C	Status byte received	BUS	FISR	No message-in phase, or if message in, message was not CC, LCC, or LCCwF.

† Following these events, a Clear Transmit FIFO command should be issued to clear the unsent SCSI command bytes from the transmit FIFO.

‡ Following these events, a Clear Transmit FIFO command should be issued to clear the unsent data bytes from the transmit FIFO to prepare the SBC for a transaction with a different logical thread. The transfer counter and/or backup counter register values can be used to update SCSI data pointers so that bytes that were left in the FIFO can be resent later.

SELECT-WITH-$\overline{\text{ATN}}$-AND-TRANSFER COMMAND STATE CODES AND INTERRUPTS (Continued)

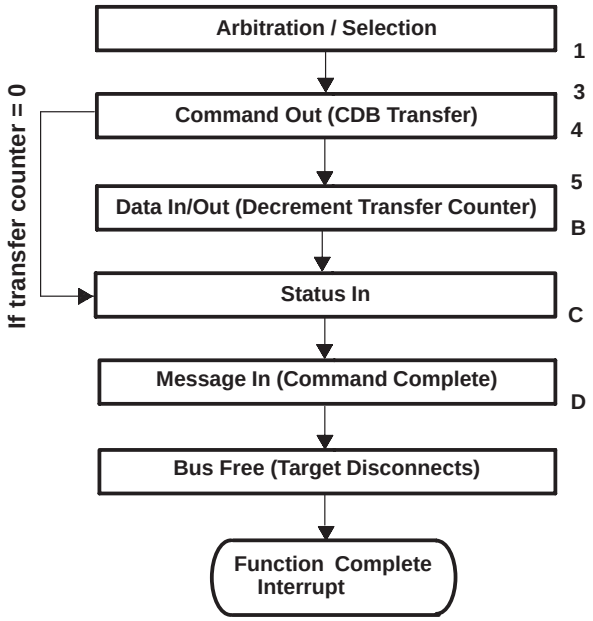
COMMAND STATE CODE (HEX)	LAST SUCCESSFUL PHASE	BIT(S) SET TO 1	REGISTER	CAUSE
D	Command-complete message received	BUS	FISR	Any new bus information phase.
		FC	FISR	Target disconnected.
E	Linked-command-complete message received	BUS	FISR	Any new bus information phase.
		FC	FISR	Immediate.
E	Linked-command-complete-with-flag message received	BUS	FISR	Any new bus information phase.
		FC	FISR	Immediate.

† Following these events, a Clear Transmit FIFO command should be issued to clear the unsent SCSI command bytes from the transmit FIFO.

‡ Following these events, a Clear Transmit FIFO command should be issued to clear the unsent data bytes from the transmit FIFO to prepare the SBC for a transaction with a different logical thread. The transfer counter and/or backup counter register values can be used to update SCSI data pointers so that bytes that were left in the FIFO can be resent later.

4.6.2 Select without $\overline{\text{ATN}}$ and Transfer

This command is similar to Select with $\overline{\text{ATN}}$ and Transfer, with the following exceptions: 1) $\overline{\text{ATN}}$ is not asserted during selection, which implies that the initiator does not support any messages other than the command-complete message; 2) the command-complete message is not stored in the receive FIFO since no linked-command-complete message can be received; 3) there are not multiple entry points into the command. After successful selection, the chip is connected as an initiator and expects a flow of bus phases from the target as defined in Figure 4–2. Deviation from the illustrated flow results in command termination with an interrupt. The command state code indicates the last successful phase of the operation as shown in the command state code/interrupt table.



NOTE: Letter and number tags represent SN75C091A command state codes.

Figure 4–2. Select without $\overline{\text{ATN}}$ and Transfer Command

Any of the following events result in completion or termination of the Select without $\overline{\text{ATN}}$ and Transfer command and generation of the corresponding interrupt. SCSI bus reset and/or control error interrupts can occur during any command phase. (Note: a Chip Reset command or assertion of the SBC $\overline{\text{MR}}$ line terminates any command.)

SELECT-WITHOUT-$\overline{\text{ATN}}$-AND-TRANSFER COMMAND STATE CODES AND INTERRUPTS

COMMAND STATE CODE (HEX)	LAST SUCCESSFUL PHASE	BIT(S) SET TO 1	REGISTER	CAUSE
0	Selection unsuccessful	T-O	EISR	Target did not respond.
		SEL	FISR	Chip selected as a target by an initiator.
		RSL	FISR	Chip reselected as an initiator by a target.
		HALT	EISR	Pause command halted arbitration.
1	Selection successful	BUS	FISR	No command-out phase.
3	Start of CDB transfer	BUS, HALT	FOSR. EOSR	Command phase wrong length.
4	CDB transfer complete	BUS	FISR	No status or data phases. If status, TC > 0. If data, I/O did not match DDIR or TC = 0.
5	Data transfer started	BUS	FISR	No status or data phases. If status, TC > 0. If data, I/O did not match DDIR or TC = 0
		PE	EISR	Parity error on data byte.
		HALT	EISR	Pause command halted data transfer.
B	Data transfer completed (TC = 0)	BUS	FISR	No status phase.
C	Status byte received	BUS	FISR	No message-in phase, or if message in, message was not command-complete.
D	Command-complete message received	BUS	FISR	Any new bus information phase.
		FC	FISR	Target disconnected.

4.6.3 Reselect and Receive Data

This command automatically handles the SCSI phase sequence required for a target device that wishes to reconnect and continue a SCSI data-out phase (receive data for the target). The low-level commands Reselect, Send Message, and Receive Data are chained together as shown in Figure 4–3. Intermediate interrupts are handled internally provided that an unexpected event does not occur. SCSI bus phase lines are driven by the target, so the only unexpected events which can occur after successful reselection are an attention condition from the initiator or a parity error. If a premature command termination and interrupt do occur, the command state code corresponds to the last successfully completed phase. A function-complete interrupt is generated when the command successfully completes.

4.6.3.1 Reselect

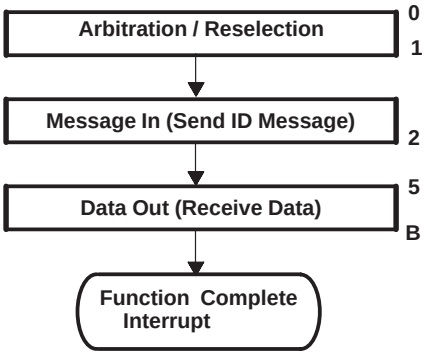
As in the reselect command, the SBC arbitrates for the bus and attempts to reselect the initiator identified in the destination ID register. If successful, the command state code is set to 1 and the SBC continues without interrupting the processor (unless the ATN bit in the functional interrupt status register is set to 1). Assertion of $\overline{\text{ATN}}$ following reselect terminates the command with an attention interrupt.

4.6.3.2 Send ID Message

The SBC drives the SCSI bus to a message-in phase and proceeds to send the ID message by placing the message on the SCSI bus and asserting $\overline{\text{REQ}}$. This message consists of the pattern “10LUNREG”, where LUNREG is the value of bits 5-0 of the target LUN register. No FIFO load of the ID message is required. Parity is generated for the SCSI bus according to the SPGE bit in the parity control register. Transfer of the Send ID message causes the command state code to be set to 2 and the command continues without processor interrupt unless an attention condition is observed. Assertion of $\overline{\text{ATN}}$ following the ID message transfer terminates the command with an attention interrupt.

4.6.3.3 Receive Data

The data phase proceeds just as if the receive data command were issued. The command state code is updated to 5 when the data transfer is started. The command state code is updated to hexadecimal B and a function-complete interrupt is generated upon a successful completion (TC = 0 and no parity errors).



NOTE: Letter and number tags represent SN75C091A command state codes.

Figure 4–3. Reselect and Receive Data Command

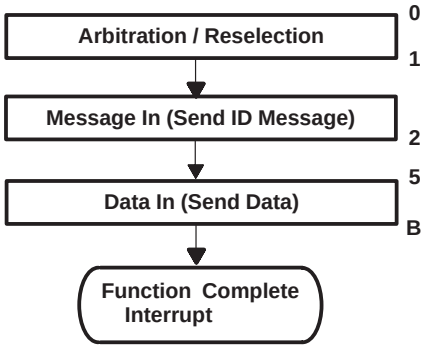
Any of the following events result in completion or termination of the Reselect and Receive Data command and generation of the corresponding interrupt. SCSI bus reset interrupt can occur during any phase. (Note: A Chip Reset command or assertion of the SBC $\overline{MR}$ line terminates any command.)

RESELECT-AND-RECEIVE DATA COMMAND STATE CODES AND INTERRUPTS

COMMAND STATE CODE (HEX)	LAST SUCCESSFUL PHASE	BIT(S) SET TO 1	REGISTER	CAUSE
0	Reselection unsuccessful	T-O	EISR	Initiator did not respond.
		SE:	FISR	Chip selected as a target by an initiator.
		RSL	FISR	Chip reselected as an initiator by a target.
		HALT	EISR	Pause command halted reselect.
1	Reselection successful	ATN	FISR	Initiator asserted $\overline{ATN}$.
2	ID message sent	ATN	FISR	Initiator asserted $\overline{ATN}$.
5	Data transfer started	ATN, HALT	FISR, EISR	Initiator asserted $\overline{ATN}$, HA bit set to 1, and transfer counter > 0.
		PE, HALT	EISR, EISR	Parity error on received data byte, HPE set to 1, and transfer counter > 0.
		PE	EISR	Parity error on received data byte, transfer counter = 0.
B	Data transfer successfully completed	FC	FISR	Transfer counter = 0.

4.6.4 Reselect and Send Data

The Reselect and Send Data command (See Figure 4–4) is identical to the Reselect and Receive Data command except that the data is being sent to the initiator rather than received.



NOTE: Letter and number tags represent SN75C091A command state codes.

Figure 4–4. Reselect and Send Data (Target)

Any of the following events result in completion or termination of the Reselect and Send Data command and generation of the corresponding interrupt. SCSI bus reset interrupt can occur during any phase. (Note: a Chip Reset command or assertion of the SBC $\overline{MR}$ line terminates any command.)

RESELECT-AND-SEND DATA COMMAND STATE CODES AND INTERRUPTS

COMMAND STATE CODE (HEX)	LAST SUCCESSFUL PHASE	BITS(S) SET TO 1	REGISTER	CAUSE
1	Reselection unsuccessful	T-O	EISR	Initiator did not respond.
		SEL	FISR	Chip selected as a target by an initiator.
		RSL	FISR	Chip reselected as an initiator by a target.
		HALT	EISR	Pause command halted reselect.
0	Reselection successful	ATN	FISR	Initiator asserter $\overline{ATN}$
2	ID message sent	ATN	FISR	Initiator asserter $\overline{ATN}$
5	Data transfer started	ATN, HALT	FISR, EISR	Initiator asserted $\overline{ATN}$, HA bit set to 1, and transfer counter > 0.
		PE, HALT	EISR, EISR	Parity error on received data byte, HPE bit set to 1, and transfer counter > 0.
		PE	EISR	Parity error on received data byte, transfer counter = 0.
B	Data transfer completed	FC	FISR	Transfer counter = 0.

4.6.5 Wait for Select with $\overline{ATN}$ and Receive CDB

The Wait for Select with $\overline{ATN}$ and Receive CDB command can be issued if the SBC is in a disconnected state or in target mode. If the SBC is disconnected, the command causes the SBC to wait for an initiator to select it; the ID message and SCSI command phase are then automatically received. This command may also be issued if the SBC is already in the target mode, allowing the processor to take advantage of the message- and command-transfer features. As with the other multiphase commands, any deviation from the flow results in termination and an interrupt. The command state code indicates the last successfully completed phase. The flow for the Wait for Select with $\overline{ATN}$ and Receive command is as shown in Figure 4–5.

4.6.5.1 Selection Phase

The SBC waits to be selected if currently disconnected or immediately continues if it is already in the target mode. Following selection, if the $\overline{ATN}$ line is not asserted, the command terminates with a halted interrupt and the command state code is updated to 6 to inform the microprocessor that the initiator does not support any messages other than the command-complete message. If $\overline{ATN}$ is asserted, the command phase is updated to 1 and the command continues on to the message-out phase without interrupting the processor.

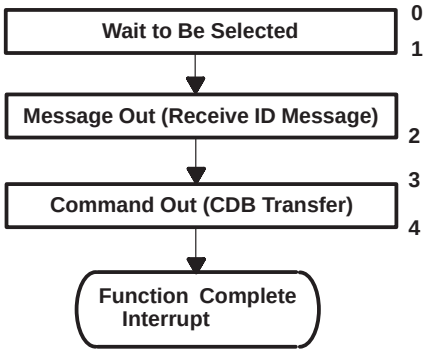
4.6.5.2 Message-Out Phase (Receive ID Message)

If the initiator sends an ID message, it is loaded into the LUN register (not the FIFO) and the command state code is updated to 2. If $\overline{ATN}$ is held asserted following ID message receipt, the command is terminated with an attention interrupt; otherwise, the SBC continues on to the command phase.

If the initiator sends a message other than an ID message, it is loaded into the FIFO and an unexpected message interrupt is generated. Erroneous messages are handled in the same way as in the receive message command (no unexpected message interrupt is generated).

4.6.5.3 4.6.5.3 Command-Out Phase (Receive Command)

The CDB is received in the same way as is described for the Receive Command command. Only automatic mode should be used. The command state code is set to 3 at the start of the command-out phase and is updated to 4 when all bytes have been successfully received.



NOTE: Letter and number tags represent SN75C091A command state codes.

Figure 4–5. Wait for Select with $\overline{ATN}$ and Receive CDB

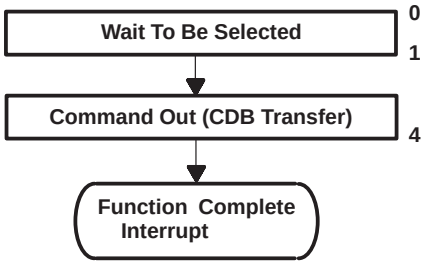
Any of the following events result in completion or termination of the Wait for Select with $\overline{ATN}$ and Receive CDB command and generation of the corresponding interrupt. SCSI bus reset interrupt can occur during any phase. (Note: a Chip Reset command or assertion of the SBC $\overline{MR}$ line terminates any command.)

WAIT FOR SELECT-WITH $\overline{\text{ATN}}$ AND RECEIVE CDB COMMAND STATE CODES AND INTERRUPTS

COMMAND STATE CODE (HEX)	LAST SUCCESSFUL PHASE	BIT(S) SET TO 1	REGISTER	CAUSE
0	Selection unsuccessful	RSL	FISR	Chip reselected as an initiator by a target.
		HALT	EISR	Pause command halted wait-for-selection.
1	Selection successful	PE, HALT	EISR, EISR	Parity error on message-out byte (first two bytes).
		PE	EISR	Parity error on message-out byte (not first 2 bytes).
		UMS	EISR	Message other than ID message received.
6	Selection successful	HALT	EISR	$\overline{\text{ATN}}$ is not asserted.
2	ID message received	ATN	EISR	$\overline{\text{ATN}}$ line asserted after message-out phase.
3	CDB transfer started	PE, HALT	EISR, EISR	Parity error on command-out byte (first byte).
		PE	EISR	Parity error on command-out byte (not first byte).
4	CDB transfer complete	FC	FISR	Correct number of bytes as decoded in CDB successfully transferred.

4.6.6 Wait for Select without $\overline{\text{ATN}}$ and Receive CDB

The Wait for Select without $\overline{\text{ATN}}$ and Receive CDB command can be issued if the SBC is in a disconnected state or in target mode. If the SBC is disconnected, the command causes the SBC to wait for an initiator to select it; the SCSI command phase is then automatically received. If $\overline{\text{ATN}}$ is asserted following selection, the command is terminated with an $\overline{\text{ATN}}$ interrupt. The ATNDS bit in the control register can be used to prevent command termination on $\overline{\text{ATN}}$ if the processor does not care that the initiator supports messages. As with the other multiphase commands, any deviation from the flow results in command termination and an interrupt. The command state code indicates the last successfully completed phase. The flow for the Wait for Select without $\overline{\text{ATN}}$ and Receive CDB command is as shown in Figure 4–6.



NOTE: Letter and number tags represent SN75C091A command state codes.

Figure 4–6. Wait for Select without $\overline{\text{ATN}}$ and Receive CDB

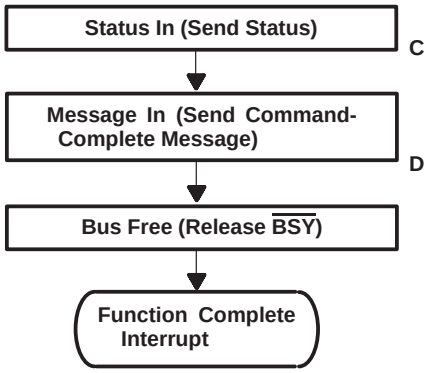
Any of the following events result in completion or termination of the Wait for Select without $\overline{\text{ATN}}$ and Receive CDB command and generation of the corresponding interrupt. SCSI bus reset interrupt can occur during any phase. (Note: a Chip Reset command or assertion of the SBC $\overline{\text{MR}}$ line terminates any command.)

WAIT FOR SELECT-WITHOUT $\overline{\text{ATN}}$ AND RECEIVE CDB COMMAND STATE CODES AND INTERRUPTS

COMMAND STATE CODE (HEX)	LAST SUCCESSFUL PHASE	BIT(S) SET TO 1	REGISTER	CAUSE
0	Selection without ATN unsuccessful	RSL	EISR, EISR	Chip reselected as an initiator by a target.
		HALT	EISR	Pause command halted wait-for-selection.
		ATN	EISR	$\overline{\text{ATN}}$ line was asserted during selection.
1	Selection successful/ CDB transfer started	PE, HALT	EISR, EISR	Parity error on command-out byte (first byte).
		PE	EISR	Parity error on command-out byte (not first byte).
4	CDB transfer complete	FC	FISR	Correct number of bytes as decoded in CDB successfully transferred.

4.6.7 Conclude

This command is used by the target to complete a SCSI transaction with a minimum number of interrupts. The SCSI status must first be loaded into the transmit FIFO. After performing a status-in phase, the chip performs a message-in phase, during which an automatically generated command-complete message is sent. The SBC then releases $\overline{\text{BSY}}$ to disconnect. A function-complete interrupt is generated after disconnection if the FCIE bit in the interrupt enable register is set to 1; otherwise, the target is not notified that the disconnection has occurred. This is useful if the target has no other outstanding threads to reestablish and simply needs to wait until the next selection. If the target has other threads to service, the interrupt enable bit should be set to 1. Figure 4–7 is a flowchart of the Conclude command.



NOTE: Letter and number tags represent SN75C091A command state codes.

Figure 4–7. Conclude Command (Target)

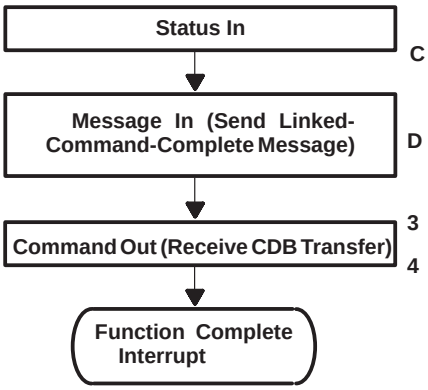
Any of the following events result in completion or termination of the Conclude command and generation of the corresponding interrupt. A SCSI bus reset interrupt can occur during any phase. (Note: a Chip Reset command or assertion of the SBC $\overline{\text{MR}}$ line terminates any command.)

CONCLUDE COMMAND STATE CODES AND INTERRUPTS

COMMAND STATE CODE (HEX)	LAST SUCCESSFUL PHASE	BIT(S) SET TO 1	REGISTER	CAUSE
C	Status sent	ATN	FISR	Initiator asserted $\overline{\text{ATN}}$.
D	Command-complete message sent	ATN	FISR	Initiator asserted $\overline{\text{ATN}}$.
	Disconnect completed	FC	FISR	Command complete.

4.6.8 Link to Next Command

This command is used by the target to support linked SCSI commands from the initiator. The SCSI intermediate status should be loaded into the transmit FIFO and followed by the appropriate linked-command-complete message before the command is issued. The chip then drives the bus through a series of phases, starting with a status-in phase, followed by a message-in phase, and concluded with a command-out phase. The status and message bytes are pulled from the transmit FIFO and the CDB bytes received during the last phase are stored in the receive FIFO. See Figure 4–8 for a flowchart of the Link to Next command.



NOTE: Letter and number tags represent SN75C091A command state codes.

Figure 4–8. Link to Next Command (Target)

Any of the following events result in completion or termination of the Link to Next command and generation of the corresponding interrupt. SCSI bus reset interrupt can occur during any phase. (Note: a Chip Reset command or assertion of the SBC $\overline{MR}$ line terminates any command.)

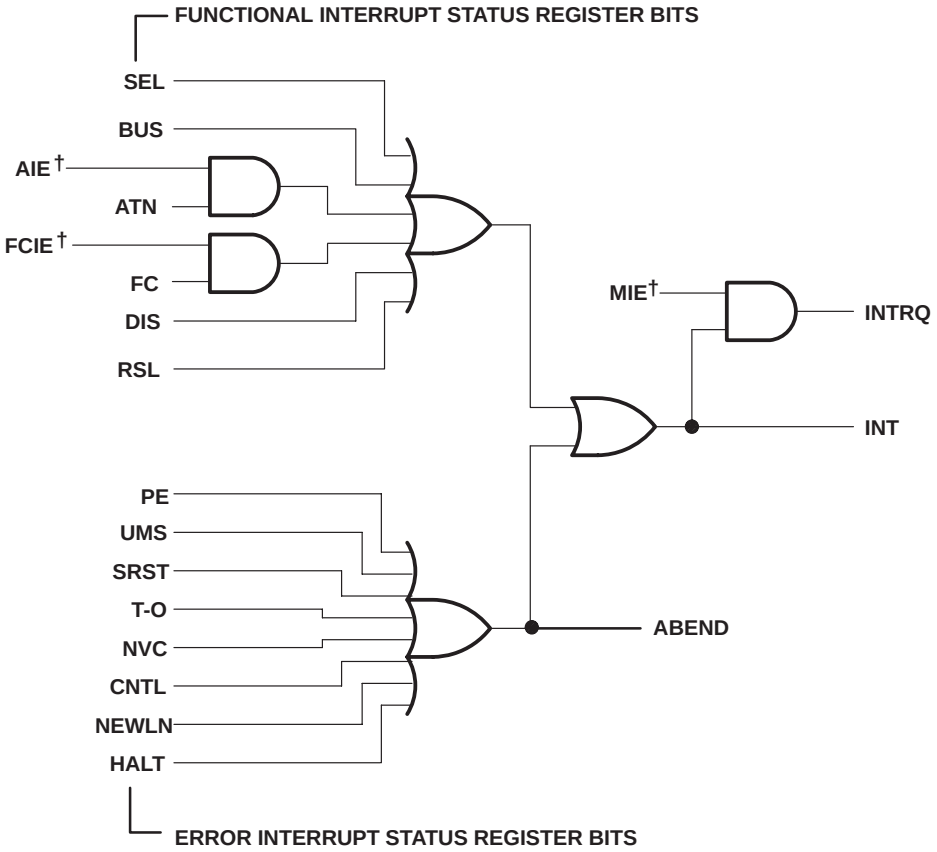
LINK-TO-NEXT-COMMAND COMMAND STATE CODES AND INTERRUPTS				
COMMAND STATE CODE (HEX)	**LAST SUCCESSFUL PHASE**	**BIT(S) SET TO 1**	**REGISTER**	**CAUSE**
C	Status sent	ATN	FISR	Initiator asserted $\overline{ATN}$.
D	Linked-command-complete message sent	ATN	FISR	Initiator asserted $\overline{ATN}$.
3	CDB transfer started	PE, HALT	EISR	Parity error on command-out byte (first byte).
		PE	EISR	Parity error on command-out byte (not first byte).
4	CDB transfer complete	FC	FISR	Correct number of bytes as decoded in CDB successfully transferred.

5 Interrupt Handler

SBC interrupts alert the microprocessor to events that require microprocessor intervention. The microprocessor can detect interrupts by polling the INT bit in the SBC transfer status register or by monitoring the SBC INTRQ line. Interrupts are enabled onto the INTRQ line by setting the SBC interrupt enable register MIE (Master Interrupt Enable) bit to 1.

The functional interrupt and error interrupt status registers provide the microprocessor with information about the cause of an interrupt. The error interrupt status register bits indicate error conditions; all the functional interrupt status register bits except the ABEND bit indicate conditions that occur as part of a normal SCSI transaction. The ABEND bit is set to 1 whenever any bit in the error interrupt status register is set to 1; therefore, the error interrupt status register need only be read when the ABEND bit is set to 1.

Any bit set to 1 in either interrupt status register can cause the transfer status register INT bit to be set to 1 or the INTRQ line to be asserted. The FCIE (Function Complete Interrupt Enable), AIE ($\overline{\text{ATN}}$ Interrupt Enable), and MIE bits in the interrupt enable register are used to control interrupt generation as shown in Figure 5–1.



† These inputs are from the Interrupt Enable register.

Figure 5–1. Interrupt Control

Both interrupt status registers are latched at the beginning of a read operation to insure that the data is stable. With the exception of the ABEND bit in the functional interrupt status register, the read operation clears all interrupts that were active when the read commenced. The ABEND bit is cleared by reading the error interrupt status register.

Any interrupt that occurs during a read from the interrupt register is loaded into the interrupt register immediately following completion of the read operation. In this case, INTRQ remains active upon read completion, signifying that there is another interrupt pending. This eliminates the possibility of missing an interrupt that occurred during a read operation. Devices that interface with INTRQ must be level sensitive.

6 DMA Interface Operation

6.1 General

The DMA interface provides the control necessary to transfer data between memory and the SBC using a DMA controller. DREQ is an output that requests that the DMA controller read data from, or write data to, the SBC or external byte stack register(s). The DMA controller responds to DREQ by asserting $\overline{\text{DACK}}$ and performing a read or write. Non-byte-stack mode is provided for 8-bit memory bus interfacing and byte stack mode is provided to accommodate 16-, 24-, or 32-bit memory buses.

6.2 Non-Byte-Stack Mode

In non-byte-stack mode, a word is defined as one byte (8 bits data plus one parity bit). Setting the word length bits in the byte stack control register to 0 selects non-byte-stack mode. The DMA controller responds to DREQ by asserting $\overline{\text{DACK}}$ and reading or writing directly to the SBC using the $\overline{\text{MRD}}$ or $\overline{\text{MWR}}$ inputs, respectively.

6.2.1 Nondemand Transfers (Non-Byte-Stack Mode)

If the DMD bit in the byte stack control register is set to 0, then transfers require a full DREQ→$\overline{\text{DACK}}$ handshake for each byte transferred. Figure 6–1 illustrates the handshake protocol; timing diagrams in Section 7 detail the timing requirements. Nondemand transfers return control of the memory bus to the processor after each word is transferred.

6.2.2 Demand Transfers (Non-Byte-Stack Mode)

If the DMD bit in the byte stack control register is set to 1, then multiple bytes can be transferred with one DREQ→$\overline{\text{DACK}}$ handshake. This is the fastest transfer mode because many DMA memory bus request cycles are eliminated. Reads or writes can be performed continuously until there is not enough data (read) or enough room in the FIFO (write) to continue. At this point, the $\overline{\text{WAIT}}$ line is asserted to hold off the rising edge of $\overline{\text{MRD}}$ or $\overline{\text{MWR}}$. DREQ is released to indicate transition to nondemand mode and then $\overline{\text{WAIT}}$ is released to allow the current read or write to complete. $\overline{\text{DACK}}$ can then be released. Transfers resume (DREQ is asserted) when there is enough data (read) or room (write) in the FIFO to continue in demand mode. The final bytes of the command are always transferred using the nondemand flow. $\overline{\text{WAIT}}$ is used to perform nondemand transfers while the chip is in demand mode in order to prevent the demand device from writing or reading unneeded data. By automatically switching to nondemand transfers when the available data or FIFO space becomes limited, the processor can use the memory bus while the DMA is waiting for the SCSI bus to catch up.

6.2.3 Command Completion (Non-Byte-Stack Mode)

A command using the DMA interface completes when all of the data for the current command has been transferred through the SBC. However, there are several situations in which a transfer is halted prematurely, e.g., Halt on $\overline{\text{ATN}}$ or Halt on Parity Error, an unexpected SCSI bus phase change, or a halt command. In these instances, the command using the DMA interface completes differently depending on whether reads or writes are being performed:

DMA Read: The DMA interface always empties the FIFO prior to command completion.

DMA Write: The DMA interface finishes only the current write transfer (and/or one nondemand write when in demand mode) prior to command completion. Data is left in the transmit FIFO if a command is aborted prematurely. Note that, for initiators operating with multiple threads, this case may arise when a target disconnects and the initiator is reselected by a different target or LUN. In this case, the initiator host should read the transfer counter and/or backup counter registers to update SCSI data pointers so that bytes which were left in the FIFO can be resent later. A Transmit FIFO Clear command should then be issued prior to continuing with the new thread.

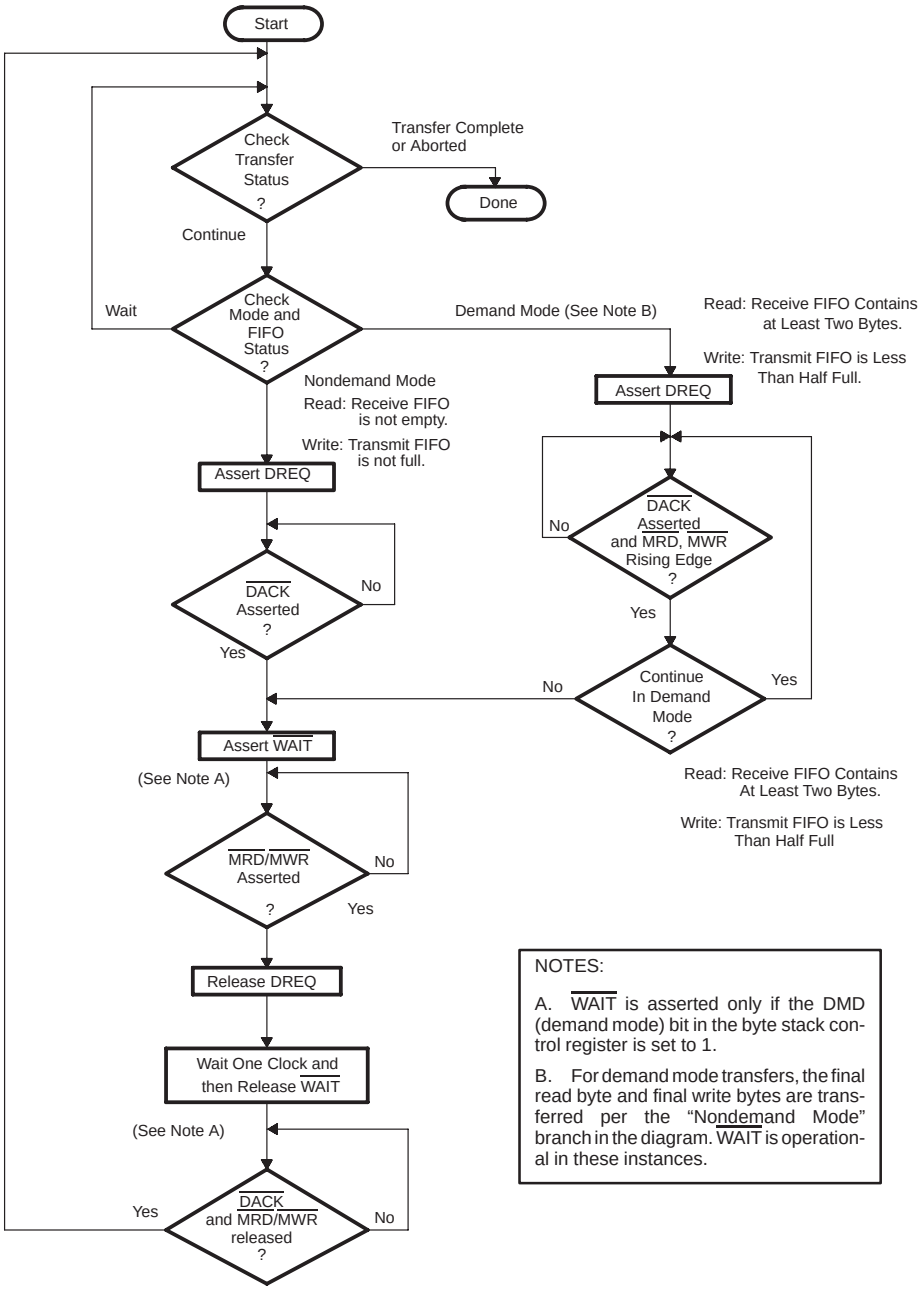


Figure 6–1. Flowchart for DMA Read/Write (Non-Byte-Stack Mode)

6.3 Byte Stack Mode

The byte stack logic included on the SBC device allows an efficient interface to two-, three-, or four-byte wide memory through an external DMA controller. Since the SCSI bus is defined as an eight-bit wide data path, any transfers between wider buses require data compression on the source side and data expansion on the destination side as shown in Figure 6–2. Operation 1 loads a full word into the byte stack buffer, operations 2 through 5 send one byte each across the bus, and operation 6 loads the full word into the destination

memory. The word is partitioned before, and rebuilt after, the transfer across the SCSI bus. The SBC provides the control signals to partition and rebuild the bytes. Figure 6–2 shows a maximum implementation in the sense that both sides of the bus are four bytes wide. It is equally likely that only one side requires byte stacking and that the other has an eight-bit bus, as shown in Figure 6–3. In this case, only compression or expansion, but not both, are required per transfer.

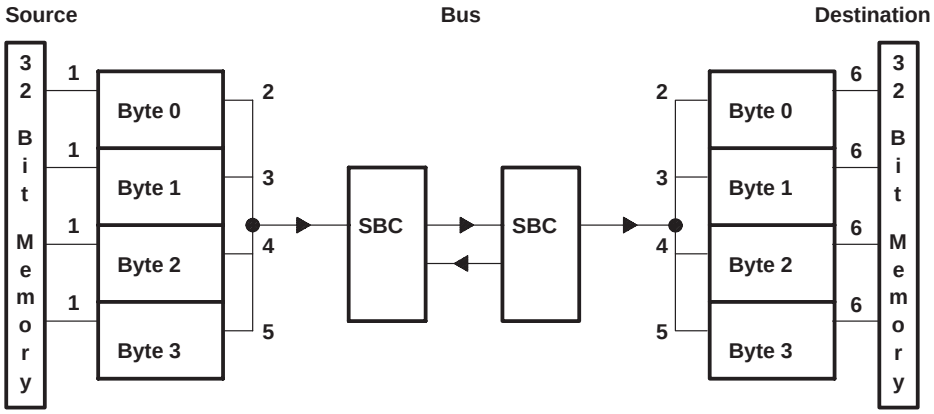


Figure 6–2. Byte Stack Mode Data Compression and Expansion

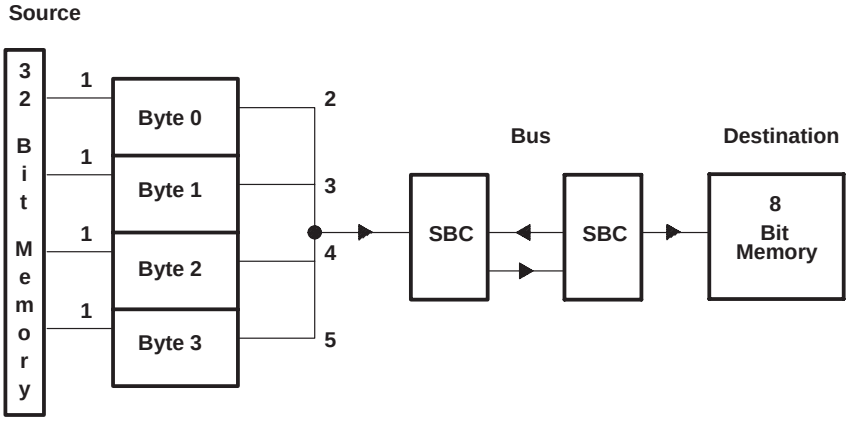


Figure 6–3. Byte Stack Mode Data Compression

6.3.1 DMA Controller Interface

The SBC interfaces to a DMA controller through a DREQ—>$\overline{\text{DACK}}$ handshake. DREQ is asserted to initiate a transfer. The DMA controller then responds with $\overline{\text{DACK}}$ and reads or writes the byte stack register, causing assertion of $\overline{\text{BSSTB}}$ to signal the SBC that the byte stack register access is complete. A byte stack register is a two-port register used to implement the partition and rebuild functions mentioned in Section 6.3. $\overline{\text{WAIT}}$ is asserted to prevent the DMA controller from prematurely accessing the byte stack register or to transition from demand- to nondemand mode as described below.

6.3.1.1 Nondemand Transfers (Byte Stack Mode)

If the DMD bit in the byte stack control register is set to 0, then transfers require a full DREQ→$\overline{\text{DACK}}$ handshake for each word transferred. Figure 6–2 illustrates the handshake protocol; timing diagrams in Section 7 detail the timing requirements. Nondemand transfers allow the processor to gain control of the memory bus after each word is transferred. $\overline{\text{WAIT}}$ is asserted in nondemand mode to delay byte stack accesses during a byte stack load or unload cycle.

6.3.1.2 Demand Transfers (Byte Stack Mode)

If the DMD bit in the byte stack control register is set to 1, then multiple words can be transferred within one DREQ→DACK handshake. This is the fastest transfer mode because many DMA memory bus request cycles are eliminated. Reads or writes may be performed continuously until there is not enough data (read) or enough room in the FIFO (write) to accommodate another full byte stack of data. The WAIT line is asserted to disable byte stack register accesses when data is being loaded or unloaded from the byte stack. WAIT is also asserted when the SBC switches to nondemand mode (i.e., when there is not enough data left in the FIFO (read) or enough space left in the FIFO (write) to continue in demand mode). DREQ is released to indicate transition to nondemand mode and then WAIT is released to allow the current read or write to complete. DACK may then be released. Transfers resume (DREQ is asserted) when there is enough data (read) or room (write) in the FIFO to continue in demand mode. The final bytes of the command are always transferred using the nondemand flow. Automatically switching to nondemand transfers when the available data or FIFO space becomes limited allows the processor to access the memory bus while the DMA controller waits for the SCSI bus to catch up.

6.3.1.3 Command Completion (Byte Stack Mode)

A command using the DMA interface completes execution when all data for the current command has been transferred through the SBC (in contrast, a command which uses the processor interface completes execution when all data received from the SCSI bus is in the FIFO). There are several situations, however, in which a transfer is halted prematurely, e.g., Halt on ATN or Halt on Parity Error, an unexpected SCSI bus phase change, or a halt command. In these instances, the command using the DMA interface completes differently depending on whether reads or writes are being performed:

> DMA Read: The DMA interface always empties the FIFO to the point where there is not enough data remaining in the FIFO to completely fill another byte stack. Data left in the FIFO must be removed through the processor interface or by issuing another data transfer command using the DMA interface. Note that a successfully completed transfer always empties the FIFO regardless of whether or not there are enough bytes to complete the last word.The processor must keep track of transfers which use uneven byte stack boundaries (see use of byte offset feature).
>
> DMA Write: The DMA interface finishes only the current write transfer (and/or one nondemand write when in demand mode) prior to command completion. Data is left in the transmit FIFO if a command is aborted prematurely. Data is never left in the byte stack register. Note that, for initiators operating with multiple threads, this case may arise when a target disconnects and the initiator is reselected by a different target or LUN. In this case, the initiator host should read the transfer counter and/or backup counter registers to update SCSI data pointers so that bytes which were left in the FIFO can be resent later. A Transmit FIFO Clear command should then be issued prior to continuing with the new thread.

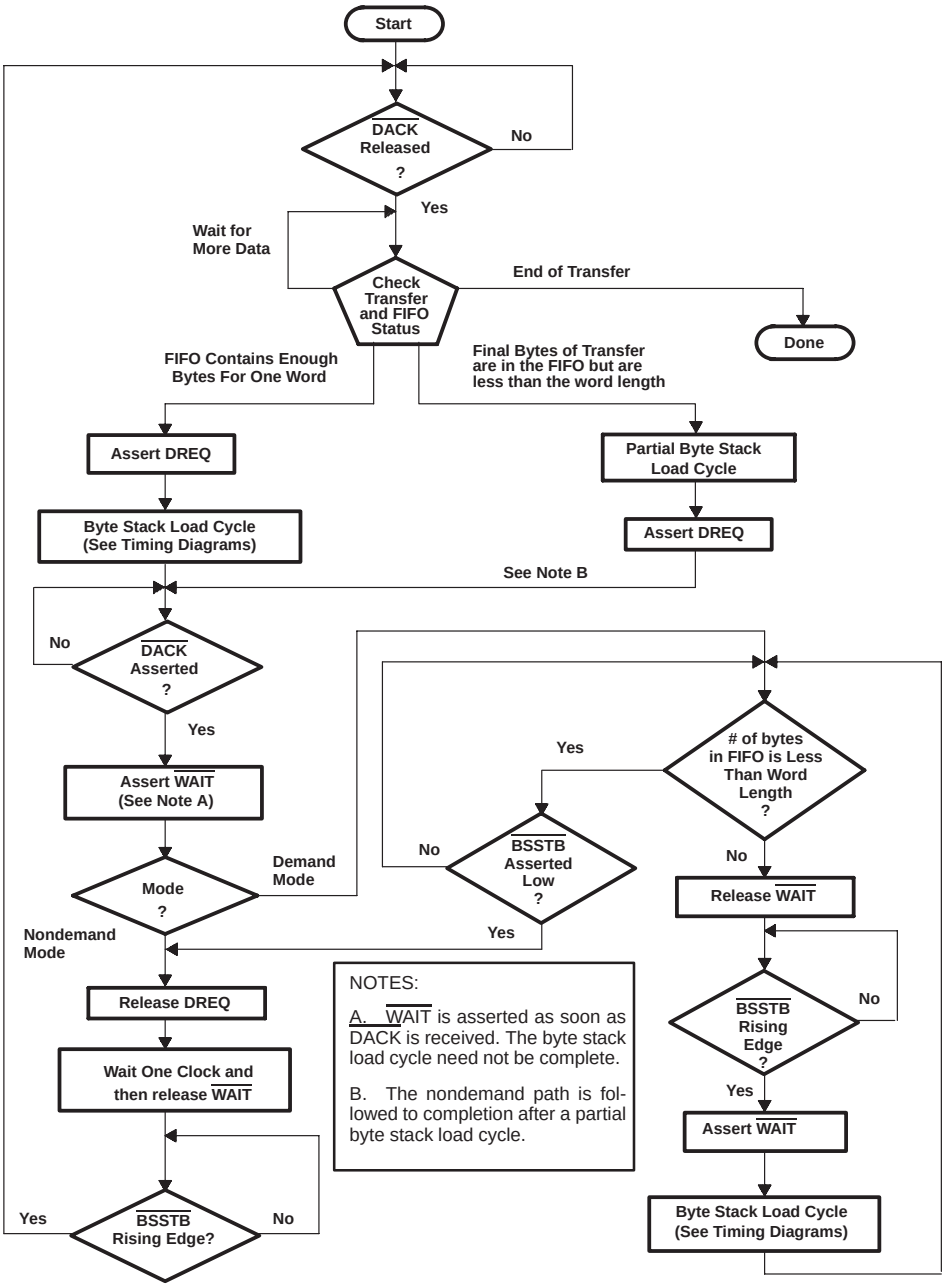


Figure 6–4. Flowchart for DMA Read (Byte Stack Mode)

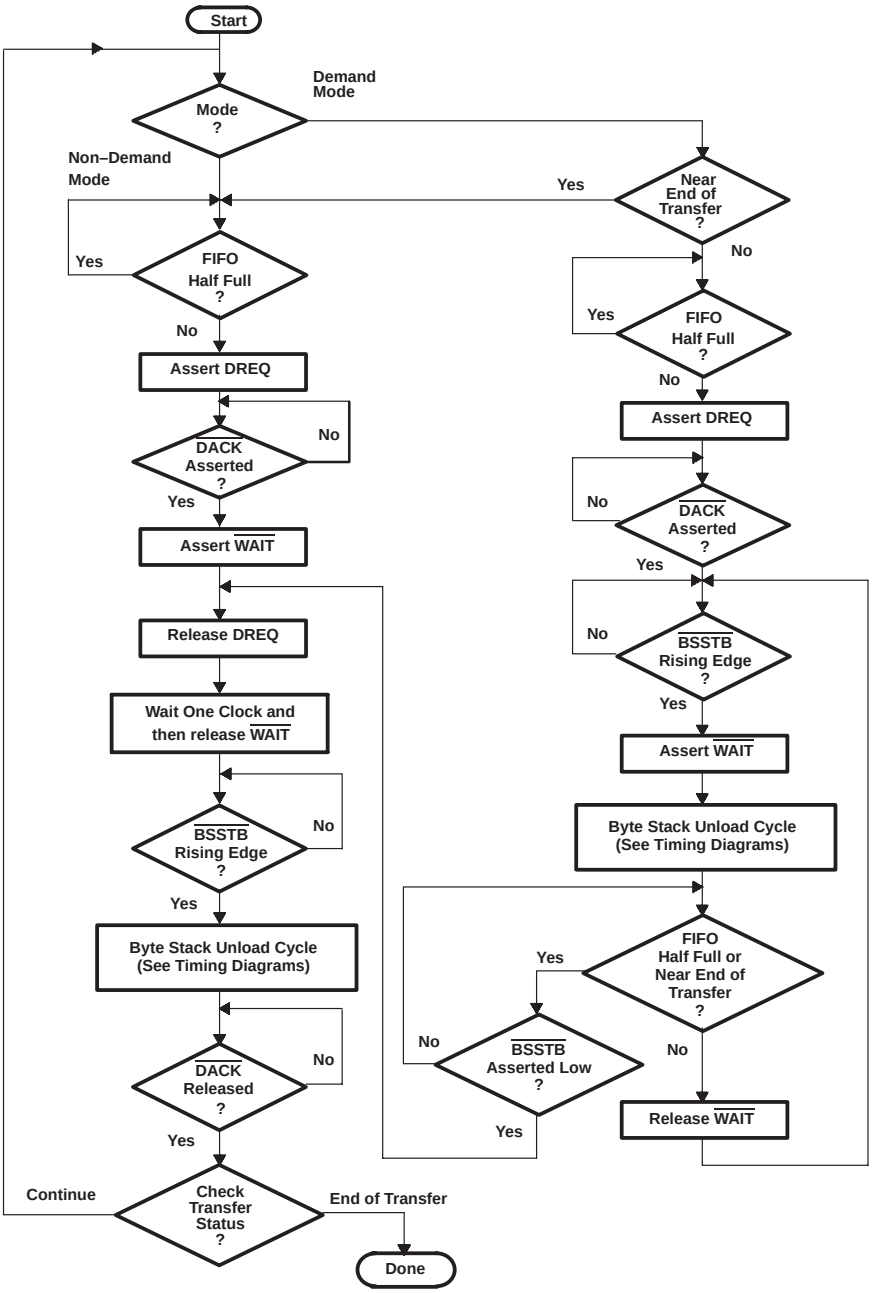


Figure 6–5. Flowchart for DMA Write (Byte Stack Mode)

6.3.2 Byte Stack Register Interface

Several control lines are provided on the SBC to simplify the interface between the SBC, the DMA controller, and external registers used as temporary storage for stacking (unstacking) bytes. Some signals are multiply-defined depending on whether a 16-, 24-, or 32-bit byte stack is selected by the word length bits in the byte stack control register. For DMA reads, the byte stack registers are loaded by the SBC using a byte stack load cycle. The DMA controller may read the data when the load is complete. For DMA writes, the DMA controller writes a word to the byte stack register. The SBC unloads the bytes using a byte stack unload cycle.

6.3.2.1 $\overline{\text{BSSTB}}$ Generation

The $\overline{\text{BSSTB}}$ line is used to signal the SBC that the external byte stack registers have been read or written by the DMA controller and the SBC can now take further action. For controllers which pulse the $\overline{\text{DACK}}$ line for each access (nondemand mode), $\overline{\text{BSSTB}}$ can be tied directly to $\overline{\text{DACK}}$. For demand-mode systems, the DMA controller lines that read and write the byte stack register must be used to produce an active-low $\overline{\text{BSSTB}}$ pulse when either a byte stack register read or write occurs.

6.3.2.2 16-Bit Byte Stack Implementation

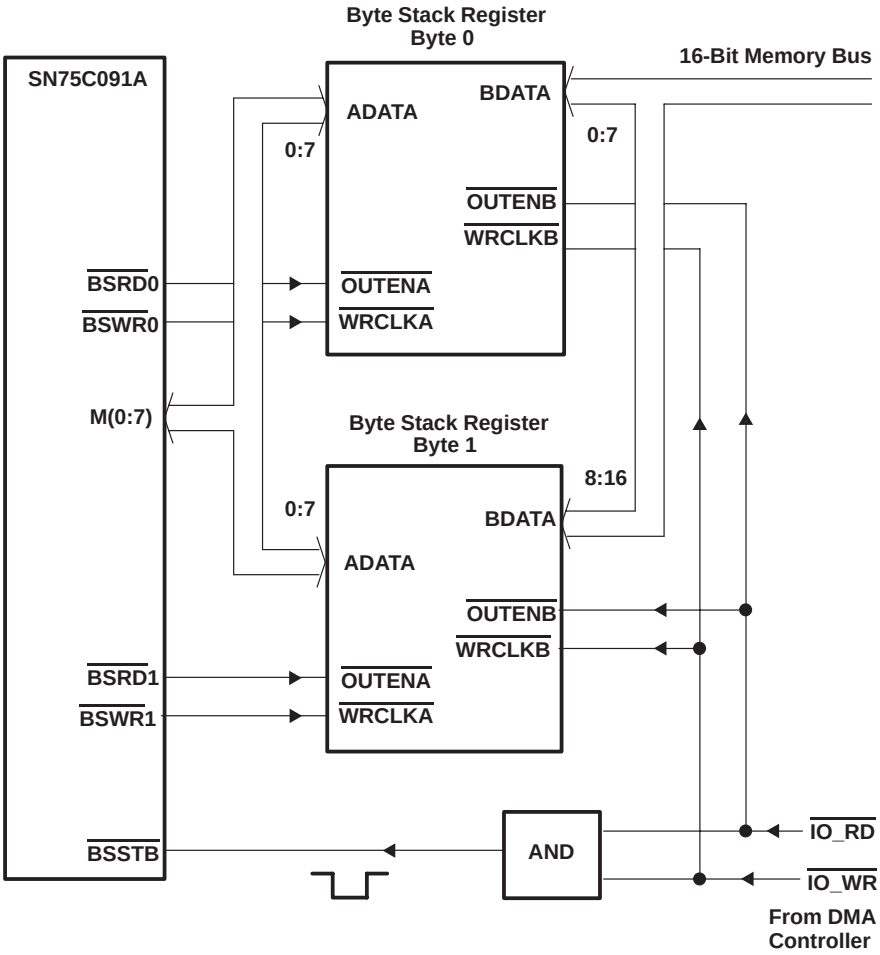
16-bit (2 byte) byte stack operation is selected by writing “01” to the word length bits in the byte stack control register. Note that, for 16-bit operation, no additional logic (such as a decoder) is needed for implementing a byte stack register interface.

6.3.2.2.1 $\overline{\text{BSWR1}}$ and $\overline{\text{BSWR0}}$ (SBC Byte Stack Register Write)

$\overline{\text{BSWR1}}$ and $\overline{\text{BSWR0}}$ are active-low signals that write data into byte stack register bytes 1 and 0, respectively. These lines go low for one clock cycle and data is latched on the trailing (low-to-high) edge.

6.3.2.2.2 $\overline{\text{BSRD1}}$ and $\overline{\text{BSRD0}}$ (SBC Byte Stack Register Read)

$\overline{\text{BSRD1}}$ and $\overline{\text{BSRD0}}$ are output signals that directly enable the byte stack register outputs for bytes 1 and 0, respectively, onto memory bus lines M(0:7) for loading into the SBC. Each of these signals goes low for one clock cycle, during which the data is loaded into the SBC chip.



Note: The SBC M(0:7) bus provides an optional parity bit. Byte stack registers should be designed for nine-bit operation if this parity bit is used.

Figure 6–6. 16-Bit Byte Stack Register Implementation

6.3.2.3 24- or 32-Bit Byte Stack Implementation

24- or 32-bit byte stack operation is selected by writing 10 or 11, respectively, to the word length bits (bits 3-2) in the byte stack control register. A 24-bit implementation is identical to a 32-bit implementation except that one less byte of the byte stack register is used.

6.3.2.3.1 $\overline{\text{BSRD}}$ (Byte Stack ReaD)

$\overline{\text{BSRD}}$ is an output control line which is used along with the BSEN1 and BSEN0 lines to enable the outputs of one of the byte stack registers onto M(0:7).

6.3.2.3.2 $\overline{\text{BSWR}}$ (Byte Stack WRite)

$\overline{\text{BSWR}}$ is an output control line which is used along with the BSEN1 and BSEN0 lines to clock M(0:7) data into one of the byte stack registers on its rising edge.

6.3.2.3.3 BSEN1 and BSEN0 (Byte Stack ENable)

BSEN1 and BSEN0 are output control lines which are encoded to select one of three or four external registers when 24- or 32-bit byte stack mode is being used. BSEN1 and BSEN0 are used in conjunction with $\overline{\text{BSRD}}$ and $\overline{\text{BSWR}}$ to transfer bytes between the byte stack register and the SBC.

BSEN1	BSEN0	BYTE STACK BYTE ACCESSED
0	0	byte 0
0	1	byte 1
1	0	byte 2
1	1	byte 3 (32-bit mode only)

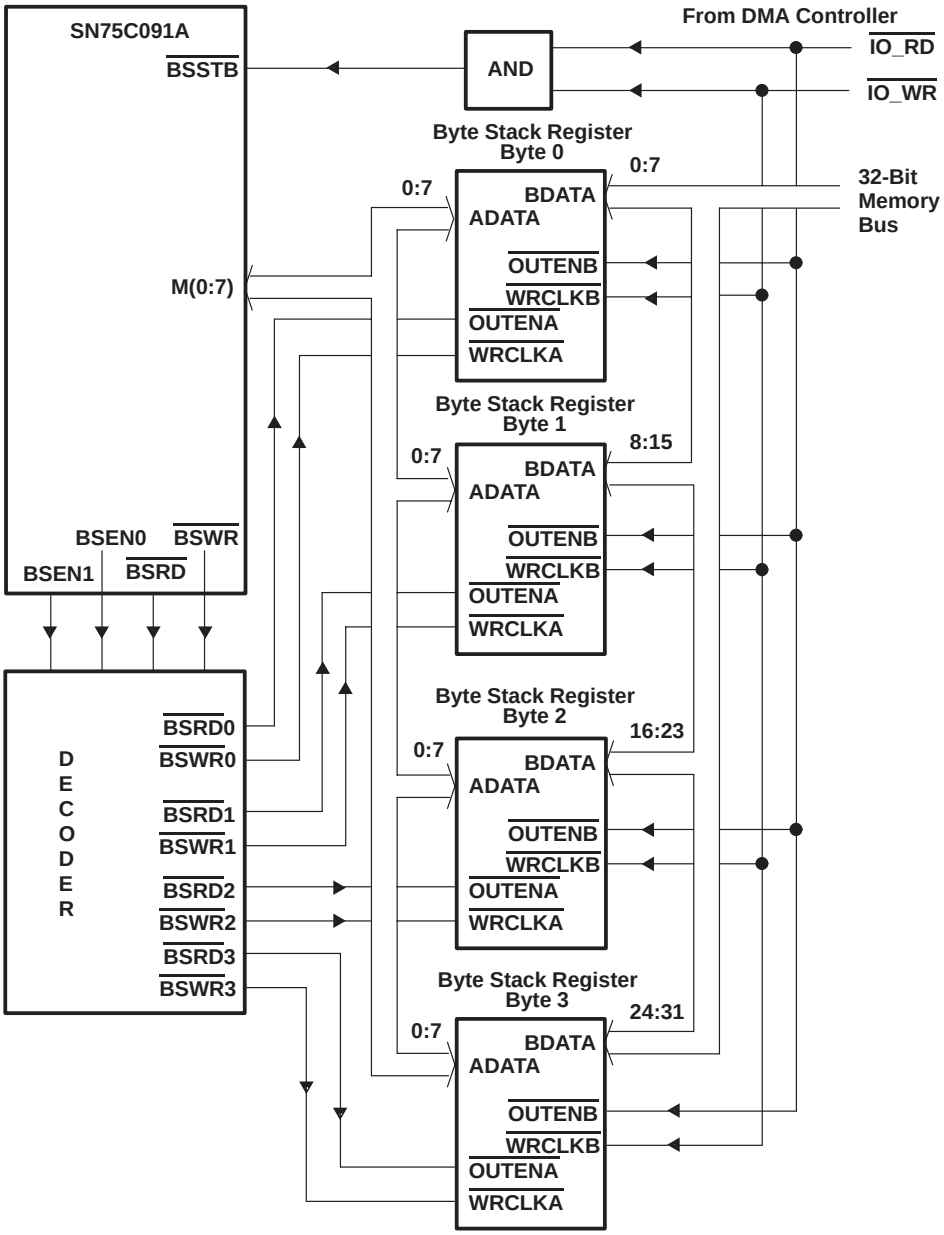


Figure 6–7. 32-Bit Byte Stack Register Implementation

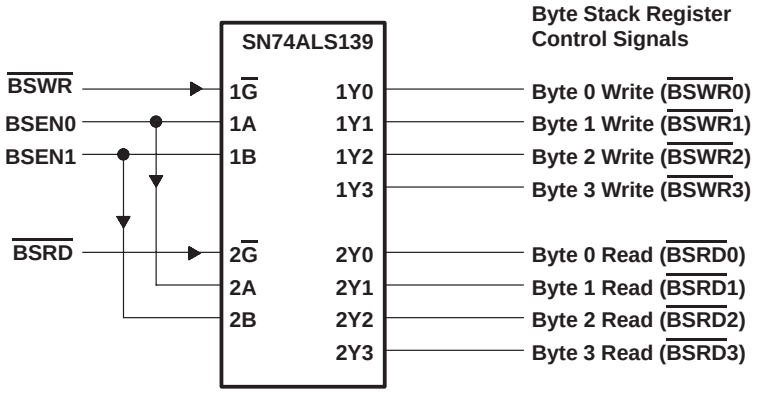


Figure 6–8. Possible Decoding Scheme Using SN74ALS139

6.3.3 Byte offset

The byte stacking feature allows the user to program a byte offset into the first word to be transferred by using the BOF bits (bits 1 and 0 of the byte stack control register). These bits are set to all zeros after the first word is transferred. An offset of one is allowed for 16-bit words, an offset of one or two for 24-bit words, and an offset of one, two, or three for 32-bit words. This allows the transfer from memory of data blocks which are not aligned on word boundaries. The end of the of the block to be transferred may also lie on a nonword boundary, depending on the number of bytes set up to transfer in the transfer counter. It is the system designer's responsibility to assure that no unwanted overwriting occurs when this option is used. This may necessitate preloading of the byte stack registers with the current memory contents by the microprocessor or DMA controller before the first write from the SCSI bus to memory occurs. This action maintains the integrity of the bytes which are not updated due to the offset. The following block examples illustrate the use of byte offsets:

HEX ADDRESS	BYTE 0 MSB	BYTE 1	BYTE 2	BYTE 3 LSB	
00FC		XXXX	XXXX	XXXX	
0100	XXXX	XXXX	XXXX	XXXX	BYTE OFFSET = 1 TRANSFER COUNTER = 8
0104	XXXX				
0A00			XXXX	XXXX	
0A04	XXXX	XXXX	XXXX	XXXX	BYTE OFFSET = 2 TRANSFER COUNTER = 10
0A08	XXXX	XXXX	XXXX	XXXX	
FFAC	XXXX	XXXX	XXXX	XXXX	
FFB0	XXXX	XXXX	XXXX	XXXX	BYTE OFFSET = 0 TRANSFER COUNTER = 10
FFB4	XXXX	XXXX			

Figure 6–9. Examples of Byte Offset

7 Electrical Characteristics

7.1 Absolute Maximum Ratings Over Free-Air Temperature Range (Unless Otherwise Noted)

Supply voltage range, V_{CC} (see Note 1) . . . . . –0.5 V to 7 V
Input voltage range, V_I, at any input . . . . . –0.5 V to 7 V
Output voltage range, V_O . . . . . –0.5 V to 7 V
Storage temperature range . . . . . –65°C to 150°C
Case temperature for 10 seconds . . . . . 260°C

NOTE 1: All voltage values are with respect to GND.

7.2 Recommended Operating Conditions

	MIN	NOM	MAX	UNIT
Supply voltage, V_{CC}	4.75	5	5.25	V
High-level input voltage, V_{IH}	2		V_{CC}	V
Low-level input voltage, V_{IL}†	–0.5		0.8	V
Clock frequency, f_{clock}		20		MHz
Operating free-air temperature, T_A	0		70	°C

† The algebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data manual for logic voltage levels only.

7.3 Electrical Caracteristics Over Recommended Ranges of Supply Voltage and Operating Free-Air Temperature (Unless Otherwise Noted)

PARAMETER			TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V_{OH}	High-level output voltage		I_{OH} = –4 mA (see Note 2)	3.7			V
			I_{OH} = –2 mA (see Note 3)				
V_{OL}	Low-level output voltage		I_{OL} = 48 mA (see Note 4)			0.5	V
			I_{OL} = 4 mA (see Note 2)				
			I_{OL} = 2 mA (see Note 3)				
I_I	Input current		V_{CC} = 5.25 V, V_I = 0 to 5.25 V			±10	µA
I_{OZ}	High-impedance output current		V_{CC} = 5.25 V, V_I = 0 to 5.25 V			±10	µA
I_{CC}	Supply current		No load on outputs, f = 20 MHz			30	mA
C_i	Input capacitance	Input pins	V_{CC} = 5 V, T_A = 25°C		5		pF
		Bidirectional pins			13		pF
C_o	Output capacitance	Output pins	V_{CC} = 5 V, T_A = 25°C		8		pF

† All typical vaules are at V_{CC} = 5 V and T_A = 25°C.

NOTES: 2. Applies to MP, M(0:7) and DP, D(0:7) only.
3. Applies to all other outputs or bidirectional signals.
4. Applies to SCSI interface signals only.

7.4 Timing Characteristics

In the following timing diagrams, all measurements are made at 1.5 V. A 50-pF load applies for SCSI interface signals; a 15-pF load applies for all other signals.

7.5 timing requirements over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{su1}	Setup time, A(0:4) valid before $\overline{RD}$ low	See Figure 7–1	0		ns
t_{h1}	Hold time, A(0:4) valid after $\overline{RD}$ high		0		ns
t_{su2}	Setup time, $\overline{CS}$ low before $\overline{RD}$ low		0		ns
t_{h2}	Hold time, $\overline{CS}$ low after $\overline{RD}$ high		0		ns
$t_{w(L)}$	Pulse duration, $\overline{RD}$ low		110		ns
$t_{w(H)}$	Pulse duration, $\overline{RD}$ high		110		ns

7.6 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{en}	Enable time, $\overline{RD}$ low to D(0:7) active	C_L = 15 pF, See Figure 7–1	5		ns
t_a	Access time, $\overline{RD}$ low to D(0:7) valid			95	ns
t_{DIS}	Disable time, $\overline{RD}$ high to D(0:7) in high-impedance state		5	25	ns

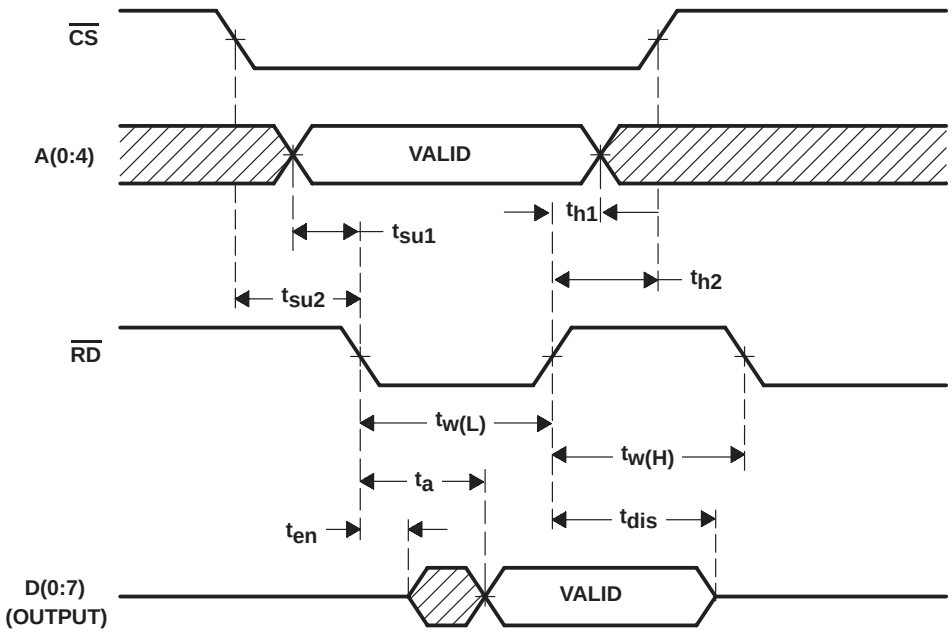


Figure 7–1. Processor Read Cycle (Nonmultiplexed Address/Data Buses)

7.7 timing requirements over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{su1}	Setup time, A(0:4) valid before $\overline{WR}$ low	See Figure 7–2	0		ns
t_{h1}	Hold time, A(0:4) valid after $\overline{WR}$ high		0		ns
t_{su2}	Setup time, $\overline{CS}$ low before $\overline{WR}$ low		0		ns
t_{h2}	Hold time, $\overline{CS}$ low after $\overline{WR}$ high		0		ns
t_{su3}	Setup time, D(0:7) valid before $\overline{WR}$ high		50		ns
t_{h3}	Hold time, D(0:7) valid after $\overline{WR}$ high		0		ns
$t_{w(L)}$	Pulse duration, $\overline{WR}$ low		70		ns
$t_{w(H)}$	Pulse duration, $\overline{WR}$ high		110		ns

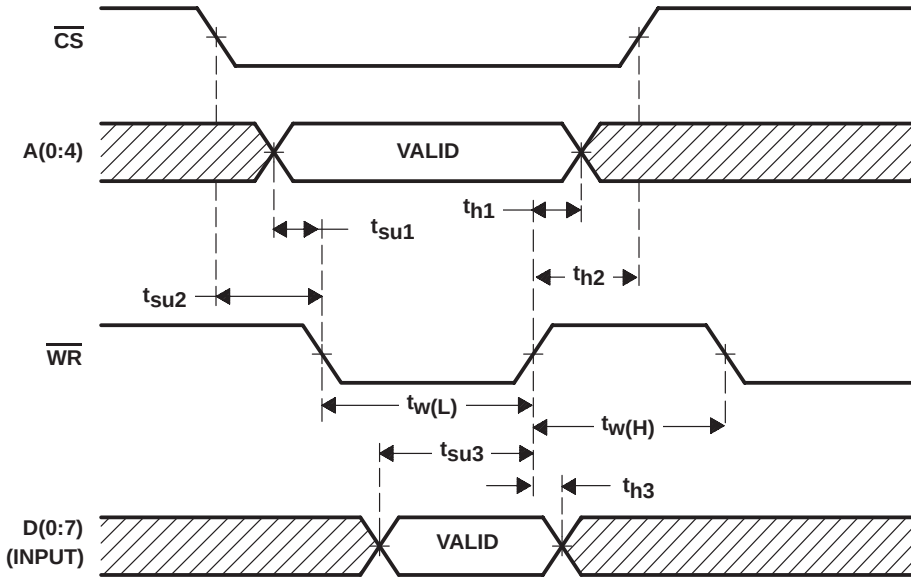


Figure 7–2. Processor Write Cycle (Nonmultiplexed Address/Data Buses)

7.8 timing requirements over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
$t_{w1(H)}$	Pulse duration, ALE high	See Figure 7–3	20		ns
t_{su1}	Setup time, A(0:4) valid before ALE low		20		ns
t_{h1}	Hold time, A(0:4) valid after ALE low		0		ns
t_{su2}	Setup time, $\overline{CS}$ low before ALE low		20		ns
t_{h2}	Hold time, $\overline{CS}$ low after ALE low		0		ns
t_{su3}	Setup time, $\overline{CS}$ low before $\overline{RD}$ low		0		ns
t_{su4}	Setup time, A(0:4) valid before $\overline{RD}$ low		0		ns
$t_{w2(L)}$	Pulse duration, $\overline{RD}$ low		110		ns
$t_{w2(H)}$	Pulse duration, $\overline{RD}$ high		110		ns

7.9 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{en}	Enable time, $\overline{RD}$ low to D(0:7) active	C_L = 15 pF, See Figure 7–3	5		ns
t_a	Access time, $\overline{RD}$ low to D(0:7) valid			95	ns
t_{dis}	Disable time, $\overline{RD}$ high to D(0:7) in high-impedance state		5	25	NS

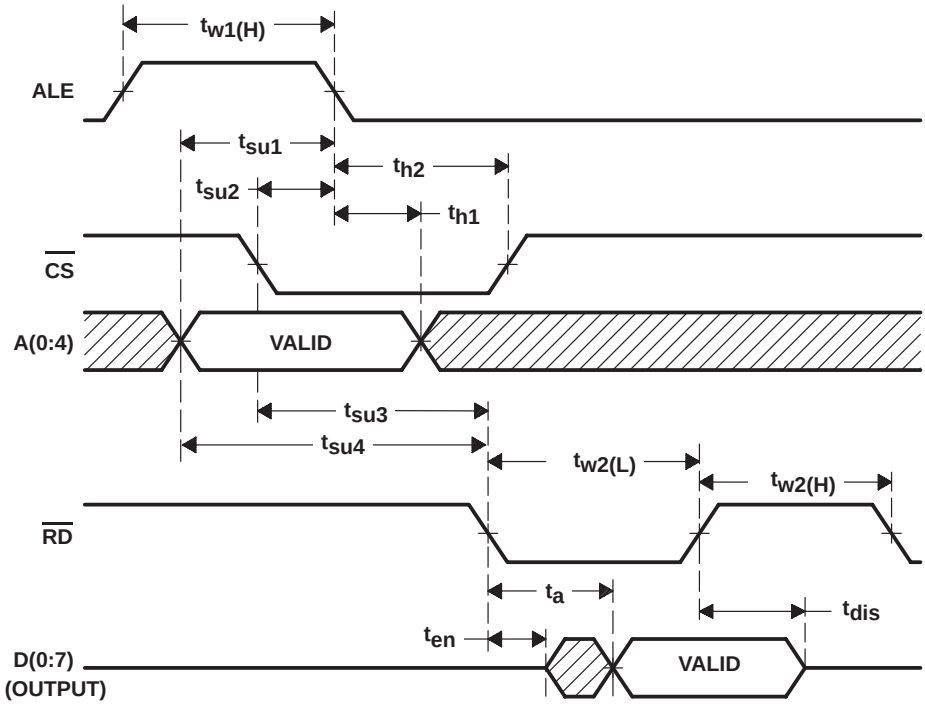


Figure 7–3. Processor Read Cycle (Multiplexed Address/Data Buses)

7.10 timing requirements over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
$t_{w1(H)}$	Pulse duration, ALE high	See Figure 7–4	20		ns
t_{su1}	Setup time, A(0:4) valid before ALE low		20		ns
t_{h1}	Hold time, A(0:4) valid after ALE low		0		ns
t_{su2}	Setup time, $\overline{CS}$ low before ALE low		20		ns
t_{h2}	Hold time, $\overline{CS}$ low after ALE low		0		ns
t_{su3}	Setup time, data valid before $\overline{WR}$ high		50		ns
t_{h3}	Hold time, data valid after $\overline{WR}$ high		0		ns
t_{su4}	Setup time, $\overline{CS}$ low before $\overline{WR}$ low		0		ns
t_{su5}	Setup time, A(0:4) valid before $\overline{WR}$ low		0		ns
$t_{w2(L)}$	Pulse duration, $\overline{WR}$ low		70		ns
$t_{w2(H)}$	Pulse duration, $\overline{WR}$ high		110		ns

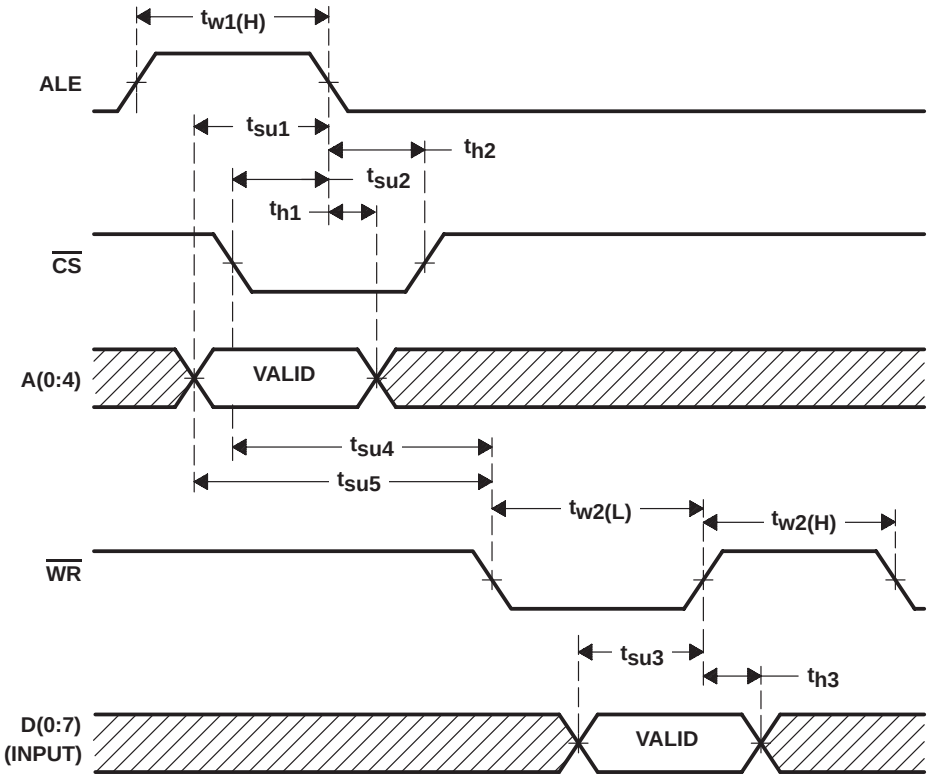


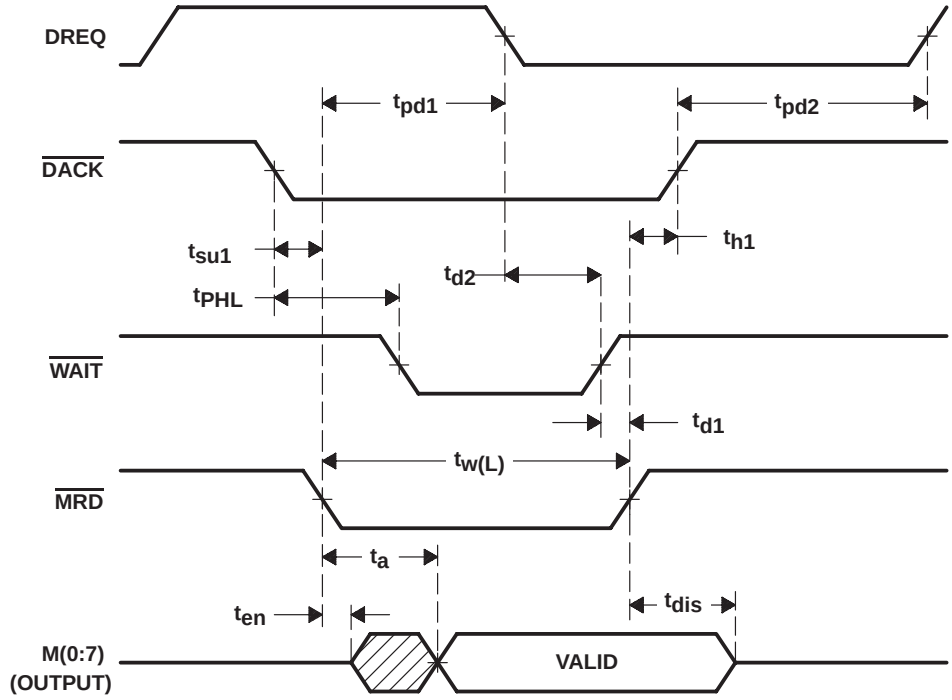
Figure 7–4. Processor Write Cycle (Multiplexed Address/Data Buses)

7.11 timing requirements over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
$t_{w(L)}$	Pulse duration, $\overline{MRD}$ low	See Figure 7–5	60		ns
t_{su1}	Setup time, $\overline{DACK}$ low before $\overline{MRD}$ low		0		ns
t_{h1}	Hold time, $\overline{DACK}$ low after $\overline{MRD}$ high		0		ns
t_{d1}	Delay time, $\overline{WAIT}$ high to $\overline{MRD}$ high		0		ns

7.12 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{en}	Enable time, $\overline{MRD}$ low to M(0:7) active	C_L = 15 PF, See Figure 7–5	5		ns
t_a	Access time, $\overline{MRD}$ low to M(0:7) valid			35	ns
t_{DIS}	Disable time, $\overline{MRD}$ high to M(0:7) in high-impedance state		5	25	ns
t_{pd1}	Propagation delay, $\overline{MRD}$ low to DREQ low			175	ns
t_{pd2}	Propagation delay, $\overline{DACK}$ high to DREQ high			275	ns
t_{d2}	Delay time, DREQ low to $\overline{WAIT}$ high			55	ns
t_{PHL}	Propagation delay, high to low, $\overline{DACK}$ low to $\overline{WAIT}$ low			25	ns



NOTE: $\overline{WAIT}$ timing is shown only for demand-mode transfers that have automatically entered the nondemand mode due to inadequate FIFO status or approaching end of transfer. The $\overline{WAIT}$ signal is inactive for nondemand transfers.

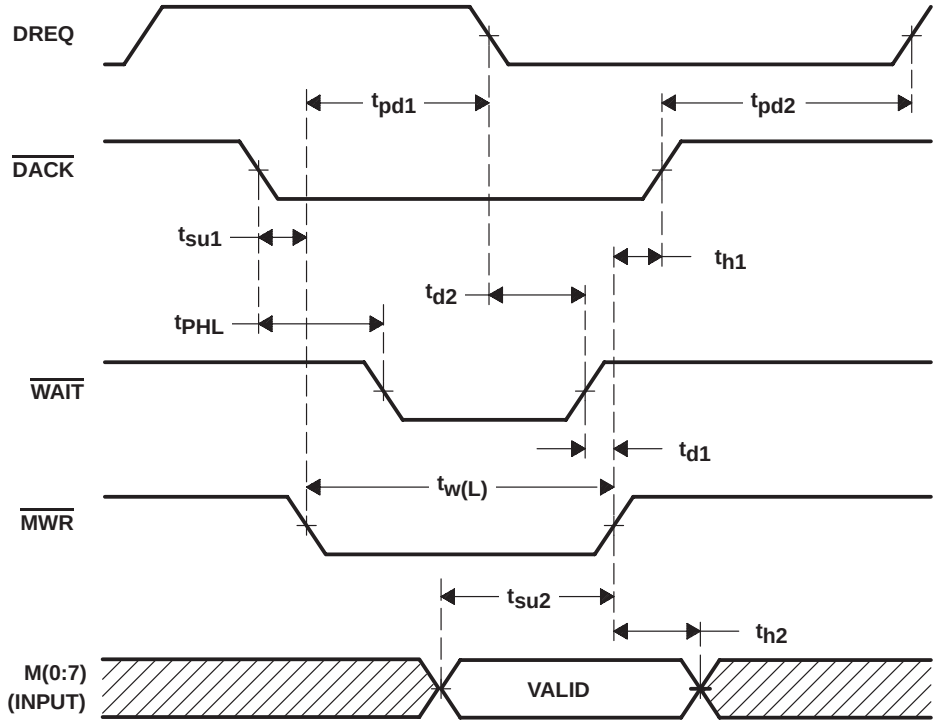
Figure 7–5. DMA Read – Non-Byte-Stack, Nondemand Mode

7.13 timing requirements over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
$t_{w(L)}$	Pulse duration, $\overline{MWR}$ low	See Figure 7–6	60		ns
t_{su1}	Setup time, $\overline{DACK}$ low before $\overline{MWR}$ low		0		ns
t_{h1}	Hold time, $\overline{DACK}$ low after $\overline{MWR}$ high		0		ns
t_{su2}	Setup time, M(0:7) valid before $\overline{MWR}$ high		40		ns
t_{h2}	Hold time, M(0:7) valid after $\overline{MWR}$ high		0		ns
t_{d1}	Delay time, $\overline{WAIT}$ high to $\overline{MWR}$ high		0		ns

7.14 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{pd1}	Propagation delay, $\overline{MWR}$ low to DREQ low	C_L = 15 pF, See Figure 7–6		175	ns
t_{pd2}	Propagation delay, $\overline{DACK}$ high to DREQ high			275	ns
t_{d2}	Delay time, DREQ low to $\overline{WAIT}$ high			55	ns
t_{PHL}	Propagation delay, high to low, $\overline{DACK}$ low to $\overline{WAIT}$ low			25	NS



NOTE: $\overline{WAIT}$ timing is shown only for demand-mode transfers that have automatically entered the nondemand mode due to inadequate FIFO status or approaching end of transfer. The $\overline{WAIT}$ signal is inactive for nondemand transfers.

Figure 7–6. DMA Write – Non-Byte-Stack, Nondemand Mode

7.15 timing requirements over recommended operating free-air temperature and supply voltage ranges

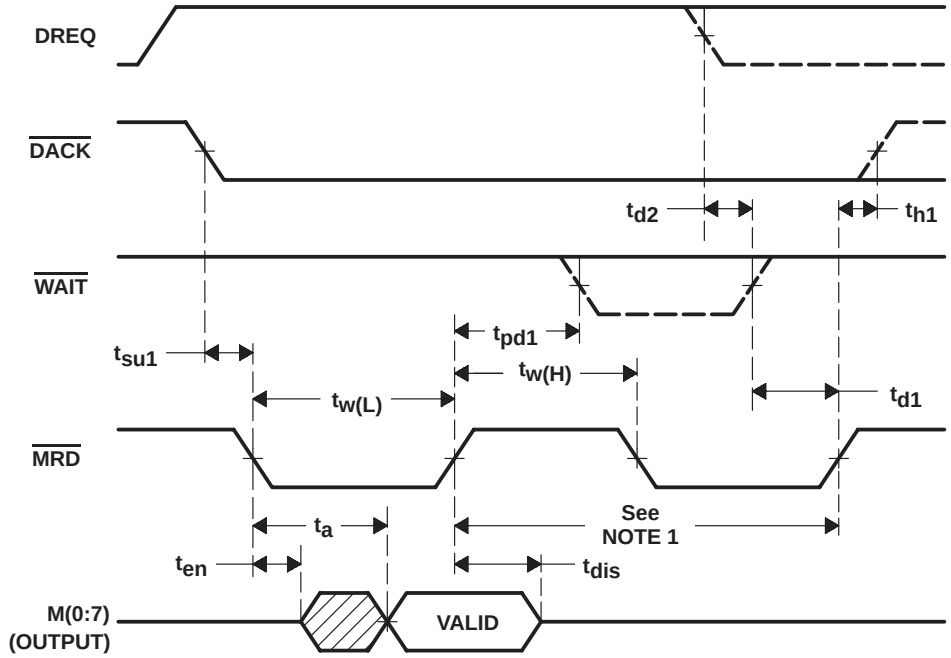
PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
$t_{w(L)}$	Pulse duration, $\overline{MRD}$ low	See Figure 7–7	60		ns
$t_{w(H)}$	Pulse duration, $\overline{MRD}$ high		60		ns
t_{su1}	Setup time, $\overline{DACK}$ low before $\overline{MRD}$ low		0		ns
t_{h1}	Hold time, $\overline{DACK}$ low after $\overline{MRD}$ high		0		ns
t_{d1}	Delay time, $\overline{WAIT}$ high to $\overline{MRD}$ high		0		ns

7.16 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
t_{en}	Enable time, $\overline{MRD}$ low to M(0:7) active	C_L = 15 pF, See Figure 7–7	5			ns
t_a	Access time, MRD low to M(0:7) valid				35	ns
t_{dis}	Disable time, $\overline{MRD}$ high to M(0:7) in high-impedance state		5		25	ns
t_{Pd1}	Propagation delay, $\overline{MRD}$ high to $\overline{WAIT}$ low (see Note 1)				225	ns
t_{d2}	Delay time, DREQ low to $\overline{WAIT}$ high		45	50		ns

† All typical values are at V_{CC} = 5 V and T_A = 25°C.

NOTE 5: A wait state (due to the approaching end of a transfer or inadequate FIFO status) may occur as much as t_{pd1} ns after the rising edge of $\overline{MRD}$ as the SN75C091A automatically enters the nondemand transfer mode. Thus, the rising edge of a successive $\overline{MRD}$ pulse must not occur until the $\overline{WAIT}$ line has been observed inactive (high) following the t_{pd1} delay. The chip will return to the demand mode and continue unless the transfer is nearly complete (in which case it will remain in the nondemand mode).



NOTE: "– – – – " waveform indicates change to the nondemand mode.

Figure 7–7. DMA Read – Non-Byte-Stack, Demand Mode

7.17 timing requirements over recommended operating free-air temperature and supply voltage ranges

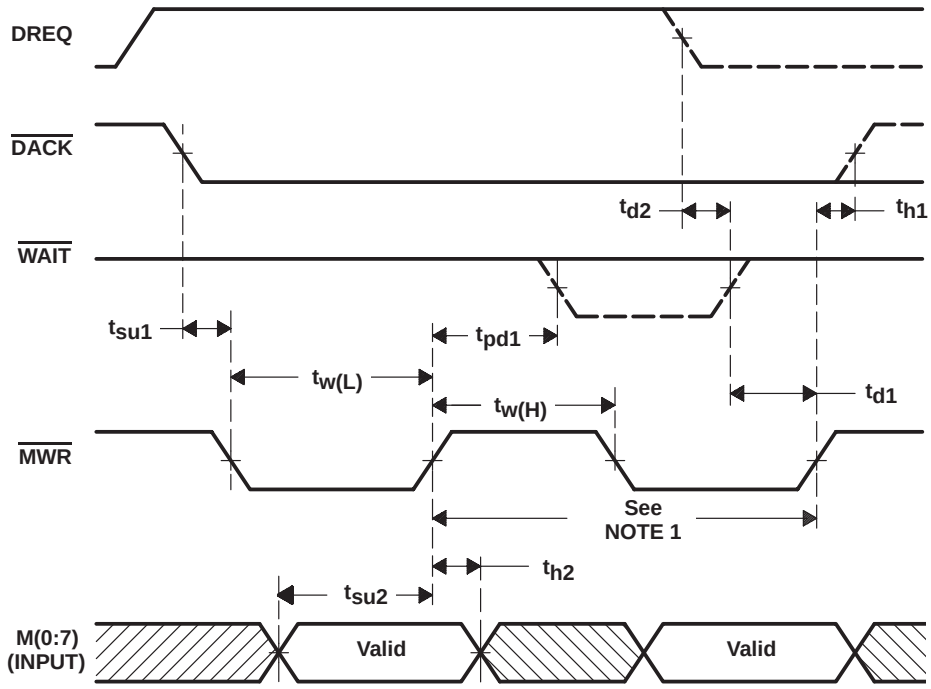
PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
$t_{w(L)}$	Pulse duration, $\overline{MWR}$ low	See Figure 7–8	60		ns
$t_{w(H)}$	Pulse duration, $\overline{MWR}$ high		60		ns
t_{su1}	Setup time, $\overline{DACK}$ low before $\overline{MWR}$ low		0		ns
t_{h1}	Hold time, $\overline{DACK}$ low after $\overline{MWR}$ high		0		ns
t_{su2}	Setup time, M(0:7) valid before $\overline{MWR}$ high		40		ns
t_{h2}	Hold time, M(0:7) valid after $\overline{MWR}$ high		0		ns
t_{d1}	Delay time, $\overline{WAIT}$ high to $\overline{MWR}$ high		0		ns

7.18 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
t_{pd1}	Propagation delay, $\overline{MWR}$ high to $\overline{WAIT}$ low (see Note 5)	C_L = 15 pF, See Figure 7–8			225	NS
t_{d2}	Delay time, DREQ low to WAIT high		45	50		ns

† All typical values are at V_{CC} = 5 V and T_A = 25°C.

NOTE 5: A wait state (due to the approaching end of a transfer or inadequate FIFO status) may occur as much as t_{pd1} ns after the rising edge of $\overline{MWR}$ as the SN75C091A automatically enters the nondemand transfer mode. Thus, the rising edge of a successive $\overline{MWR}$ pulse must not occur until the $\overline{WAIT}$ line has been observed inactive (high) following the t_{pd1} delay. The chip will return to the demand mode and continue unless the transfer is nearly complete (in which case it will remain in the nondemand mode).



NOTE: "– – – – " waveform indicates change to the nondemand mode.

Figure 7–8. DMA Write – Non-Byte-Stack, Demand Mode

7.19 timing requirements over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
$t_{w(L)}$	Pulse duration, $\overline{BSSTB}$ low	See Figure 7–9	60		ns
t_{su1}	Setup time, $\overline{DACK}$ low before $\overline{BSSTB}$ low		0		ns
t_{h1}	Hold time, $\overline{DACK}$ low after $\overline{BSSTB}$ high		0		ns
t_{d1}	Delay time, $\overline{WAIT}$ high to $\overline{BSSTB}$ high		0		ns

7.20 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
t_{d2}	Delay time, DREQ high to beginning of byte stack load cycle	C_L = 15 pF, See Figure 7–9		50		ns
t_{d3}	Delay time, byte stack load cycle complete to DREQ low (see Note 5)			100		ns
t_{d4}	Delay time, DREQ low to $\overline{WAIT}$ high		45	50		ns
t_{pd1}	Propagation delay, $\overline{DACK}$ low to DREQ low (see Note 5)				225	ns
t_{pd2}	Propagation delay, $\overline{DACK}$ high to DREQ high				275	ns
t_{PHL}	Propagation delay, high to low, $\overline{DACK}$ low to $\overline{WAIT}$ low				25	ns

† All typical values are at V_{CC} = 5 V and T_A = 25°C.

NOTES: 5. The time at which DREQ goes low depends on the end of the byte stack load cycle as specified by t_{d3} if $\overline{DACK}$ goes low at least 3 clock cycles prior to the end of the byte stack load cycle. If $\overline{DACK}$ goes low after the period three clock cycles prior to the end of the byte stack load cycle, then DREQ goes low as specified by t_{pd1}.

6. If the final byte stack load of the transfer is not a full word, DREQ will be asserted after the byte stack load is complete.

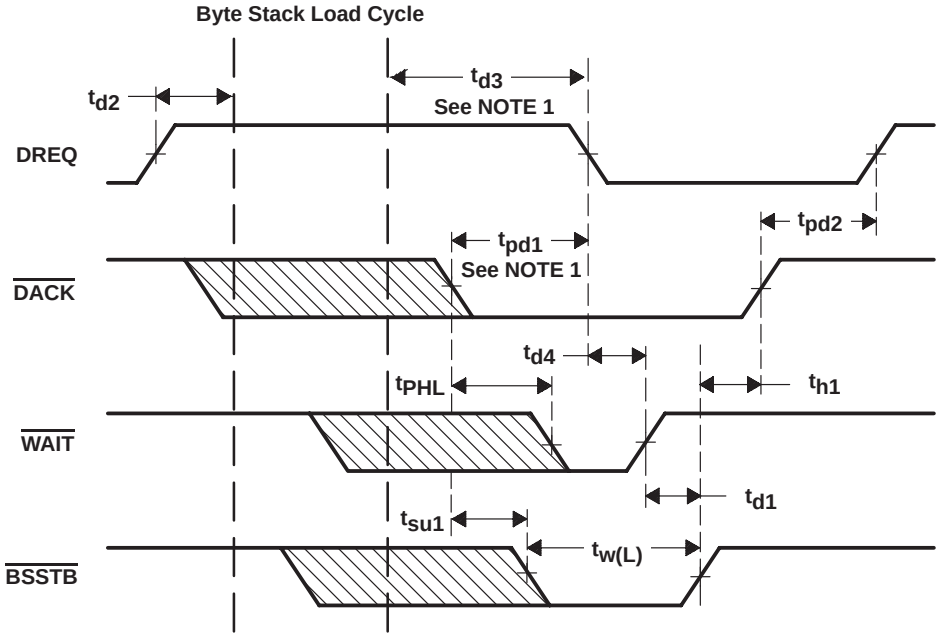


Figure 7–9. DMA Read – Byte Stack, Nondemand Mode

7.21 timing requirements over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
$t_{w(L)}$	Pulse duration, $\overline{BSSTB}$ low	See Figure 7–10	60		ns
t_{su1}	Setup time, $\overline{DACK}$ low before $\overline{BSSTB}$ low		0		ns
t_{h1}	Hold time, $\overline{DACK}$ low after $\overline{BSSTB}$ high		0		ns
t_{d1}	Delay time, $\overline{WAIT}$ high to $\overline{BSSTB}$ high		0		ns

7.22 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
t_{d2}	Delay time, $\overline{BSSTB}$ high to beginning of byte stack unload cycle	C_L = 15 pF, See Figure 7–10		200		ns
t_{d3}	Delay time, byte stack load cycle complete to DREQ high (see Note 1)			100		ns
t_{d4}	Delay time, DREQ low to $\overline{WAIT}$ high		45	50		ns
t_{pd1}	Propagation delay, $\overline{DACK}$ low to DREQ low				175	ns
t_{PHL}	Propagation delay, high to low, $\overline{DACK}$ low to $\overline{WAIT}$ low				25	ns

† All typical values are at V_{CC} = 5 V and T_A = 25°C.

NOTE 1 : t_{d3} applies when $\overline{DACK}$ is released at least 3 clock cycles before the end of the byte stack load cycle.

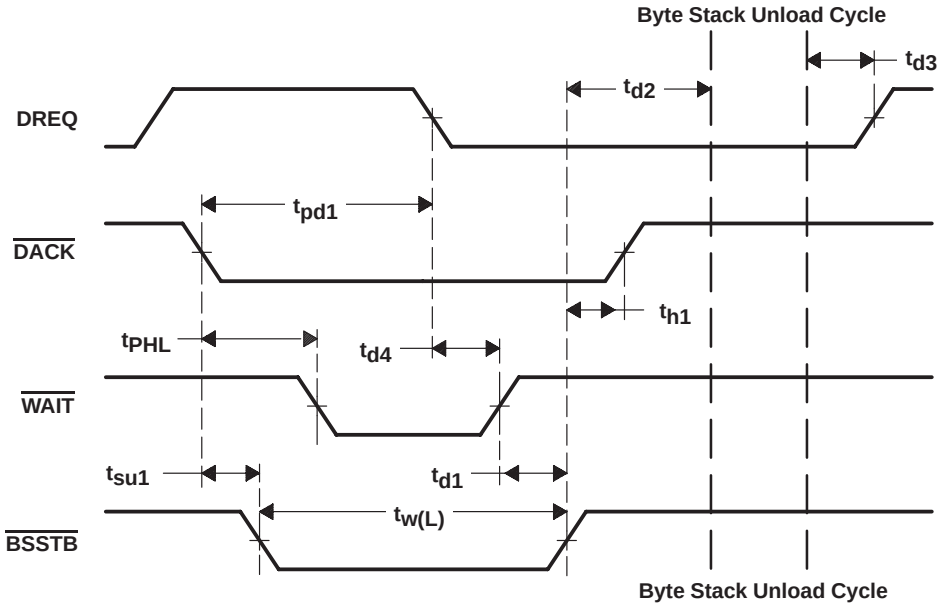


Figure 7–10. DMA Write – Byte Stack, Nondemand Mode

7.23 timing requirements over recommended operating free-air temperature and supply voltage ranges

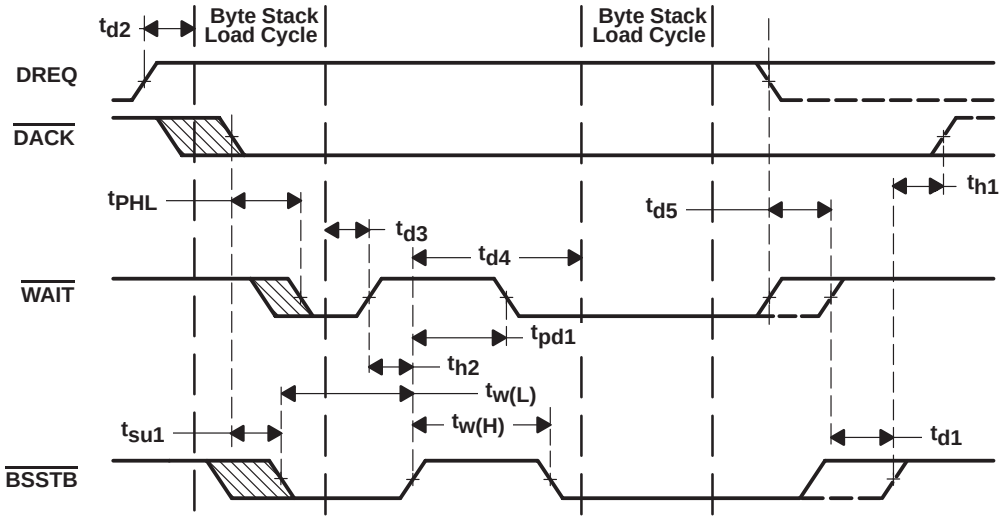
PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
$t_{w(L)}$	Pulse duration, $\overline{BSSTB}$ low	See Figure 7-11	60		ns
$t_{w(H)}$	Pulse duration, BSSTB high		60		ns
t_{su1}	Setup time, $\overline{DACK}$ low before $\overline{BSSTB}$ low		0		ns
t_{h1}	Hold time, $\overline{DACK}$ low after $\overline{BSSTB}$ high		0		ns
t_{d1}	Delay time, $\overline{WAIT}$ high to $\overline{BSSTB}$ high		0		ns

7.24 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
t_{d2}	Delay time, DREQ high to beginning of byte stack load cycle	C_L = 15 pF, See Figure 7–11		50		ns
t_{d3}	Delay time, byte stack load cycle complete to $\overline{WAIT}$ high (see Note 1)			150		ns
t_{d4}	Delay time, $\overline{BSSTB}$ high to beginning of byte stack load cycle				275	ns
td5	Delay time, DREQ low to $\overline{WAIT}$ high		45	50		ns
t_{pd1}	Propagation delay, $\overline{BSSTB}$ high to $\overline{WAIT}$ low				225	ns
t_{PHL}	Propagation delay, high to low, $\overline{DACK}$ low to $\overline{WAIT}$ low				25	ns

† All typical values are at V_{CC} = 5 V and T_A = 25°C.

NOTE 1: t_{d3} applies when $\overline{DACK}$ goes low at least 3 clock cycles before the end of the first byte stack load cycle.



NOTE: "– – – – " waveform indicates change to the nondemand mode.

Figure 7–11. DMA Read – Byte Stack, Demand Mode

7.25 timing requirements over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
$t_{w(L)}$	Pulse duration, $\overline{BSSTB}$ low	See Figure 7–12	60		ns
$t_{w(H)}$	Pulse duration, $\overline{BSSTB}$ high		60		ns
t_{su1}	Setup time, $\overline{DACK}$ low before $\overline{BSSTB}$ low		0		ns
t_{h1}	Hold time, $\overline{DACK}$ low after $\overline{BSSTB}$ high		0		ns
t_{d1}	Delay time, $\overline{WAIT}$ high to $\overline{BSSTB}$ high		0		ns

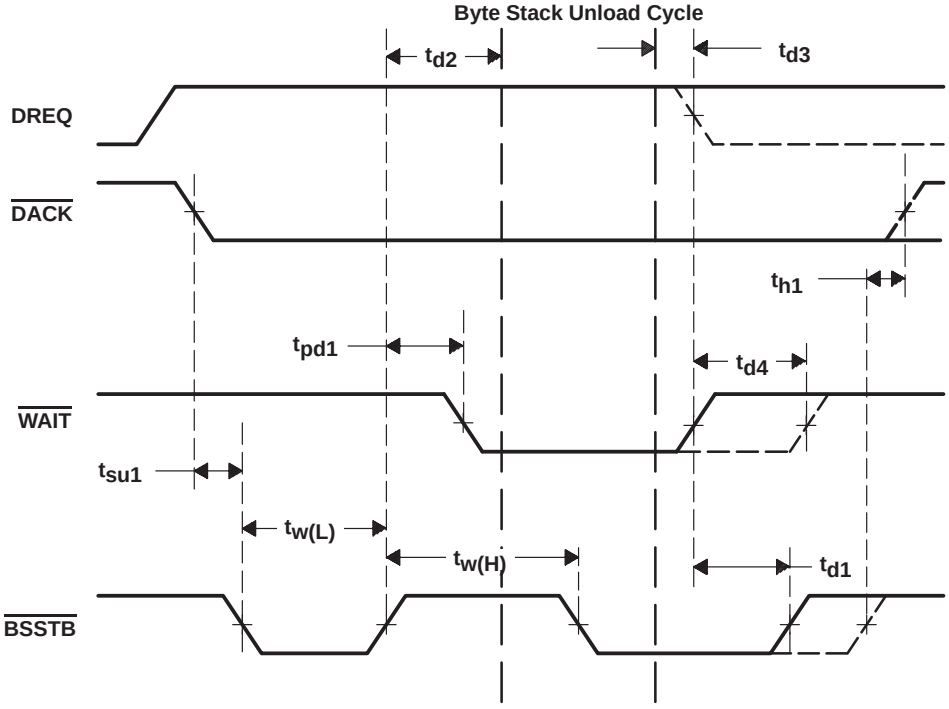
7.26 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
t_{d2}	Delay time, $\overline{BSSTB}$ high to beginningof byte stack unload cycle	C_L = 15 pF, See Figure 7–12			275	ns
t_{d3}	Delay time, byte stack unload cyclecomplete to $\overline{WAIT}$ high (see Note 1)			150		ns
t_{d4}	Delay time, DREQ low to $\overline{WAIT}$ high		45	50		ns
t_{pd1}	Propagation delay, $\overline{BSSTB}$ high to $\overline{WAIT}$ low				225	ns

† All typical values are at V_{CC} = 5 V and T_A = 25°C.

NOTE 1: It is assumed that $\overline{BSSTB}$ goes low for at least 3 clock cycles before the change to nondemand mode timing.

Byte Stack Unload Cycle



NOTE: "– – – –" waveform indicates change to the nondemand mode.

Figure 7–12. DMA Write – Byte Stack, Demand Mode

7.27 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
t_{d1}	Delay time, M(0:7) valid to $\overline{BSWR1}$ high	C_L = 15 pF, See Figure 7–13	15			ns
t_{d2}	Delay time, BSWR1 high to M(0:7) invalid		40			ns
$t_{w(L)}$	Pulse duration, $\overline{BSWR0}$ or $\overline{BSWR1}$		40			ns
t_c	Total byte stack load cycle time			200		ns

† All typical values are at V_{CC} = 5 V and T_A = 25°C.

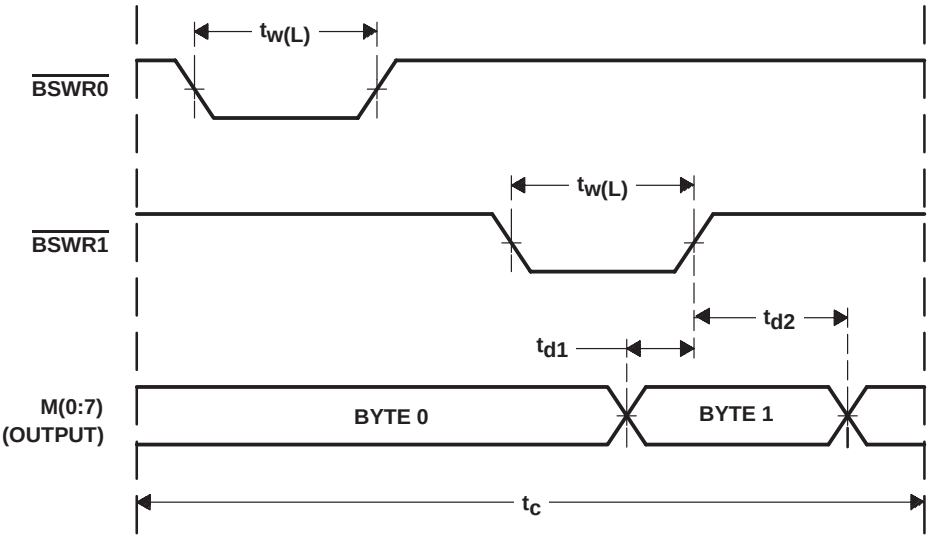


Figure 7–13. Byte Stack Load Cycle (16-bit)

7.28 timing requirements over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{d1}	Delay time, $\overline{\text{BSRD0}}$ low to M(0:7) valid	See Figure 7–14		20	ns
t_{d2}	Delay time, BSRD0 high to M(0:7) invalid		3		ns

7.29 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
$t_{w(L)}$	Pulse duration, $\overline{\text{BSRD0}}$ low or $\overline{\text{BSRD1}}$ low	C_L = 15 pF, See Figure 7–14	40			ns
t_c	Total byte stack unload cycle time			200		ns

† All typical values are at V_{CC} = 5 V and T_A = 25°C.

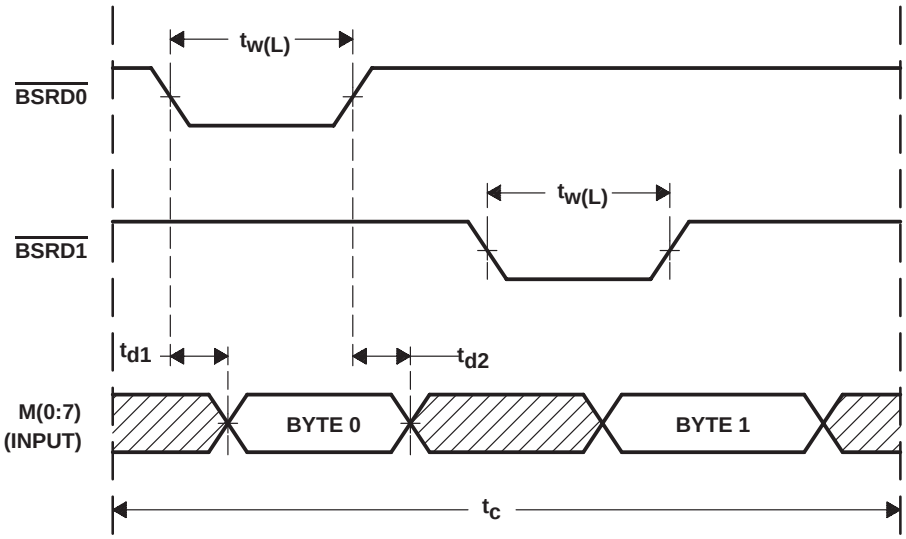


Figure 7–14. Byte Stack Unload Cycle (16-bit)

7.30 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
$t_{w(L)}$	Pulse duration, $\overline{BSWR}$ low	C_L = 15 pF, See Figure 7–15	40			ns
$t_{w(H)}$	Pulse duration, BSWR high		40			ns
t_{d1}	Delay time, BSEN(0:1) valid to $\overline{BSWR}$ low					
t_{d2}	Delay time, $\overline{BSWR}$ high to BSEN(0:1) invalid					
t_{d3}	Delay time, M(0:7) valid to $\overline{BSWR}$ high					
t_{d4}	Delay time, $\overline{BSWR}$ high to M(0:7) invalid					
t_c	Total byte stack load cycle time			400		ns

† All typical values are at V_{CC} = 5 V and T_A = 25°C.

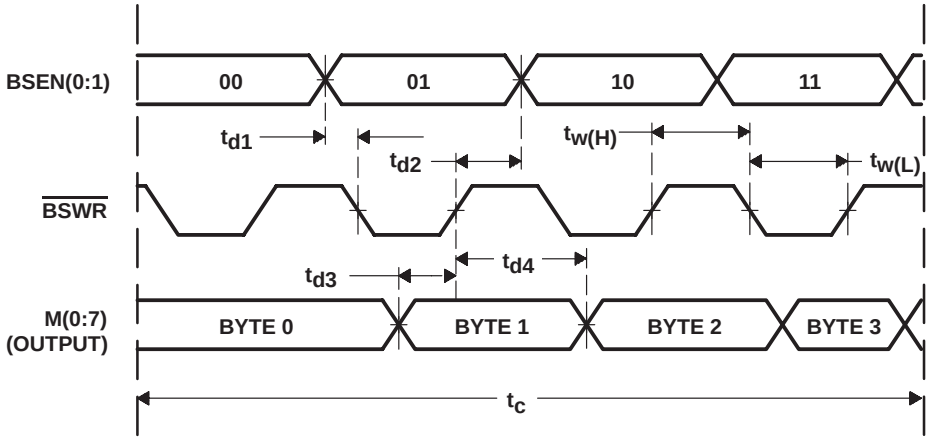


Figure 7–15. Byte Stack Load Cycle (24- or 32-Bit)

7.31 timing requirements over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{d1}	Delay time, $\overline{BSRD}$ low to M(0:7) valid	See Figure 7–16		35	ns
t_{d2}	Delay time, BSRD high to M(0:7) invalid		5		ns

7.32 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
$t_{w(L)}$	Pulse duration, $\overline{BSRD}$ low	C_L = 15 pF, See Figure 7–16	40			ns
$t_{w(H)}$	Pulse duration, $\overline{BSRD}$ high		40			ns
t_{d3}	Delay time, BSEN(0:1) valid to $\overline{BSRD}$ high		0			ns
t_{d4}	Delay time, $\overline{BSRD}$ high to BSEN(0:1) invalid		30			ns
t_c	Total byte stack unload cycle time			400		ns

† All typical values are at V_{CC} = 5 V and T_A = 25°C.

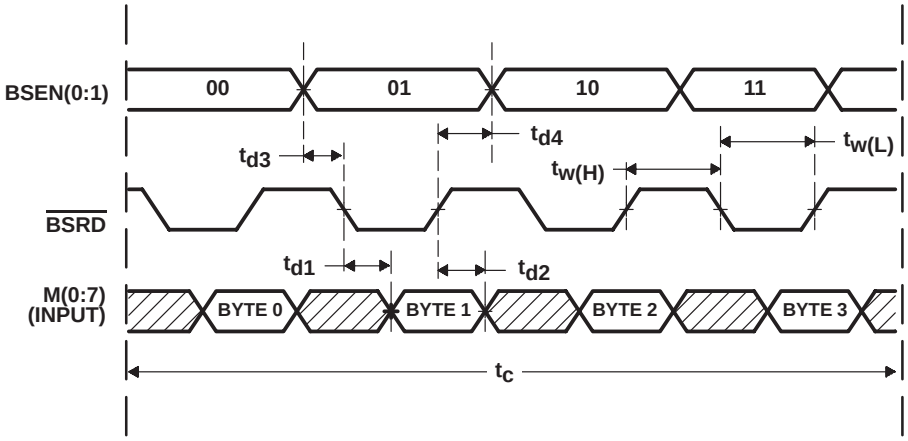


Figure 7–16. Byte Stack Unload Cycle (24- or 32-Bit)

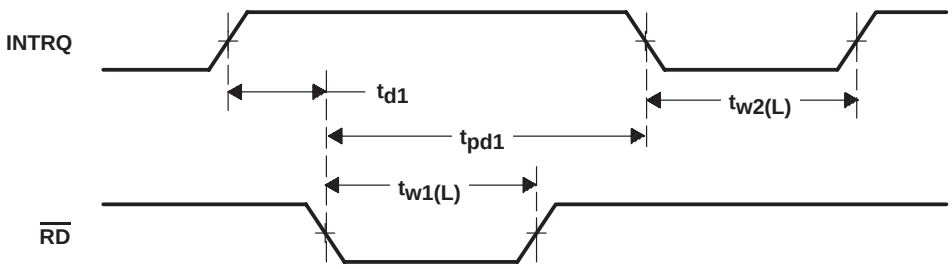
7.33 timing requirements over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{d1}	Delay time, INTRQ high to $\overline{RD}$ low	See Figure 7–17	0		ns
$t_{w1(L)}$	Pulse duration, RD low		110		ns

7.34 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{pd1}	Propagation delay, $\overline{RD}$ low to INTRQ low	C_L = 15 pF, See Figure 7–17	200		ns
$t_{w2(L)}$	Pulse duration, INTRQ low		0†		ns

† An additional interrupt queued during the read leaves INTRQ active following the interrupt register access.



NOTE: Timing for chip select and address lines is shown in earlier timing diagrams. $\overline{RD}$ is for functional or error interrupt status registers.

Figure 7–17. Interrupt Clear

7.35 timing requirements over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
$t_{W(L)}$	Pulse duration, MR low	See Figure 7–18	200		ns

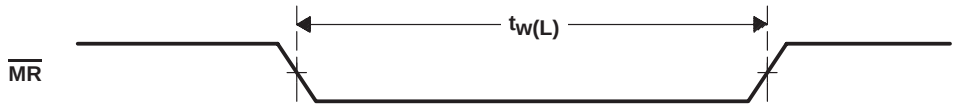
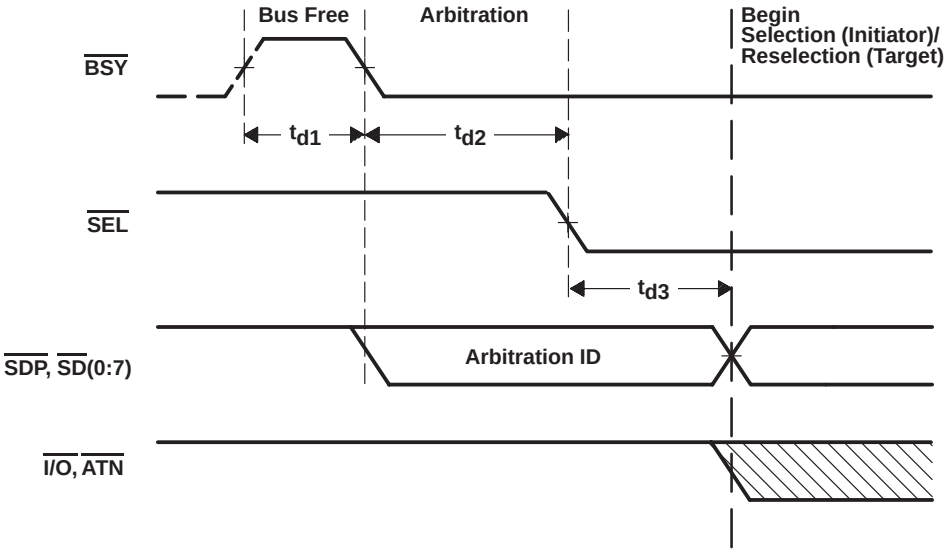


Figure 7–18. Master Clear

7.36 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{d1}	Delay time, BSY high to SDP, SD(0:7) valid	C_L = 50 pF, See Figure 7–19	1200	1350	ns
t_{d2}	Delay time, BSY low to SEL low (see Note 1)		2200	2300	ns
t_{d3}	Delay time, SEL low to start of SCSI selection/reselection phase		1200		ns

NOTE 1: Assumes that arbitration is won.

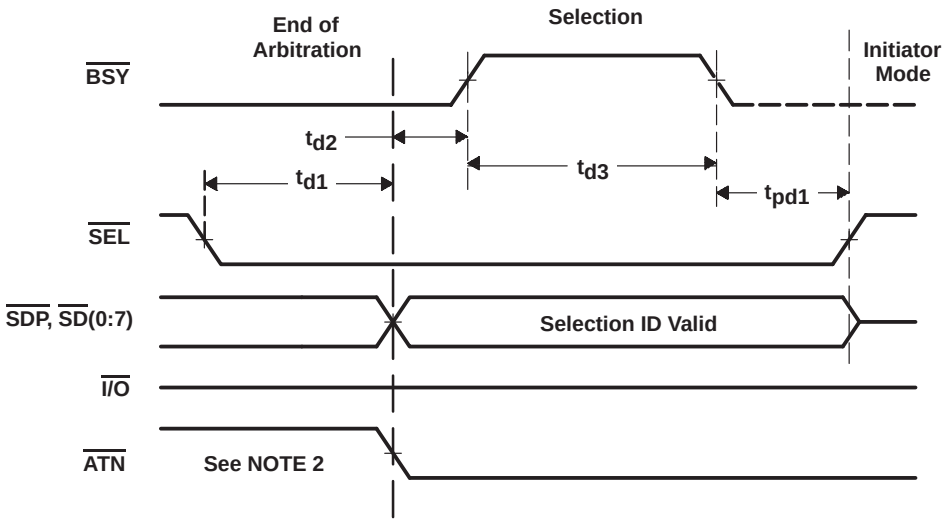


NOTE: "– – – – " waveform indicates change to the nondemand mode.

Figure 7–19. SCSI Bus Arbitration

7.37 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{d1}	Delay time, $\overline{SEL}$ low to $\overline{SDP}$, $\overline{SD}$(0:7) valid and $\overline{ATN}$ low	C_L = 50 pF, See Figure 7–20		1300	ns
t_{d2}	Delay time, SDP, SD(0:7) valid and ATN low to BSY high		100	160	ns
t_{d3}	Delay time, $\overline{BSY}$ released (high) by initiator to SN75C091A check for $\overline{BSY}$ asserted (low) by target		400		ns
t_{pd1}	Propagation delay, $\overline{BSY}$ asserted (low) by target to $\overline{SEL}$ high and $\overline{SDP}$, $\overline{SD}$(0:7) invalid		100	200	ns



NOTES: 1. " - - - - " waveform for $\overline{BSY}$ (wired-OR signal) indicates that $\overline{BSY}$ is being driven by the target.
2. Applies to Select with $\overline{ATN}$ command only.

Figure 7–20. SCSI Bus Selection (Initiator)

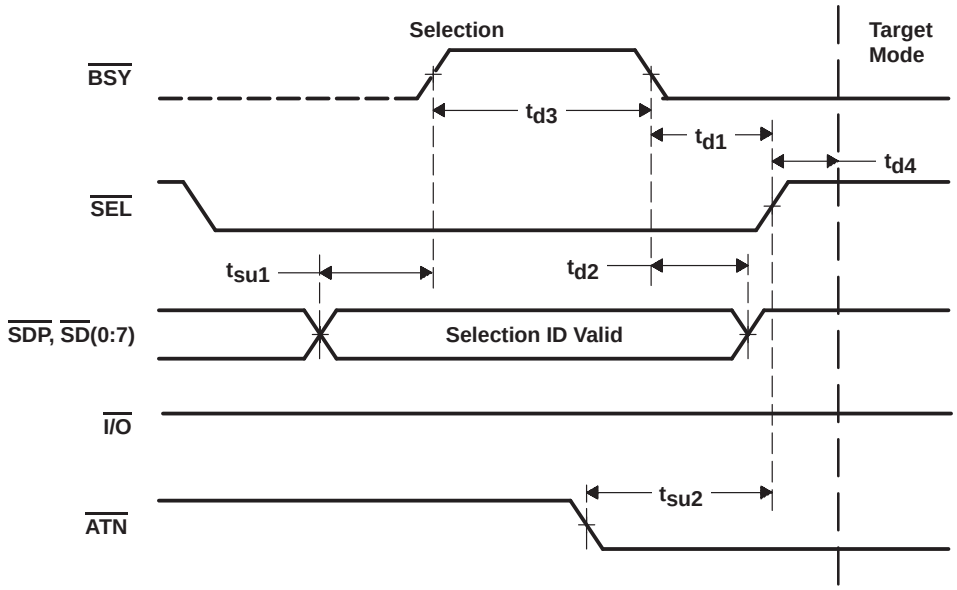
7.38 timing requirements over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{su1}	Setup time, $\overline{SDP}$, $\overline{SD}$(0:7) valid before $\overline{BSY}$ released (high) by initiator	See Figure 7–21	0		ns
t_{su2}	Setup time, $\overline{ATN}$ low before $\overline{SEL}$ high (see Note 1)		70		ns
t_{d1}	Delay time, $\overline{BSY}$ asserted (low) by target to $\overline{SEL}$ high		0		ns
td2	Delay time, $\overline{BSY}$ asserted (low) by target to $\overline{SDP}$, $\overline{SD}$(0:7) invalid		0		ns

7.39 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
td3	Delay time, $\overline{BSY}$ released (high) by initiator to $\overline{BSY}$ asserted (low) by target	C_L = 50 pF, See Figure 7–21	400	650	ns
t_{d4}	Delay time, $\overline{SEL}$ high to start of target mode			150	ns

NOTE 1: This setup time ensures that $\overline{ATN}$ will be honored by Wait for Select with $\overline{ATN}$ or Wait for Select without $\overline{ATN}$ commands at the completion of the selection phase.



NOTE: " - - - - " waveform for $\overline{BSY}$ (wired-OR signal) indicates that $\overline{BSY}$ is being driven by the initiator.

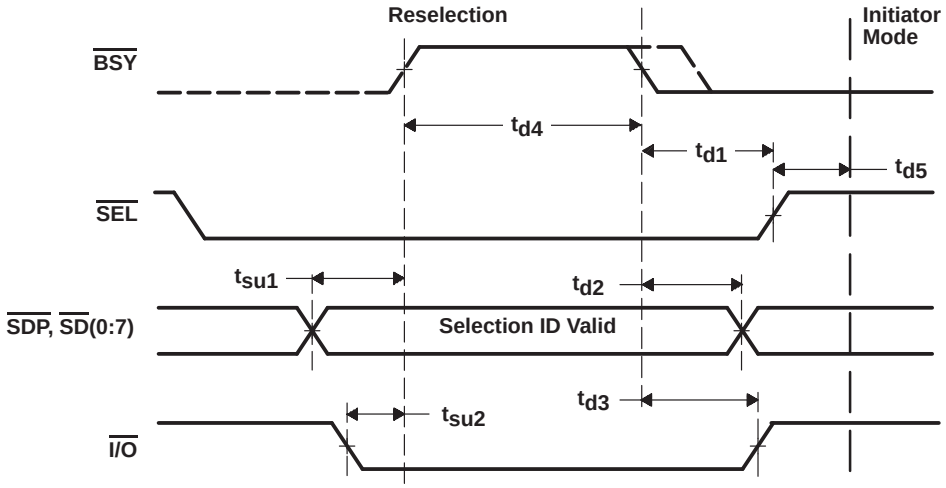
Figure 7–21. SCSI Bus Selection (Target)

7.40 timing requirements over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{su1}	Setup time, $\overline{SDP}$, $\overline{SD}$(0:7) valid before $\overline{BSY}$ released (high) by target	See Figure 7–22	0		ns
t_{d1}	Delay time, $\overline{BSY}$ asserted (low) by initiator to $\overline{SEL}$ high		0		ns
t_{d2}	Delay time, $\overline{BSY}$ asserted (low) by initiator to $\overline{SDP}$, $\overline{SD}$(0:7) invalid		0		ns
t_{su2}	Setup time, $\overline{I/O}$ low before $\overline{BSY}$ released (high) by target		0		ns
t_{d3}	Delay time, $\overline{BSY}$ asserted (low) by initiator to $\overline{I/O}$ high		0		ns

7.41 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{d4}	Delay time, $\overline{BSY}$ released (high) by target to $\overline{BSY}$ asserted (low) by initiator	C_L = 15 pF, See Figure 7–22	400	650	ns
t_{d5}	Delay time, $\overline{SEL}$ high to start of initiator mode			150	ns

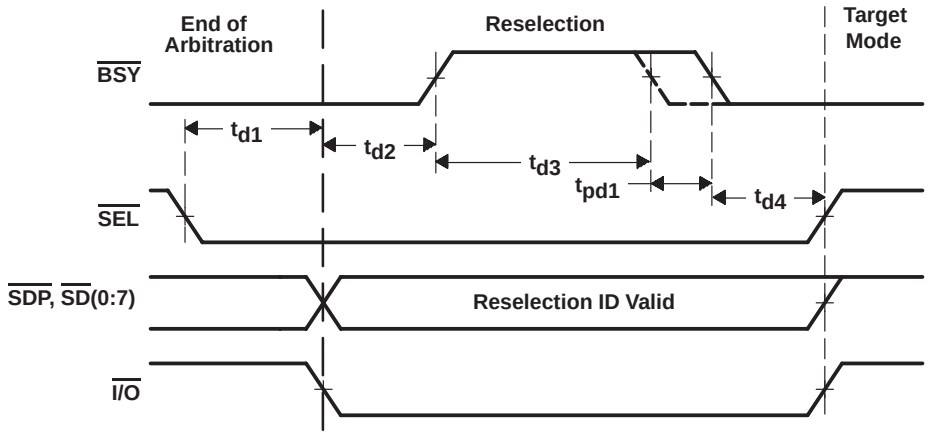


NOTE: " - - - - " waveform for $\overline{BSY}$ (wired-OR signal) indicates that $\overline{BSY}$ is being driven by the target.

Figure 7–22. SCSI Bus Reselection (Initiator)

7.42 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{d1}	Delay time, $\overline{SEL}$ low to $\overline{SDP}$, $\overline{SD}$(0:7) valid and $\overline{I/O}$ low	C_L = 50 pF, See Figure 7–23		1300	ns
t_{d2}	Delay time, SDP, SD(0:7) valid to $\overline{BSY}$ deasserted (high) by target		100	160	ns
t_{d3}	Delay time, $\overline{BSY}$ deasserted (high) by target to SN75C091A check for $\overline{BSY}$ asserted (low) by initiator		400		ns
t_{d4}	Delay time, $\overline{BSY}$ asserted (low) by target to $\overline{SEL}$ high, $\overline{SDP}$, $\overline{SD}$(0:7) invalid and $\overline{I/O}$ high		90	150	ns
t_{pd1}	Propagation delay, $\overline{BSY}$ asserted (low) by initiator to $\overline{BSY}$ also asserted (low) by target		50	150	ns



NOTE: " - - - - " waveform for $\overline{BSY}$ (wired-OR signal) indicates that $\overline{BSY}$ is being driven by the initiator.

Figure 7–23. SCSI Bus Reselection (Target)

7.43 timing requirements over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{su1}	Setup time, $\overline{MSG}$, $\overline{C/D}$ valid and $\overline{I/O}$ low before $\overline{REQ}$ low	See Figure 7–24	400		ns
t_{d1}	Delay time, $\overline{ACK}$ high to $\overline{MSG}$, $\overline{C/D}$ invalid and $\overline{I/O}$ high		0		ns
t_{d2}	Delay time, $\overline{ACK}$ low to $\overline{REQ}$ high		0		ns
t_{d3}	Delay time, $\overline{ACK}$ high to $\overline{REQ}$ low		0		ns
t_{su2}	Setup time, $\overline{SDP}$, $\overline{SD}$(0:7) valid before $\overline{REQ}$ low		30		ns
t_{d4}	Delay time, $\overline{ACK}$ low to $\overline{SDP}$, $\overline{SD}$(0:7) invalid		0		ns

7.44 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{PHL}	Propagation delay, high to low, $\overline{REQ}$ low to $\overline{ACK}$ low	C_L = 50 pF, See Figure 7–24	5		ns
t_{PLH}	Propagation delay, low to high, $\overline{REQ}$ high to $\overline{ACK}$ high		5		ns
t_{dis}	Disable time, $\overline{I/O}$ low to initiator disable data		30		ns
t_{d5}	Delay time, $\overline{ATN}$ low to $\overline{ACK}$ high (see Note 1)		10		ns

NOTE 1: Applicable only when $\overline{ATN}$ is set automatically due to a parity error.

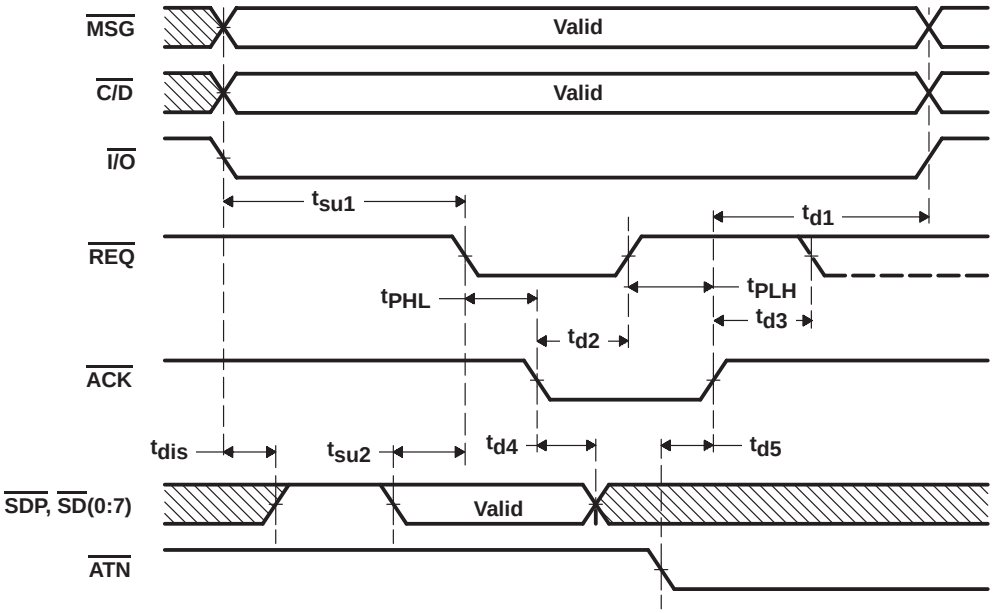


Figure 7–24. SCSI Bus Asynchronous Information Transfer In (Initiator) (TARGET TO INITIATOR)

7.45 timing requirements over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{d1}	Delay time, $\overline{REQ}$ low to $\overline{ACK}$ low	See Figure 7–25	0		ns
t_{d2}	Delay time, REQ high to ACK high		0		ns

7.46 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{d3}	Delay time, $\overline{MSG}$, $\overline{C/D}$ valid and $\overline{I/O}$ low to $\overline{SDP}$, $\overline{SD}$(0:7) active	C_L = 50 pF, See Figure 7–25	800		ns
t_{d4}	Delay time, $\overline{ACK}$ high to $\overline{MSG}$, $\overline{C/D}$ invalid and $\overline{I/O}$ high		200		ns
t_{d5}	t_{d5} Delay time, $\overline{SDP}$, $\overline{SD}$(0:7) valid to $\overline{REQ}$ low		55		ns
t_{d6}	Delay time, $\overline{REQ}$ high to $\overline{SDP}$, $\overline{SD}$(0:7) invalid		0		ns
t_{PLH}	Propagation delay, low to high, $\overline{ACK}$ low to $\overline{REQ}$ high		5		ns
t_{PHL}	t_{PHL} Propagation delay, $\overline{ACK}$ high to $\overline{REQ}$ low		5		ns

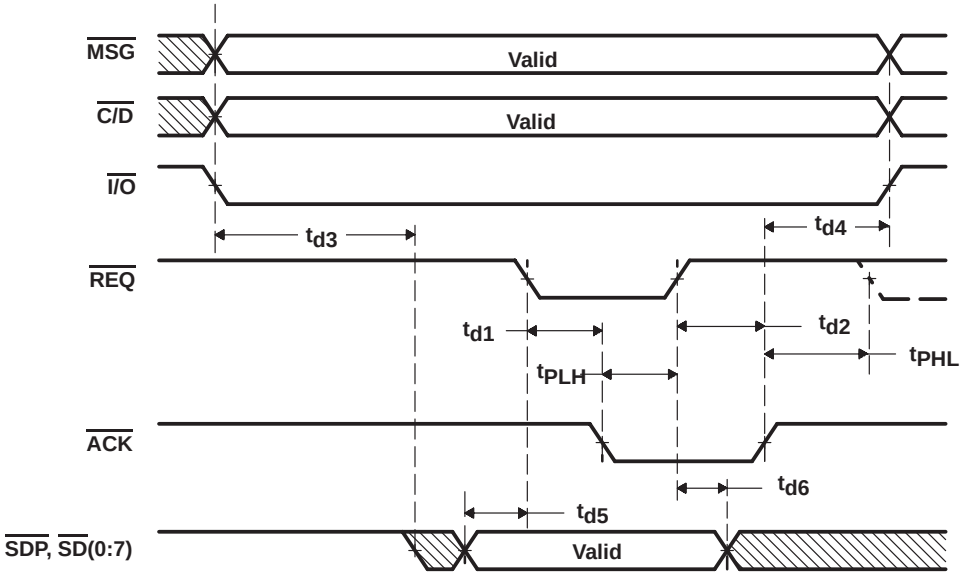


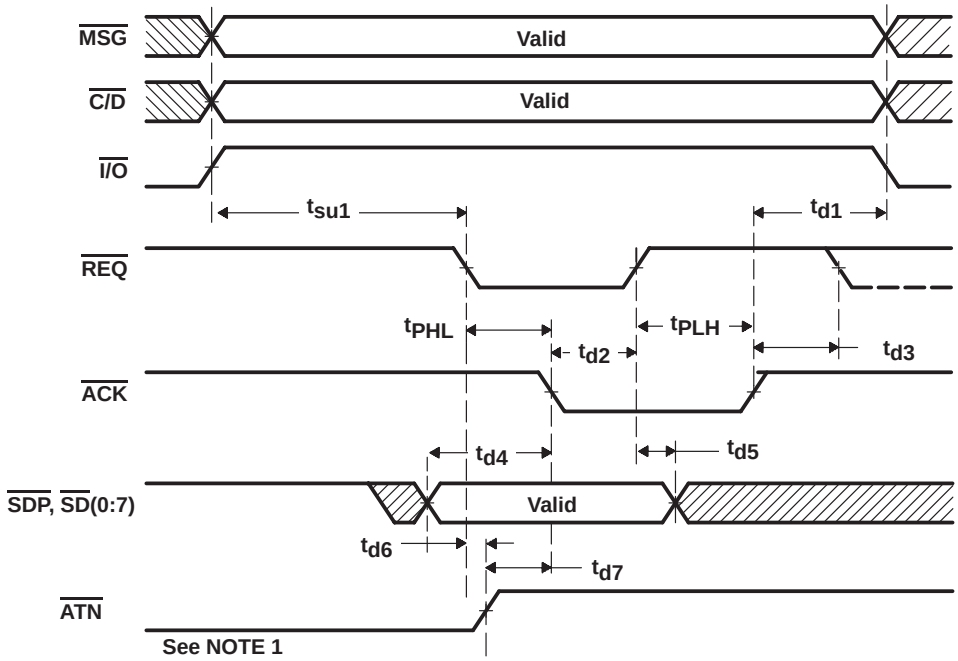
Figure 7–25. SCSI Bus Asynchronous Information Transfer In (Target) (TARGET TO INITIATOR)

7.47 timing requirements over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{su1}	Setup time, $\overline{MSG}$, $\overline{C/D}$ valid and $\overline{I/O}$ high before $\overline{REQ}$ low	See Figure 7–26	400		ns
t_{d1}	Delay time, $\overline{ACK}$ high to $\overline{MSG}$, $\overline{C/D}$ invalid and $\overline{I/O}$ low		0		ns
t_{d2}	Delay time, $\overline{ACK}$ low to $\overline{REQ}$ high		0		ns
t_{d3}	Delay time, $\overline{ACK}$ high to $\overline{REQ}$ low		0		ns

7.48 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{PHL}	Propagation delay, $\overline{REQ}$ low to $\overline{ACK}$ low	C_L = 50 pF, See Figure 7–26	5		ns
t_{PLH}	Propagation delay, $\overline{REQ}$ high to $\overline{ACK}$ high		5		ns
t_{d4}	Delay time, $\overline{SDP}$, $\overline{SD}$(0:7) valid to $\overline{ACK}$ low		55		ns
t_{d5}	Delay time, $\overline{REQ}$ high to $\overline{SDP}$, $\overline{SD}$(0:7) invalid				
t_{d6}	Delay time, $\overline{REQ}$ low to $\overline{ATN}$ high		0		ns
t_{d7}	Delay time, $\overline{ATN}$ high to $\overline{ACK}$ low		90		ns



NOTE 1: Applicable only for last byte of message-out phase when HAAM bit in control register is set to 0.

Figure 7–26. SCSI Bus Asynchronous Information Transfer Out (Initiator) (INITIATOR TO TARGET)

7.49 timing requirements over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{d1}	Delay time, $\overline{REQ}$ low to $\overline{ACK}$ low	See Figure 7–27	0		ns
t_{d2}	Delay time, REQ high to ACK high		0		ns
t_{su1}	Setup time, $\overline{SDP}$, $\overline{SD}$(0:7) valid before $\overline{ACK}$ low		30		ns
t_{d3}	Delay time, REQ high to $\overline{SDP}$, $\overline{SD}$(0:7) invalid		0		ns

7.50 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{d4}	Delay time, $\overline{MSG}$, $\overline{C/D}$ valid and $\overline{I/O}$ high to $\overline{REQ}$ low	C_L =50 pF, See Figure 7–27	400		ns
t_{d5}	Delay time, ACK high to MSG, C/D invalid and I/O low		200		ns
t_{PLH}	Propagation delay, low to high, $\overline{ACK}$ low to $\overline{REQ}$ high		5		ns
t_{PHL}	Propagation delay, high to low, $\overline{ACK}$ high to $\overline{REQ}$ low		5		ns
t_{dis}	Disable time, $\overline{SDP}$, $\overline{SD}$(0:7) invalid to $\overline{I/O}$ high		0		ns

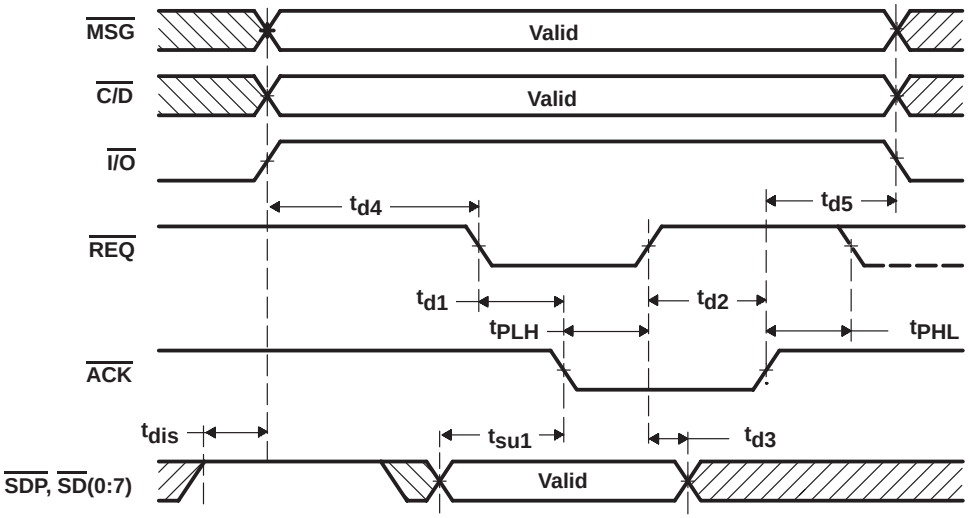


Figure 7–27. SCSI Bus Asynchronous Information Transfer Out (Target) (INITIATOR TO TARGET)

7.51 timing requirements over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{su1}	Setup time, $\overline{MSG}$, $\overline{C/D}$ valid and $\overline{I/O}$ low before $\overline{REQ}$ low	See Figure 7–28	400		ns
t_{d1}	Delay time, ACK high to MSG, C/D invalid and I/O high		0		ns
t_{su2}	Setup time, $\overline{SDP}$, $\overline{SD}$(0:7) valid before $\overline{REQ}$ low		30		ns
t_{h2}	Hold time, $\overline{SDP}$, $\overline{SD}$(0:7) valid after $\overline{REQ}$ low		45		ns
$t_{w1(L)}$	Pulse duration, $\overline{REQ}$ low		90		ns
$t_{w1(H)}$	Pulse duration, $\overline{REQ}$ high		90		ns

7.52 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
$t_{w2(L)}$	Pulse duration, $\overline{ACK}$ low	C_L = 50 pF, See Figure 7-28	90		ns
$t_{w2(H)}$	Pulse duration, ACK high		90		ns
t_{d2}	Delay time, $\overline{MSG}$, $\overline{C/D}$ valid and $\overline{I/O}$ low to $\overline{SDP}$, $\overline{SD}$(0:7) in high-impedance state		30		ns
t_{d3}	Delay time, $\overline{ATN}$ low to $\overline{ACK}$ high (see Note 1)		50		ns

NOTE 1: t_{d3} applies only when $\overline{ATN}$ is asserted automatically due to a parity error.

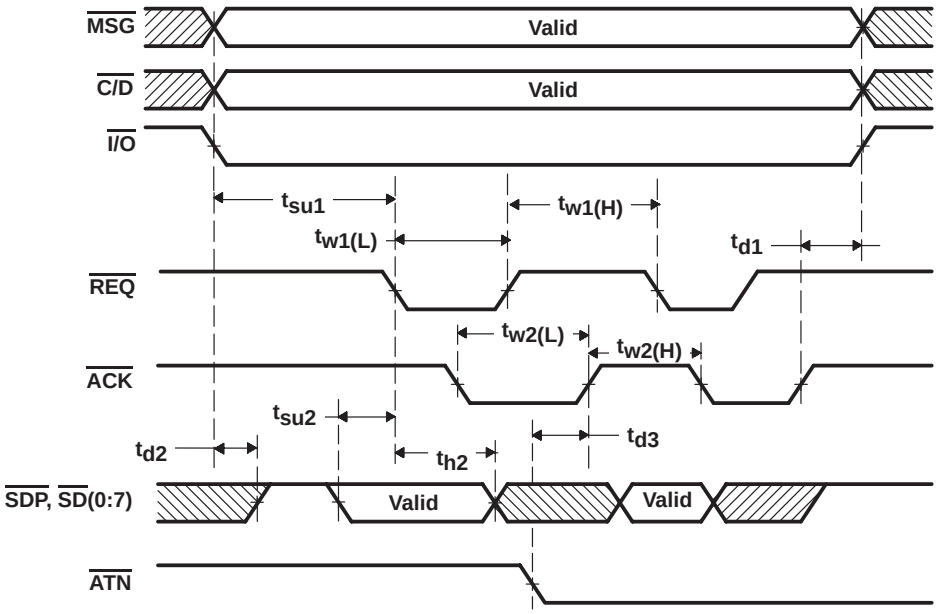


Figure 7–28. SCSI Bus Synchronous Data-In Transfer (Initiator) (TARGET TO INITIATOR)

7.53 timing requirements over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
$t_{w2(L)}$	Pulse duration, $\overline{ACK}$ low	See Figure 7–29	90		ns
$t_{w2(H)}$	Pulse duration, ACK high		90		ns

7.54 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{d1}	Delay time, $\overline{MSG}$, $\overline{C/D}$ valid and $\overline{I/O}$ low to $\overline{SDP}$, $\overline{SD}$(0:7) active	C_L = 50 pF, See Figure 7–29	800		ns
t_{d2}	Delay time, $\overline{ACK}$ high to $\overline{MSG}$, $\overline{C/D}$ invalid and $\overline{I/O}$ high		200		ns
t_{d3}	Delay time, $\overline{SDP}$, $\overline{SD}$(0:7) valid to $\overline{REQ}$ low		50		ns
t_{d4}	Delay time, $\overline{REQ}$ low to $\overline{SDP}$, $\overline{SD}$(0:7) invalid		100		ns
$t_{w1(L)}$	Pulse duration, $\overline{REQ}$ low		90		ns
$t_{w1(H)}$	Pulse duration, $\overline{REQ}$ high		90		ns

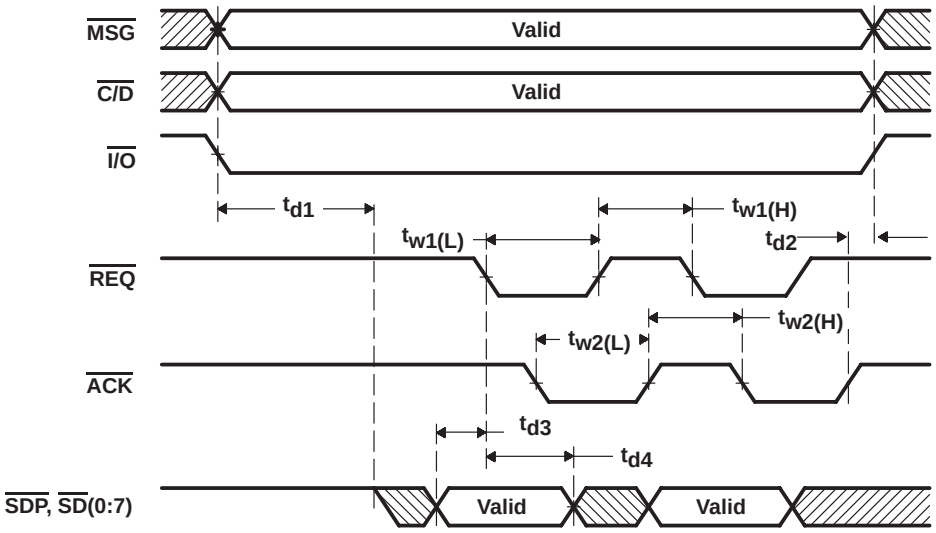


Figure 7–29. SCSI Bus Synchronous Data-In Transfer (Target) (TARGET TO INITIATOR)

7.55 timing requirements over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{su1}	Setup time, $\overline{MSG}$, $\overline{C/D}$ valid and $\overline{I/O}$ high before $\overline{REQ}$ low	See Figure 7–30	400		ns
t_{d1}	Delay time, ACK high to MSG, C/D invalid and I/O low		0		ns
$t_{w1(L)}$	Pulse duration, $\overline{REQ}$ low		90		ns
$t_{w1(H)}$	Pulse duration, $\overline{REQ}$ high		90		ns

7.56 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{d2}	Delay time, $\overline{SDP}$, $\overline{SD}$(0:7) valid to $\overline{ACK}$ low	C_L = 50 pF, See Figure 7–30	55		ns
t_{d3}	Delay time, ACK low to SDP, SD(0:7) invalid		100		ns
$t_{w2(L)}$	Pulse duration, $\overline{ACK}$ low		90		ns
$t_{w2(H)}$	Pulse duration, $\overline{ACK}$ high		90		ns

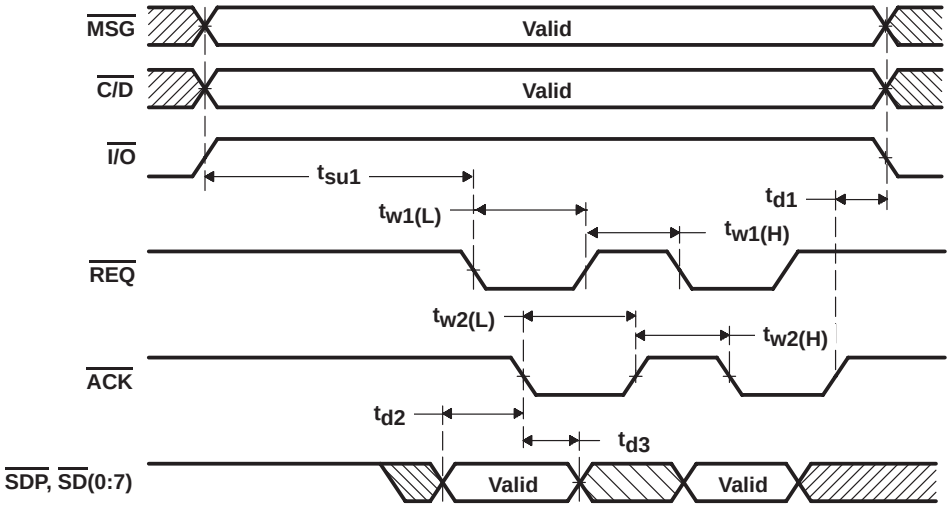


Figure 7–30. SCSI Bus Synchronous Data-Out Transfer (Initiator) (INITIATOR TO TARGET)

7.57 timing requirements over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{su1}	Setup time, $\overline{SDP}$, $\overline{SD}$(0:7) valid before $\overline{ACK}$ low	See Figure 7–31	30		ns
t_{d1}	Hold time, SDP, SD(0:7) valid after ACK low		45		ns
$t_{w2(L)}$	Pulse duration, $\overline{ACK}$ low		90		ns
$t_{w2(H)}$	Pulse duration, $\overline{ACK}$ high		90		ns

7.58 timing characteristics over recommended operating free-air temperature and supply voltage ranges

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{d1}	Delay time, $\overline{MSG}$, $\overline{C/D}$ valid and $\overline{I/O}$ high to $\overline{REQ}$ low	C_L = 50 pF, See Figure 7–31	400		ns
t_{d2}	Delay time, $\overline{ACK}$ high to $\overline{MSG}$, $\overline{C/D}$ invalid and $\overline{I/O}$ low		200		ns
$t_{w1(L)}$	Pulse duration, $\overline{REQ}$ low		90		ns
$t_{w1(H)}$	Pulse duration, $\overline{REQ}$ high		90		ns
t_{dis}	Disable time, $\overline{SDP}$, $\overline{SD}$(0:7) in high-impedance state to $\overline{I/O}$ high		0		ns

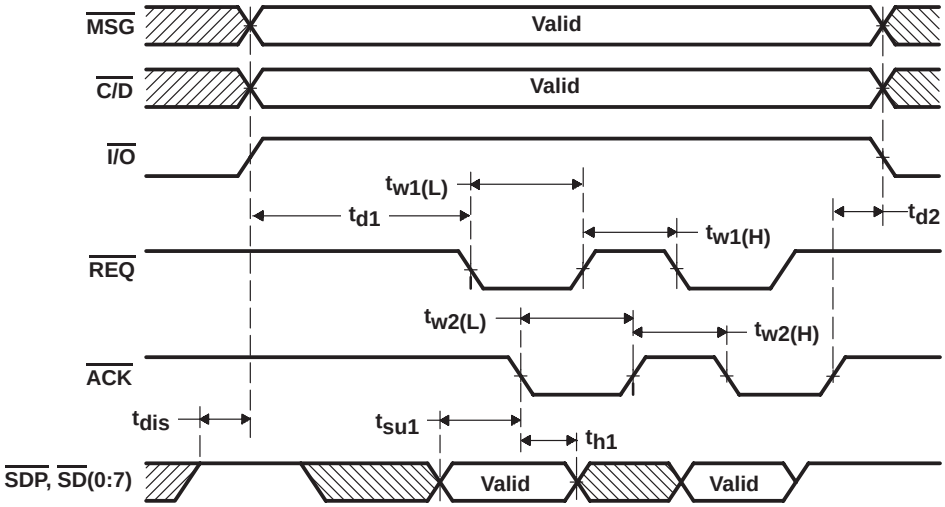


Figure 7–31. SCSI Bus Synchronous Data-Out Transfer (Target) (INITIATOR TO TARGET)

Appendix A
Bus Phases

A.1 SCSI Bus Phases

All access to and communication across the SCSI bus occurs in distinct bus phases. The bus can never be in more than one phase at any given time. Following is a list of the bus phases; subsequent paragraphs give details about each phase.

BUS FREE phase
ARBITRATION phase
SELECTION phase
RESELECTION phase

Information Transfer Phases:
COMMAND OUT phase
STATUS IN phase
DATA OUT phase
DATA IN phase
MESSAGE OUT phase
MESSAGE IN phase
UNSPECIFIED INFO OUT phase
UNSPECIFIED INFO IN phase

A.2 Bus-Free Phase

This phase indicates that no device is actively using the bus. It is indicated by $\overline{\text{BSY}}$ and $\overline{\text{SEL}}$ both being high for a minimum period of time. This phase must be detected before any device can arbitrate for the bus.

A.3 Arbitration Phase

This phase assures that one and only one device gains control of the bus. A device arbitrates by placing its single-bit ID on the SCSI data bus (e.g., device at “address 3” asserts data line 3) and then asserting $\overline{\text{BSY}}$. The arbitrating device must then compare its ID with the IDs of any other arbitrating devices (ID/address 7 has highest priority). The device with highest priority wins the arbitration and asserts $\overline{\text{SEL}}$ while continuing to assert $\overline{\text{BSY}}$. Any device which has lost arbitration must release $\overline{\text{BSY}}$ and wait for another bus-free phase.

A.4 Selection Phase

This phase allows the arbitration winner to set itself up as an initiator and to select another bus device as a target. The arbitration winner places the logical OR of its ID and the desired target ID on the data bus and then releases $\overline{\text{BSY}}$ with $\overline{\text{I/O}}$ high (optionally, it may drive only the ID of the intended target). The arbitration winner (in this case, the initiator) continues to assert the $\overline{\text{SEL}}$ line. The intended target device must recognize its own ID on the data bus and respond by asserting $\overline{\text{BSY}}$. When the initiator sees $\overline{\text{BSY}}$ asserted, it stops asserting $\overline{\text{SEL}}$ and the initiator and target IDs, ending the selection process. The target continues to assert $\overline{\text{BSY}}$ for the duration of the connection. Note that only one target and one initiator can communicate at any given time.

A.5 Reselection Phase

This phase allows the arbitration winner to set itself up as a target and reconnect to a bus device by which it had previously been selected. The arbitration winner places the logical OR of its ID and the desired initiator ID on the data bus and then drives $\overline{\text{BSY}}$ high and $\overline{\text{I/O}}$ low. The arbitration winner (in this case, the target)

continues to assert the SEL line. The intended initiator device must recognize its own ID on the data bus and respond by asserting BSY. When the target sees BSY low, it also asserts BSY and then stops asserting SEL and the initiator and target IDs. When the initiator sees that SEL is no longer being asserted, it stops asserting BSY and the reselection is complete. This process leaves the bus in the same state as the selection phase, with the target driving the BSY signal throughout the connection. Note that only one target and one initiator can communicate at any given time.

A.6 Information Transfer Phases

All information transfers across the SCSI bus occur during bus phases governed by the three target-controlled bus signals I/O, C/D, and MSG. Information is clocked across the bus by the handshake signals REQ and ACK. REQ is driven by the target; the initiator responds with ACK. Information is valid on the rising edge of REQ for a transfer from the target to the initiator, and on the rising edge of ACK for a transfer from the initiator to the target. In this sense, the target is the bus master and the initiator is the slave device.

The bus phases are defined as follows:

MSG	C/D	I/O	BUS PHASE	DIRECTION
L	L	L	Data out	Initiator to target
L	L	H	Data in	Target to initiator
L	H	L	Command out	Initiator to target
L	H	H	Status in	Target to initiator
H	L	L	Unspecified information out	Initiator to target
H	L	H	Unspecified information in	Target to initiator
H	H	L	Message out	Initiator to target
H	H	H	Message in	Target to initiator

Note that the sense of information flow is **out** from the initiator to the target and **in** from the target to the initiator. Since the initiator starts the transfer, it is the reference point, even though the target ultimately controls the information flow.

A.7 SCSI Signal Sources

Certain SCSI bus signals are driven only by the initiator or only by the target. Others are driven by either the initiator or the target depending on the bus phase. The following tables list all the SCSI bus signals (except RST) and their relationship to the bus phases. RST can be driven by any device but is completely asynchronous and is not constrained to any bus phase.

BUS PHASE	SIGNALS AND THEIR DRIVE SOURCES						
	BSY	SEL	SD0-SD7, SDP	I/O	C/D, MSG, REQ	ATN	ACK
Bus free	None	None	None	None	None	None	None
Arbitration	All	Winner	ID bit	None	None	None	None
Selection	Initiator, Target	Initiator	Initiator	None	None	Initiator	None
Reselection	Initiator, Target	Target	Target	Target	None	None	None
Data out	Target	None	Initiator	Target	Target	Initiator	Initiator
Data in	Target	None	Target	Target	Target	Initiator	Initiator
Command out	Target	None	Initiator	Target	Target	Initiator	Initiator
Status in	Target	None	Target	Target	Target	Initiator	Initiator
Message out	Target	None	Initiator	Target	Target	Initiator	Initiator
Message in	Target	None	Target	Target	Target	Initiator	Initiator

Appendix B
Register File Summary

Register #	Register Name	Hex Address	Access
0	Receive FIFO	00	R
0	Transmit FIFO	00	W
1	Command	01	R/W
2	Transfer Status	02	R
3	Bus Phase Status	03	R
4	Functional Interrupt Status	04	R
5	Error Interrupt Status	05	R
6	Interrupt Enable	06	R/W
7	Unused	07	n/a
8	Control	08	R/W
9	Byte Stack Control	09	R/W
10	Parity Control	0A	R/W

Register #	Parity	7	6	5	4	3	2	1	0
0									
0									
1		DMA	M/$\overline{A}$	DDIR	CC4	CC3	CC2	CC1	CC0
2		INT	RFE	RFHF	TFF	TFHF	TC0	OC0	CDACT
3		INIT	TARG	0	ATN	MSG	C/D	I/O	SRST
4		SEL	BUS	ATN	FC	DIS	0	RSL	AB-END
5		PE	UMS	SRST	T-O	NVC	CNTL	NEW-LN	HALT
6		0	0	0	0	0	FCIE	AIE	MIE
7		0	0	0	0	0	0	0	0
8		SE	RE	HA	HPE	AAPE	HD	HAAM	ATN-DS
9		DMD	0	0	0	WL1	WL0	BOF1	BOF0
10		PMPE	MPCE	MPGE	PPCE	PPGE	SPE	SPCE	SPGE

Register #	Register Name					Hex Address		Access
Parity	7	6	5	4	3	2	1	0
11	Synchronous Transfer					0B		R/W
	TP3	TP2	TP1	TP0	OL3	OL2	OL1	OL0
12	Selection/Reselection Time-Out					0C		R/W
	TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0
13	Self ID					0D		R/W
	0	0	0	0	0	ID2	ID1	ID0
14	Destination ID					0E		R/W
	0	0	0	0	0	ID2	ID1	ID0
15	Source ID					0F		R
	0	0	0	0	SIV	ID2	ID1	ID0
16	Target LUN					10		R/W
	1	DSC-PRV	LUN-TAR	Re-served	Re-served	TL2	TL1	TL0
17	Command State					11		R/W
	SDP	0	0	0	CS3	CS2	CS1	CS0
18	Transfer Counter 7-0					12		R/W
								lsb
19	Transfer Counter 15-8					13		R/W
20	Transfer Counter 23-16					14		R/W
	msb							
21	Backup Counter 7-0					15		R
								lsb
22	Backup Counter 15-8					16		R
23	Backup Counter 23-16					17		R
	msb							
24	Offset Counter					18		R/W
	OC7	OC6	OC5	OC4	OC3	OC2	OC1	OC0

Appendix C
Conditions for Invalid Command Interrupt

Occurrence of the invalid command interrupt is driven by different conditions depending on which chip command is executing. The following table lists, by command, the SBC conditions that will cause the device to generate an invalid command interrupt. NOTE: not every incorrect use of a command will cause generation of an invalid command interrupt. See Section 4, Commands, for details on the proper use of each command.

CONDITIONS FOR INVALID COMMAND INTERRUPT	
COMMAND EXECUTING	**INVALID COMMAND INTERRUPT CONDITION(S)**
Chip Reset	Processor interface parity error
Disconnect	Command active
	Interrupt pending
	not TIME-OUT and not in TARGET mode
	Command not active
	Interrupt pending
Assert $\overline{\text{ATN}}$	not in INITIATOR mode
	Multiphase command not active
	Interrupt pending
Negate $\overline{\text{ACK}}$	not in INITIATOR mode
	Processor interface parity error
	Multiphase command active
	Single-phase command not active
	not in MESSAGE-IN phase
Clear Receive FIFO	
Clear Transmit FIFO	
SCSI Bus Reset	Processor interface parity error
Select with $\overline{\text{ATN}}$	Command busy
Select without $\overline{\text{ATN}}$	Interrupt pending
Reselect	in TARGET mode or in INITIATOR mode

CONDITIONS FOR INVALID COMMAND INTERRUPT (CONTINUED)	
COMMAND EXECUTING	**INVALID COMMAND INTERRUPT CONDITION(S)**
Receive Command	Command busy Interrupt pending not in TARGET mode
Receive Data	
Receive Message Out	
Receive Unspecified Information	
Send Status	
Send Data	
Send Message In	
Send Unspecified Information	
Transfer Information	Command busy Interrupt pending not in INITIATOR mode
Transfer Pad	
Select with $\overline{\text{ATN}}$ and Transfer	Command busy
	Interrupt pending
	in TARGET mode
Select without $\overline{\text{ATN}}$ and Transfer	Command Busy
	Interrupt pending
	in TARGET mode or in INITIATOR mode
Reselect and Receive Data	Command busy Interrupt Pending inTARGET mode or in INITIATOR mode
Reselect and Send Data	
Wait for Select with $\overline{\text{ATN}}$	Command busy Interrupt pending
Wait for Select without $\overline{\text{ATN}}$	
Conclude	Command busy Interrupt pending not inTARGET mode
Link to Next Command	

Appendix D
Mechanical Data

NO ART IN THE ORIGINAL INTERLEAF FILE – – – – NEEDS TO BE INSERTED. 9–7–93 fran c

Appendix E
Selected Acronyms

CC: Command Complete

CDB: Command Data Block

DMA: Direct Memory Access

EISR: Error Interrupt Status Register

FISR: Functional Interrupt Status Register

LCC: Linked Command Complete

LCCwF: Linked Command Complete with Flag

LUN: Logical Unit Number

RP: Restore Pointers

SBC: SCSI Bus Controller (SN75C091A)

SCSI: Small Computer Systems Interface

SDP: Save Data Pointer

TC: Transfer Counter

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current and complete.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

Copyright © 1998, Texas Instruments Incorporated