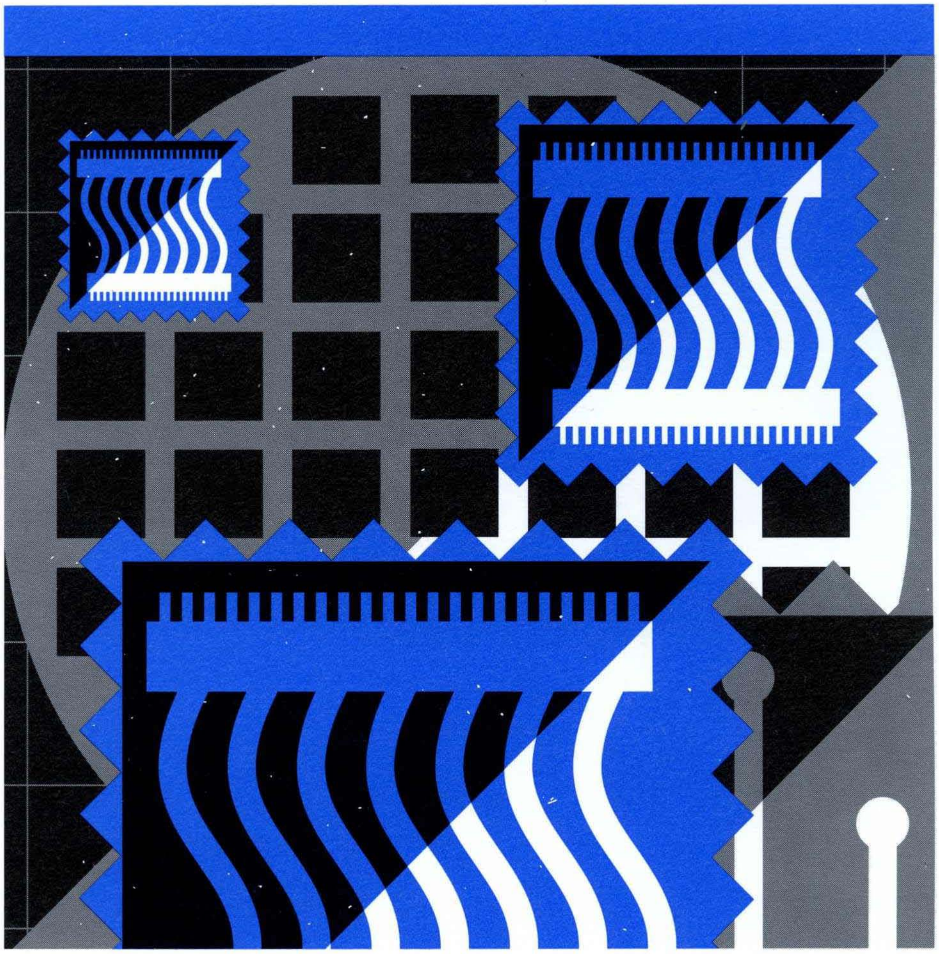
NCR 53C700/53C700-66 SCSI I/O Processor

NCR



Data Manual

INCREASING SCSI RELIABILITY
TolerANT
ACTIVE NEGATION TECHNOLOGY

The product(s) described in this publication is a licensed product of NCR Corporation.

It is the policy of NCR Corporation to improve products as new technology, components, software, and firmware become available. NCR Corporation, therefore, reserves the right to change specifications without notice.

NCR products are not intended for use in life-support appliances, devices, or systems. Use of an NCR product in such applications without the written consent of the appropriate NCR officer is prohibited.

For information on updates to this or other NCR products, contact the NCR Microelectronic Products Division electronic bulletin board at (719) 596-1649.

Preface

SCSI Specifications

This manual assumes some prior knowledge of current and proposed SCSI standards. For background information, please contact:

ANSI
11 West 42nd Street
New York, NY 10036
(212) 642-4900
Ask for document number X3.131-1986 (SCSI-1)

Global Engineering Documents
2805 McGaw
Irvine, CA 92714
(800)-854-7179 or (714) 261-1455
Ask for document number X3.131-199X (SCSI-2)

ENDL Publications
14426 Black Walnut Court
Saratoga, CA 95070
(408) 867-6642
Document names: *SCSI Bench Reference, SCSI Encyclopedia*

Prentice Hall
Englewood Cliffs, NJ 07632
(201) 767-5937
Ask for document number ISBN 0-13-796855-8, *SCSI: Understanding the Small Computer System Interface*

NCR Microelectronic Products Division Electronic Bulletin Board
(719) 596-1649

The SCSI Electronic Bulletin Board
(719) 574-0424

Revision Record

Page No.	Date	Remarks
n/a	10/89	Revision 2.3, Draft
n/a	12/89	Revision 2.4, Draft
n/a	2/90	Revision 2.5, Draft
n/a	1/92	Revision 3.0, Preliminary, Added the NCR 53C700-66 and NCR TolerANT
7-1	2/92	Added Input Leakage-SCSI RST electrical specification
n/a	11/92	Revision 4.0, Final, Reformatted to new standard

Table of Contents

Chapter One
Introduction

Chapter Two
Functional Description

Chapter Three
Signal Descriptions

Chapter Four
Registers

Chapter Five
Command Set

Chapter Six
How to Use the 53C700/53C700-66

Chapter Seven
Electrical Specifications

Appendix A
Register Summary

Appendix B
Mechanical Drawing

Appendix C
SCSI Engineering Notes

Appendix D
53C700 vs 53C700-66 Differences

Chapter One
Introduction

General Description

The NCR 53C700 SCSI I/O Processor (SIOP) is the industry's first intelligent host adapter on a chip. The 53C700 chip is a powerful, next generation solution for high-performance host-to-peripheral interfacing. The 53C700 can be integrated directly on the motherboard or externally adapted to EISA, Micro Channel™, or other buses.

The 53C700 is available as either a 50 MHz chip (labeled and called the 53C700) or a 66 MHz chip (labeled and called the 53C700-66). The 53C700 and the 53C700-66 are both packaged as a 160-pin Quad Flat Pack (QFP). In this manual, when differences exist between the two chips they will be called by their names, but when they are functionally equivalent they will be referred to by SIOP.

SIOP Features Summary

- Supports 25 and 33 MHz 80386 and 80486 memory bus speeds
- Supports 32-bit word data bursts with variable burst lengths
- Unique interrupt status reporting
- High-speed asynchronous/synchronous SCSI bus transfers
- High performance CMOS technology
- Supports variable block size & scatter/gather data transfers
- Minimizes SCSI I/O start latency
- Performs complex bus sequences without interrupts
- Memory transfers up to 47 MB/s for the 53C700 and up to 62 MB/s for the 53C700-66
- Single + 5 V supply
- Active negation of SCSI Data, Parity, REQ and ACK signals with NCR's TolerANT technology improves rise times and Fast SCSI transfer rates in both single-ended and differential modes

NCR TolerANT™ Active Negation Technology

TolerANT is an NCR trade marked term referring to NCR's SCSI transceiver technology. NCR's TolerANT (Active Negation Technology) allows optional active negation of SCSI signals during information transfer phases. More specifically, TolerANT refers to NCR's Active Negation Technology in SCSI drivers and to the conditioning of input signal in SCSI receivers. This feature is only in the 53C700-66 chip. Following is a summary of the TolerANT features and benefits in the 53C700-66 chip.

- Filters high frequency noise on SCSI signal inputs due to bus reflections or voltage transients on VTERM
- Wide hysteresis Schmitt triggers with optimal threshold points
- Fast SCSI-2 performance on both single-ended and differential cables

- Better high level noise margin on SCSI outputs
- Reduced data double clocking problems
- Controlled slew rate and controlled V_{OH}/I_{OH}
- Compatible with ALT-2 termination, i.e. no terminator overload
- Low power dissipation in terminator
- Glitchless SCSI outputs on power up/down
- No current leakage from SCSI bus when powered down
- Active Negation

TolerANT provides improved data integrity in poor cabling environments. NCR SCSI devices are tolerant of cabling environments in which other devices would be subject to data corruption. Active negation is some times called active deassertion.

Active negation is enabled by setting the EAN bit (bit 4) in the CTEST8 register. It can be used in both single-ended and differential mode. TolerANT is compatible with both the Alternative One and Alternative Two termination schemes proposed by the American National Standards Institute.

Active negation causes the SCSI REQ, ACK, data, and parity signals to be actively pulled up to approximately three volts by internal transistors on each pin. The benefits of this technology include reduced noise when the signal is going high (deasserted), increased performance due to balanced duty cycles, and improved Fast SCSI transfer rates.

Benefit Summary

Performance

- Supports 25 and 33 MHz 80386 or 80486 memory bus speeds
- Supports variable block size & scatter/gather data transfers
- Supports 32-bit word data bursts with variable burst lengths
- Minimizes SCSI I/O start latency – Only 500 ns to begin compared to 2- 8 ms
- Performs complex bus sequences without interrupts including restore data pointers
- Unique interrupt status reporting – Reduces ISR overhead
- High-speed asynchronous/synchronous SCSI bus transfers

  53C700/53C700-66 = 5.0 MB/s asynchronous
  53C700 = 6.25 MB/s synchronous
  53C700-66 = 10 MB/s synchronous
- Memory transfers in excess of 50 MB/s

Integration

- Full 32-bit DMA bus master
- High performance SCSI core
- Integrated SCRIPTS processor
- Allows intelligent Host adapter performance on a mother board

Ease of Use

- Reduces SCSI development effort
- Emulates existing intelligent host adapters
- Easily adapted to the SCSI Common Access Method (CAM)
- Preserves existing software
- Development tools and SCSI SCRIPTS provided
- All interrupts are maskable and pollable

Flexibility

- High level programmer's interface (SCSI SCRIPTS)
- Allows tailored SCSI sequences to be executed from main memory

- Flexible sequences to tune I/O performance or to adapt to unique SCSI devices
- Accommodates changes in the logical I/O interface definition
- Low level programmability (register oriented)
- 80286, 80386SX, 80386 or 80486 support
- Externally adaptable to EISA, MCA, and other system buses
- Supports changes from initiator to target roles dynamically

Reliability

- 2 K volts ESD protection SCSI signals
- Typical 350 mV SCSI bus hysteresis
- Protection against bus reflections due to impedance mismatches
- Controlled bus assertion times (reduces RFI, improves reliability, and eases FCC certification)
- Latch-up protection greater than 100 mA
- Voltage feed-through protection (minimum leakage current through SCSI pads)
- 20% of signals are power and ground
- Ground plane isolation of I/O pads and chip logic

Testability

- All SCSI signals accessible through programmed I/O
- SCSI loopback diagnostics
- Self-selection capability
- SCSI bus signal continuity checking

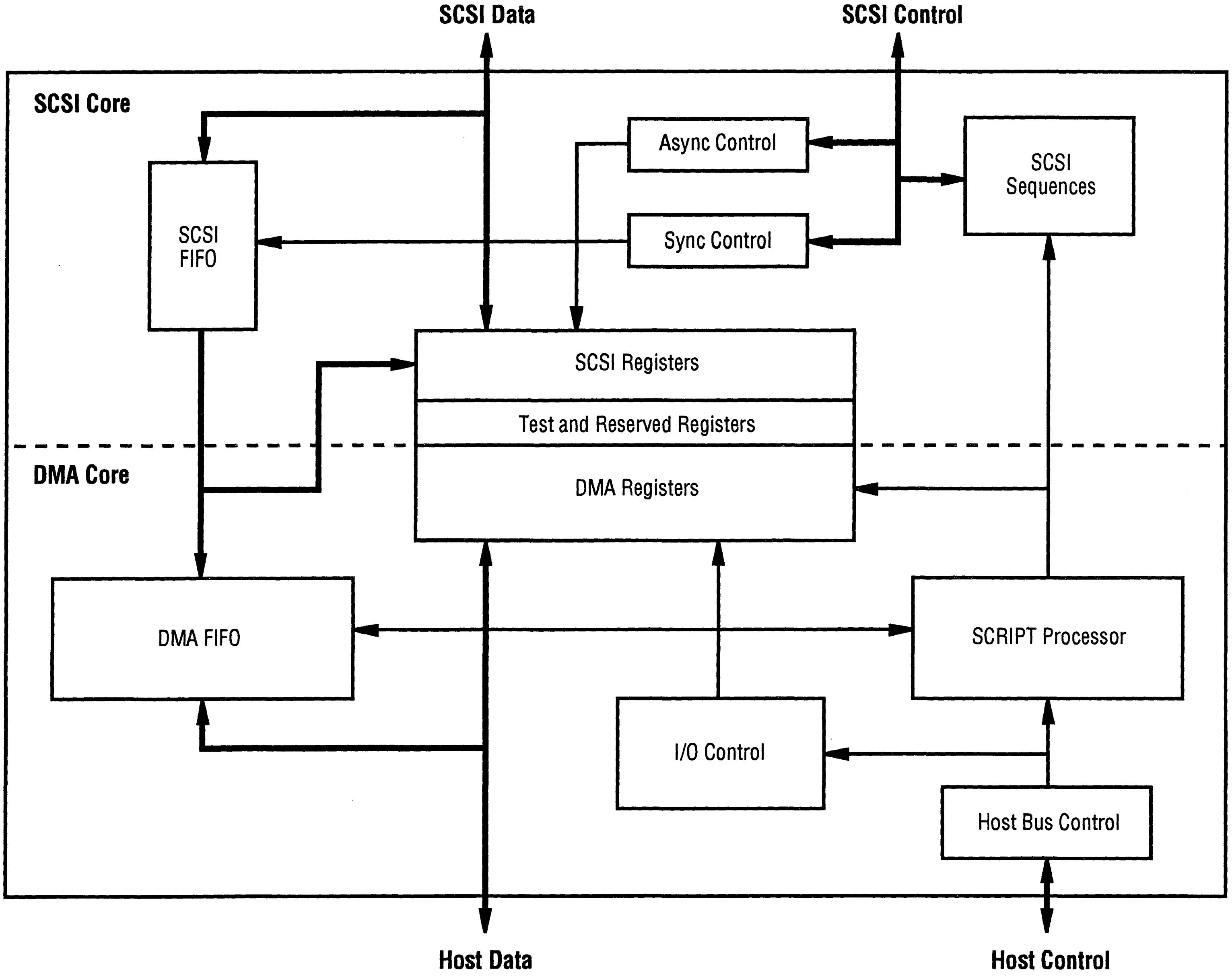
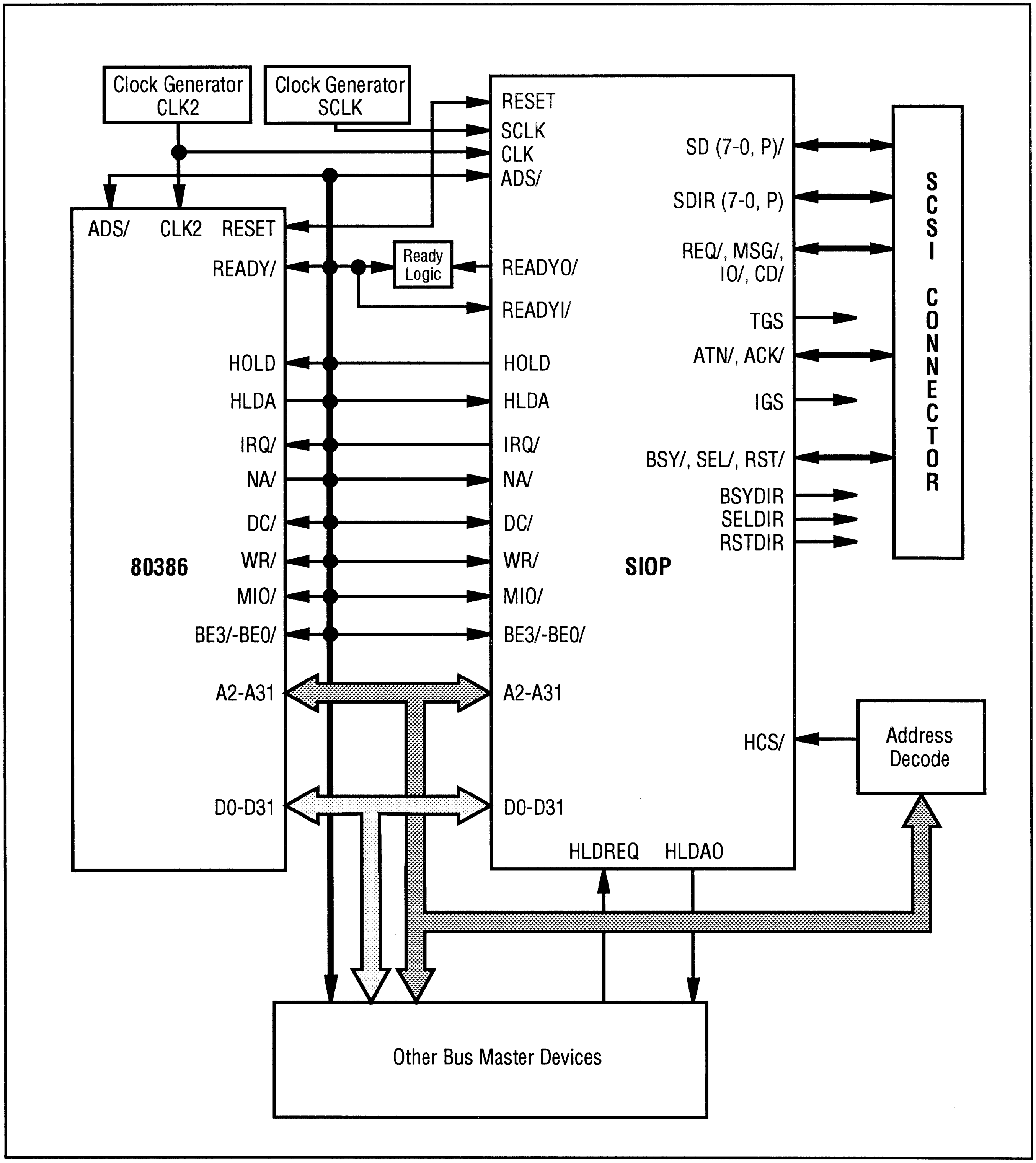


Figure 1-1. NCR 53C700/53C700-66 Block Diagram

Figure 1-2. NCR 53C700/53C700-66 Block Diagram



Chapter Two
Functional Description

The NCR 53C700 SCSI I/O Processor (SIOP) is the first intelligent SCSI Host adapter on a chip. A high-performance SCSI core and an intelligent 32-bit bus master DMA controller have been integrated with a SCSI SCRIPTS Processor™ to accommodate the flexibility requirements of not only SCSI-1, but SCSI-2, and eventually SCSI-3. This flexibility is supported while solving the protocol performance problems that have plagued both intelligent and non-intelligent adapter designs (See the SCSI I/O Processor Block Diagrams in Figures 1-1 and 1-2). The 53C700-66 contains the same structure as the 53C700 plus added features.

SCSI Core

The SCSI core is designed to allow simple migration to SCSI-2 wide bus and enhanced synchronous transfer rate requirements. The 53C700 offers synchronous transfers up to 6.25 MB/s with asynchronous transfers up to 5 MB/s. The 53C700-66 offers synchronous transfers up to 10 MB/s with asynchronous transfers up to 5 MB/s. The programmable SCSI interface makes it easy to “fine tune” the system for specific mass storage requirements or SCSI-2 requirements.

The SCSI core offers low level register access or a high level control interface. Like first generation SCSI devices, the SIOP SCSI core can be accessed as a register oriented device. The ability to sample and/or assert any signal on the SCSI bus can be used in error recovery and diagnostic procedures. Loopback diagnostics are also supported. The SCSI core may perform a self-selection and operate as both an initiator and a target. The 53C700 can test the SCSI pins for physical connection to the board or the SCSI bus.

Unlike previous generation devices, the SCSI core can be controlled by the SCRIPTS Processor through a high level logical interface. Commands controlling the SCSI core are fetched out of the main Host memory. These commands instruct the SCSI core to select, reselect, disconnect, wait for a disconnect, transfer information, change bus phases and in general, implement all aspects of the SCSI protocol.

DMA Core

The DMA core is a bus master DMA device that attaches easily to the 80286, 80386, 80386SX, and 80486 processors. The 53C700 supports 25 MHz 80386/80486 (the 53C700-66 supports 33 MHz 80386/80486) bus timings and may be externally adapted to other system buses such as EISA, Micro Channel or attached through a “bus gasket” to a 680X0 device.

The SIOP supports 16 or 32-bit memory and automatically supports misaligned DMA transfers. As with the 80386/80486, data bus enables are provided for each byte lane. A 32-byte FIFO allows the SIOP to support two, four, or eight (16 or 32-bit) words to be burst across the memory bus interface providing memory transfer rates up to 47 MB/s for the 53C700 and 62 MB/sec for the 53C700-66.

The DMA core is tightly coupled to the SCSI core through the SCRIPTS processor which supports uninterrupted scatter/gather memory operations. A flexible arbitration scheme allows either daisy-chained or "ORed" memory bus request implementations.

SCSI SCRIPTS™ Processor

The SCSI SCRIPTS Processor is a 2 MIPS processor that allows both DMA and SCSI instructions to be fetched from Host memory. Algorithms written in SCSI SCRIPTS can control the actions of the SCSI and DMA cores and are executed from 16 or 32-bit system memory. Complex SCSI bus sequences are executed independently of the Host CPU.

The SCRIPTS Processor can begin a SCSI I/O operation in 500 ns. This compares to 2-8 ms required for traditional intelligent Host adapters. The SCRIPTS Processor offers performance and customized algorithms. Design your own algorithms to tune SCSI bus performance, to adjust to new bus device types (i.e. scanners, communication gateways, etc.), or to incorporate changes in the SCSI-2/3 logical bus definitions without sacrificing I/O performance.

SCSI SCRIPTS are independent of the CPU and system bus in use. Therefore, scripts for an EISA implementation of an 80386 can be identical to the scripts for an 80386SX Micro Channel implementation.

SIOP Data Paths

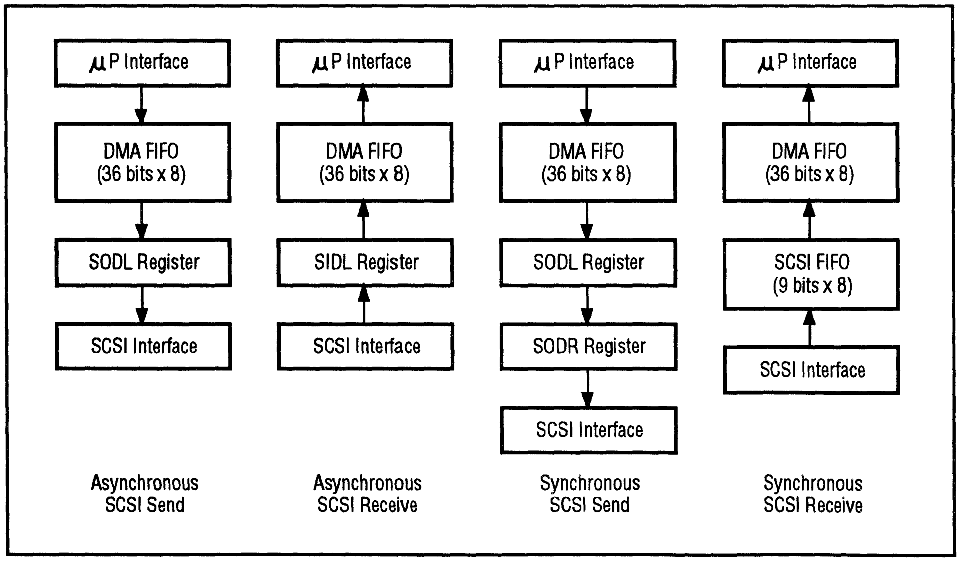
The data paths through the SIOP are dependent on two things.

- Is data being moved in or out of the chip
- Is SCSI data being sent asynchronously or synchronously.

Figure 2-1 shows how data is moved to or from the SCSI bus in each of the four modes.

To determine if any bytes remain in the data path when the chip halts any operation, take the following steps.

Figure 2-1. SIOP Data Paths



- **Asynchronous SCSI Send – Initiator and Target Operation**

1. Use the algorithm described in the DFIFO register description to determine if any bytes are left in the DMA FIFO.
2. Read the SSTAT1 register and examine bit 5 to determine if any bytes are left in the SODL register. If bit 5 equals 1, then there is a byte in the SODL register and add one to the number of bytes left.

- **Synchronous SCSI Send – Initiator and Target Operation**

1. Use the algorithm described in the DFIFO register description to determine if any bytes are left in the DMA FIFO.
2. Read the SSTAT1 register and test bit 5 to determine if any bytes are left in the SODL register. If bit 5 equals 1, then there is one byte in the SODL register, therefore, add one to the number of bytes left.
3. Read the SSTAT1 register and test bit 6 to determine if any bytes are remaining in the SODR register. If bit 6 equals 1, then there is one byte in the SODR register, therefore, add one to the number of bytes left.

- **Asynchronous SCSI Receive – Initiator and Target Operation**

1. Use the algorithm described in the DFIFO register description to determine if any bytes are left in the DMA FIFO.
2. Read the SSTAT1 register and test bit 7 to determine if any bytes are left in the SIDL register. If bit 7 equals 1, then there is one byte in the SIDL register, therefore, add one to the number of bytes left.

- **Synchronous SCSI Receive – Initiator and Target Operation**

1. Use the algorithm described in the DFIFO register description to determine if any bytes are left in the DMA FIFO.
2. Read the SSTAT2 register and test bits 7-4 (the binary representation of the number of valid bytes in the SCSI FIFO) to determine if any bytes are left in the SCSI FIFO.

How the SIOP Transfers 16 or 32-Bit Data

The SIOP can transfer data 16-bits or 32-bits per transfer. The two bits controlling how data is transferred are summarized below. These two bits do not determine how SCSI SCRIPTS are fetched. The Scripts-16 bit in the DCNTL register controls whether SCSI SCRIPTS are loaded 16-bits per transfer or 32-bits per transfer.

Table 2-1. Block Move Transfer Bit Descriptions

BW16	286M	Description
0	0	32-bit data transfers, SIOP asserts and expects 80486/80386 signals
X	1	16-bit data transfers, (slave & master mode) SIOP asserts and expects 80286 signals
1	0	16-bit data transfers, (master mode data transfers only) SIOP asserts and expects 80386 signals

Key: BW16 = Bus Width 16-bit in the DMODE register
286M = 80286 Mode bit in the DMODE register
X = Don't Care

In bus master mode, the SIOP optimizes moving data to an odd-byte boundary address.

16-Bit Data Transfers for Bus Master Read and Write Cycles 80286 Mode or 80386 Mode

The starting address for each Block Move instruction is specified in the second 32-bit word of the instruction. That address is stored in the DNAD register. If starting at an odd address, then the first Block Move operation is a 1-byte transfer on D15-D8, DP1. Each successive transfer is a word transfer to an even address on D15-D0, DP1-DP0. If the SIOP has one byte left to transfer to complete a Block Move instruction, then that byte will be transferred to an even address on D7-D0, DP0.

80286 Mode 16-Bit Data Transfers

Table 2-2. 80286 Mode 16-Bit Data Transfers (DMODE register, bit 4 = 1)

Address (DNAD register)	**BHE/**	**A0**	**D15-8**	**D7-0**
Even byte transfers (address + 0)	1	0	-	xx
Odd byte transfers (address + 1)	0	1	yy	-
Even word transfers (address + 0)	0	0	yy	xx

Table 2-3. Address, Data, and Byte Enables for 16-Bit Bus

yy	xx	
15—8	7—0	Address (longword aligned)
BHE/	A0	

80386 Mode 16-Bit Data Transfers

Table 2-4. 80486/80386 Mode 16-Bit Data Transfers (DMODE register bit 4 = 0, bit 5 = 1)

	First Cycle						Second Cycle					
Address (DNAD register)	BE3/	BE2/	BE1/	BE0/	D15-8	D7-0	BE3/	BE2/	BE1/	BE0/	D15-D8	D7-D0
8-Bit Transfers												
Address + 0	1	1	1	0	-	ww	-	none	-	-	-	-
Address + 1	1	1	0	1	xx	-	-	none	-	-	-	-
Address + 2	1	0	1	1	-	yy	-	none	-	-	-	-
Address + 3	0	1	1	1	zz	-	-	none	-	-	-	-
16-Bit Transfers												
Address + 0	1	1	0	0	xx	ww	-	none	-	-	-	-
Address + 1	1	0	0	1	xx	-	1	0	1	1	-	yy
Address + 2	0	0	1	1	zz	yy	-	none	-	-	-	-
24-Bit Transfers												
Address + 0	1	0	0	0	xx	ww	1	0	1	1	-	yy
Address + 1	0	0	0	1	xx	-	0	0	1	1	zz	yy
32-Bit Transfers												
Address + 0	0	0	0	0	xx	ww	0	0	1	1	zz	yy

Table 2-5. Address, Data Bus and Byte Enables for 32-Bit Bus

zz	yy	xx	ww	
31—24	23—16	15—8	7—0	Address (longword aligned)
BE3/	BE2/	BE1/	BE0/	

Note: 80386 mode 16-bit data transfers are implemented the same as Intel's 80386 microprocessor.

32-Bit Data Transfers

The starting address for each Block Move instruction is specified in the second 32-bit word of the instruction stored in the DNAD register.

If bit 0 = 1, and the Byte Counter Value stored in the DBC register is greater than three, then the first Block Move operation involves a 3-byte transfer on D31-D8, DP3-DP1 with BE3/, BE2/, and BE1/ all driven active. Each successive transfer will occur on D31-D0, DP3-DP0 with BE3/-BE0/ all driven active.

If the SIOP has two-bytes to transfer in order to complete a Block Move instruction, then those bytes will be transferred on D15-D0, DP1-DP0 with BE1/ and BE0/ driven active.

If the SIOP has one-byte to transfer in order to complete a Block Move instruction, then that byte will be transferred on D7-D0, DP0 with BE0/ driven active.

If the SIOP has three-bytes to transfer to complete a Block Move instruction, those bytes will be transferred on D23-D0, DP2-DP0 with BE2/, BE1/, and BE0/ driven active.

80486/80386 Mode

Table 2-6. 32-Bit Data Transfers in 80486/80386 Mode

	BE 3/	BE 2/	BE 1/	BE 0/
Address + 0	0	0	0	0
Address + 1	0	0	0	1
Address + 2	0	0	1	1
Address + 3	0	1	1	1

Key: BE3/-BE0/ 0 = asserted 1 = deasserted

Address longword boundary aligned

How the SIOP Fetches Instructions

Write the address containing the first SCSI SCRIPTS to the DSP register to start the SIOP instruction fetch process. After the first SCSI SCRIPTS address is written to the DSP register, the SIOP continues to fetch and execute instructions by reading them from system memory.

SCSI SCRIPTS are not required to reside in system memory because by decoding a certain address space, they could reside in a PROM. Only store SCSI SCRIPTS in a memory-mapped address because the SIOP does not fetch instructions out of I/O-mapped address space. Each SCSI SCRIPTS instruction consists of two 32-bit words. Load SCSI SCRIPTS instructions in one of two ways: by fetching two 32-bit words or by fetching four 16-bit words.

If SCSI SCRIPTS are to be loaded 16-bits per transfer, write the "SCRIPTS Loaded in 16-Bit Mode" bit in the DCNTL register to 1. The DC/ control signal can be driven high or low depending on the status of the "DC/ Output Low for Instruction Fetch" bit in the CTEST7 register. If this bit is 1, then the DC/ signal will be low during instruction fetch cycles. Allowing the DC/ signal to be driven low during instruction fetches allows the system designer to choose whether SCSI SCRIPTS instructions should reside in cacheable or alternate memory space.

If this bit is 0, then the DC/ signal will be high during instruction fetch cycles. Usually only control (DC/ low) resides in the cache memory and data information (DC/ high) does not.

How To Transfer Data as a Bus Master

When the SIOP becomes bus master, it takes over control of the system bus and can transfer data in a variety of ways. The SIOP can transfer data to I/O addresses, memory addresses, or a fixed address. The following three bits determine the width and type of data transfer that will occur once the SIOP assumes bus mastership and is ready to transfer data.

Table 2-7. SIOP Bus Master Data Transfer Descriptions

BW16	IOM	FAM	Transfer Description
0	0	0	32-bit transfers to a memory address which is incremented after each transfer
0	0	1	32-bit transfers to a memory address which is not incremented after each transfer
0	1	0	32-bit transfers to an I/O address which is incremented after each transfer
0	1	1	32-bit transfers to an I/Oaddress which is not incremented after each transfer
1	0	0	16-bit transfers to a memory address which is incremented after each transfer
1	0	1	16-bit transfers to a memory address which is not incremented after each transfer
1	1	0	16-bit transfers to an I/O address which is incremented after each transfer
1	1	1	16-bit transfers to an I/O address which is not incremented after each transfer

Key:	BW16	= Bus Width 16-bit in the DMODE register
	IOM	= I/O or Memory Mapped bit in the DMODE register
	FAM	= Fixed Address Mode bit in the DMODE register

Interrupts

Polling vs. Hardware Interrupts

The external microprocessor can be informed of an interrupt condition by polling or hardware interrupts. Polling means that the microprocessor must continually loop and read a register until it detects a bit set that indicates an interrupt. This method is the fastest, but it wastes CPU time that could be used by other system tasks. The preferred method of detecting interrupts in most systems is hardware interrupts. In this case, the 53C7X0 will assert the Interrupt Request (IRQ/) line that will interrupt the microprocessor, causing the microprocessor to execute an interrupt service routine. A hybrid approach would use hardware for long waits, and use polling for short waits.

Registers

The five registers in the 53C7X0 that are used for detecting or defining interrupts are the ISTAT (register 0x21), the SSTAT0 (register 0x0D), the DSTAT (register 0x0C), the SIEN (register 0x03), and the DIEN (register 0x39).

The ISTAT is the only register than can be accessed as a slave during SCRIPTS operation, therefore it is the register that is polled when polled interrupts are used. It is also the first register that should be read when the IRQ/ pin has been asserted in association with a hardware interrupt. If the SIP bit in the ISTAT register is set, then a SCSI-type interrupt has occurred and the SSTAT0 register should be read. If the DIP bit in the ISTAT register is set, then a DMA-type interrupt has occurred and the DSTAT register should be read. SCSI-type and DMA-type interrupts may occur simultaneously, so in some cases both SIP and DIP may be set.

The SSTAT0 register contains the SCSI-type interrupt bits. Reading this register will determine which condition or conditions caused the SCSI-type interrupt, and will clear that SCSI interrupt condition. If the 53C7X0 is receiving data from the SCSI bus and a fatal interrupt condition occurs, the 53C7X0 will attempt to send the contents of the DMA FIFO to memory before generating the interrupt. If the 53C7X0 is sending data to the SCSI bus and a fatal SCSI interrupt condition occurs, data could be left in the DMA FIFO. Because of this the DFE bit in DSTAT should be checked. If this bit is clear, set the CLF (Clear DMA and SCSI FIFOs) bit before continuing.

The DSTAT register contains the DMA-type interrupt bits. Reading this register will determine which condition or conditions caused the DMA-type interrupt, and will clear that DMA interrupt condition. Bit 7 in DSTAT, DFE (DMA FIFO Empty), is purely a status bit; it will not generate an interrupt under any circumstances and will not be cleared when read. DMA interrupts will flush neither the DMA nor SCSI FIFOs before generating the interrupt, so the DFE bit in the DSTAT register should be checked after any DMA interrupt. If the DFE bit is clear, then the FIFOs must be cleared by setting the CLF (Clear DMA and SCSI FIFOs) bit, or flushed by setting the FLF (Flush DMA FIFO) bit. The CLF bit is bit 6 in the DFIFO register on the 53C700 and 53C700-66, bit 2 in CTEST8 in the 53C710, and bit 2 in CTEST3 in the 53C720. The FLF bit is bit 7 in the DFIFO register in the 53C700 and 53C700-66, bit 3 in CTEST8 in the 53C710, and bit 3 in CTEST3 in the 53C720.

The SIEN register is the interrupt enable register for the SCSI interrupts in SSTAT0.

The DIEN register is the interrupt enable register for DMA interrupts in DSTAT.

Fatal vs. Non-Fatal Interrupts

A fatal interrupt, as the name implies, always causes SCRIPTS to stop running. A non-fatal interrupt will cause SCRIPTS to stop running only if it is not masked. Masking will be discussed later in this engineering note. All DMA interrupts (indicated by the DIP bit in ISTAT and one or more bits in DSTAT being set) are fatal.

Some SCSI interrupts (indicated by the SIP bit in the ISTAT and one or more bits in SSTAT0 being set) are non-fatal. When the chip is operating in Initiator mode, only the CMP (Function Complete) and SEL (Selected or Reselected) interrupts are non-fatal. When operating in Target mode CMP, SEL, and M/A (Target mode: ATN/ active) are non-fatal. Refer to the description for the DHP (Disable Halt on a Parity Error or ATN/ active (Target Mode Only)) bit in the SXFER register to configure the chip's behavior when the ATN/ interrupt is enabled during Target mode operation.

The reason for non-fatal interrupts is to prevent SCRIPTS from stopping when an interrupt occurs that does not require service from the CPU. This prevents an interrupt when arbitration is complete (CMP set), when the 53C7X0 has been selected or reselected (SEL set), or when the initiator has asserted ATN (target mode: ATN/ active). These interrupts are not needed for events that occur during high-level SCRIPTS operation.

Masking

Masking an interrupt means disabling or ignoring that interrupt. Interrupts can be masked by clearing bits in the SIEN (for SCSI interrupts) register or DIEN (for DMA interrupts) register. How the chip will respond to masked interrupts depends on: whether polling or hardware interrupts are being used; whether the interrupt is fatal or non-fatal; and whether the chip is operating in Initiator or Target mode.

If a non-fatal interrupt is masked and that condition occurs, SCRIPTS will not stop, the appropriate bit in the SSTAT0 will still be set, the SIP bit in the ISTAT will not be set, and the IRQ/ pin will not be asserted. See the section on non-fatal vs. fatal interrupts for a list of the non-fatal interrupts.

If a fatal interrupt is masked and that condition occurs, then SCRIPTS will still stop, the appropriate bit in the DSTAT or SSTAT0 register will be set, the SIP or DIP bits in the ISTAT will be set, and the IRQ/ pin will not be asserted.

When the chip is initialized, enable all fatal interrupts if you are using hardware interrupts. If a fatal interrupt is disabled and that interrupt condition occurs, SCRIPTS will halt and the system will never know it unless it times out and checks the ISTAT after a certain period of inactivity.

If you are polling the ISTAT instead of using hardware interrupts, then masking a fatal interrupt will make no difference since the SIP and DIP bits in the ISTAT inform the system of interrupts, not the IRQ/ pin.

Masking an interrupt after IRQ/ is asserted will not cause IRQ/ to be deasserted.

Stacked Interrupts

The 53C7X0 has the ability to stack interrupts if they occur one after the other. If the SIP or DIP bits in the ISTAT register are set (first level), then there is already at least one pending interrupt and any future interrupts will be stacked in extra registers behind the SSTAT0 and DSTAT registers (second level). When two interrupts have occurred and the two levels of the stack are full, any further interrupts will set additional bits in the extra registers behind SSTAT0 and DSTAT. When the first level of interrupts are cleared, all the interrupts that came in afterward will move into the SSTAT0 and DSTAT. After the first interrupt is cleared by reading the appropriate register, the IRQ/ pin will be deasserted for a set time as published in the product Data Manual; the stacked interrupt(s) will move into the SSTAT0 or DSTAT; and the IRQ/ pin will be asserted once again.

Since a masked non-fatal interrupt will not set the SIP or DIP bits, interrupt stacking will not occur as a result of a masked, non-fatal interrupt. A masked, non-fatal interrupt will still post the interrupt in SSTAT0 but will not assert the IRQ/ pin. Since no interrupt is generated, future interrupts will move right into the SSTAT0 instead of being stacked behind another interrupt. When another condition occurs that generates an interrupt, the bit corresponding to the earlier masked non-fatal interrupt will still be set.

A related situation to interrupt stacking is when two interrupts occur simultaneously. Since stacking does not occur until the SIP or DIP bits are set, there is a small timing window in which multiple interrupts can occur but will not be stacked. These could be multiple SCSI interrupts (SIP set), multiple DMA interrupts (DIP set), or multiple SCSI and multiple DMA interrupts (both SIP and DIP set).

As previously mentioned, DMA interrupts will not attempt to flush the FIFOs before generating the interrupt. It is important to set either the CLF (Clear DMA and SCSI FIFOs) bit or the FLF (Flush DMA FIFO) bit if a DMA interrupt occurs and the DFE (DMA FIFO Empty) bit is not set. This is because any future SCSI interrupts will not be posted until the DMA FIFO is clear of data. These 'locked out' SCSI interrupts will be posted as soon as the DMA FIFO is empty.

Halting in an Orderly Fashion

When an interrupt occurs, the 53C7X0 will attempt to halt in an orderly fashion.

- If in the middle of an instruction fetch, the fetch will be completed, except in the case of a Bus Fault or Watchdog Timeout. Execution will not begin, but the DSP will point to the next instruction since it is updated when the current Script is fetched.
- If the DMA direction is a write to memory and a SCSI interrupt occurs, the 53C7X0 will attempt to flush the DMA FIFO to memory before halting. Under any other circumstances only the current cycle will be completed before halting, so the DFE bit in DSTAT should be checked to see if any data remains in the DMA FIFO.
- SCSI REQ/ACK handshakes that have begun will be completed before halting.
- The 53C7X0 will attempt to clean up any outstanding synchronous offset before halting.
- In the case of Transfer Control Instructions, once instruction execution begins it will continue to completion before halting.
- If the instruction is a JUMP/CALL WHEN <phase>, the DSP will be updated to the transfer address before halting.
- All other instructions may halt before completion.

Sample Interrupt Service Routine

The following is a sample of an interrupt service routine for the 53C7X0. It can be repeated if polling is used, or should be called when the IRQ/ pin is asserted if hardware interrupts are used.

1. Read ISTAT
2. If only the SIP bit is set, read SSTAT0 to clear the SCSI interrupt condition and get the SCSI interrupt status. The bits in the SSTAT0 tell which SCSI interrupt(s) occurred and determine what action is required to service the interrupt(s).
3. If only the DIP bit is set, read the DSTAT to clear the interrupt condition and get the DMA interrupt status. The bits in the DSTAT will tell which DMA interrupt(s) occurred and determine what action is required to service the interrupt(s).
4. If both the SIP and DIP bits are set, read SSTAT0 and DSTAT as a word to clear the SCSI and DMA interrupt condition and get the interrupt status. If using 8-bit reads of the SSTAT0 and DSTAT registers to clear interrupts, insert 10 CLKs on the 53C700 and 53C700-66, and 12 CLKs on the 53C710 between the consecutive reads to ensure that the interrupts clear properly. Both the SCSI and DMA interrupt conditions should be handled before leaving the ISR. It is recom-

mended that the DMA interrupt be serviced before the SCSI interrupt because a serious DMA interrupt condition could influence how the SCSI interrupt is acted upon.

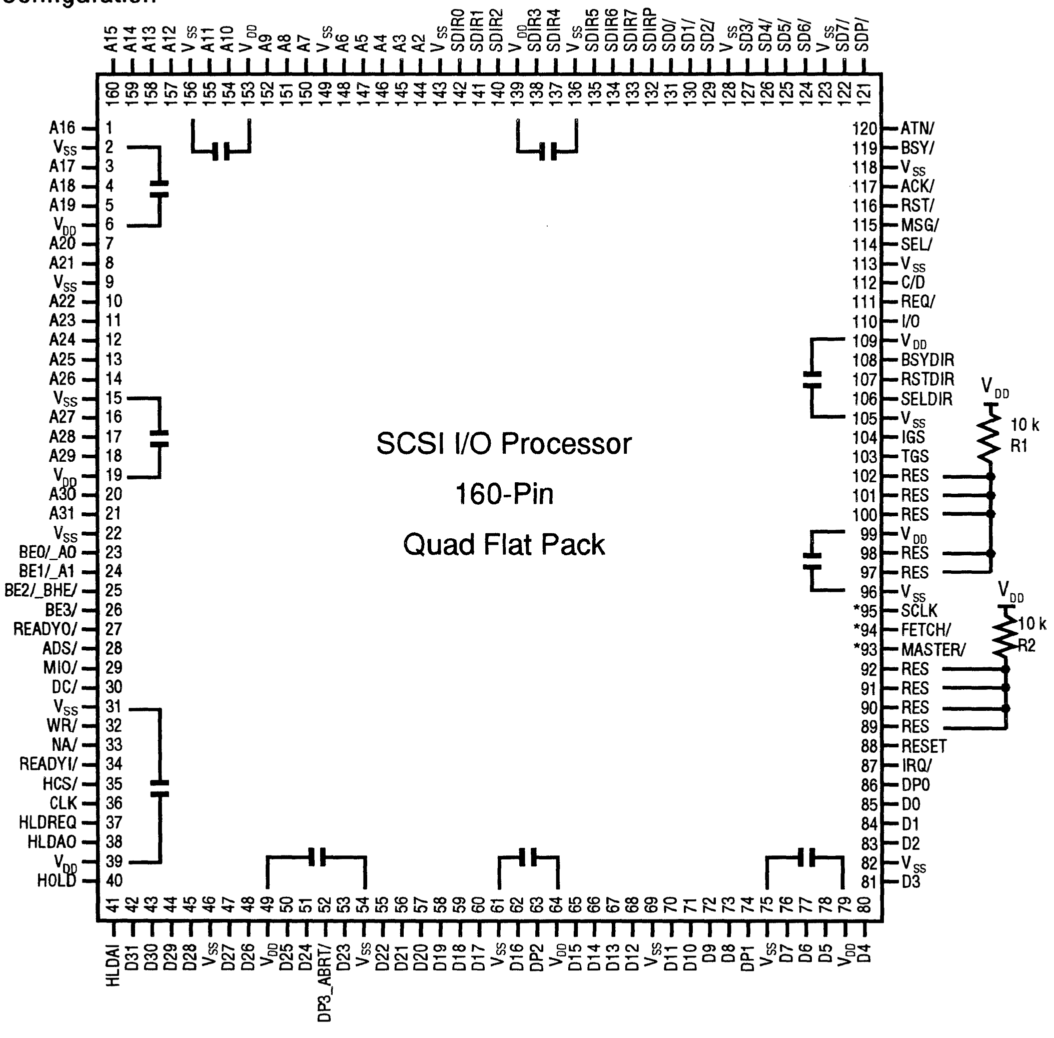
5. When using polled interrupts, go back to step 1 before leaving the interrupt service routine, in case any stacked interrupts moved in when the first interrupt was cleared. When using hardware interrupts, the IRQ/ pin will be asserted again if there are any stacked interrupts. This should cause the system to re-enter the interrupt service routine.

Chapter Three

Signal Descriptions

There are two groups of signals: Host interface signals and SCSI interface signals. The NCR 53C700 SIOP will be referred to as the SIOP throughout this document. A slash ("/") indicates an active-low signal. An underscore ("_") indicates a dual-function pin. An ("I") for Input, an ("O") for Output, and ("I/O") for Input and Output signal. Note: All unused inputs or bi-directional signals that could become an input should be tied high or low but must never be left floating. This applies to any CMOS device.

Figure 3-1. Pin Configuration



* If unused, pins should be connected to a 10 K pullup resistor

Table 3-1. Host Interface Signals

Symbol	Pin #	Type	Description
A(31-2)	21-20, 18-16, 14-10, 8-7, 5-3, 160-157, 155-154, 152-150, 148-144	I/O	*Host Address Bus* – Tri-state, active-high. These signals provide physical memory addresses or I/O port addresses dependent on the MIO/ signal state. The 80386/80486 drives the address lines in slave mode and the SIOP drives the address lines in master mode. In 80286 mode, the A(31-24) signals are driven to the value in the DNAD register to be used for the upper address signals for memory paging. The address outputs are synchronized with the CLK signal. During data transfer in slave mode to/from the SIOP's registers, use A(5-2) in 80386/ 80486 mode, and A(5-0) in 80286 mode.
D(31-0)	42-45, 47-48, 50-51, 53, 55-60, 62, 65-68, 70-73, 76-78, 80-81, 83-85	I/O	*Host Data Bus* – Tri-state, active-high. These signals provide the data path between the 80486/80386/80286 and the SIOP. Data can be transferred on 32-bit or 16-bit buses. This is controlled by bits 4 and 5 in the DMODE register. The DMODE register should not be changed while DMA is active. Data outputs are synchronized with the CLK signal.
DP3_ABRT/	52	I/O	*Host Data Parity for byte lane 3 or Abort* – Tri-state, active-low. This signal can be used as the parity bit for D31-D24. When not using the pass parity option, use this signal as an Abort Transfer pin. To enable this signal as an ABRT/ signal, parity generation must be enabled (bit 2, the SCNTL0 register). When ABRT/ is asserted, the DMA transfer is aborted. The ABRT input is asynchronous and requires a 40 ns pulse width. When this signal is a parity signal, its output is synchronized with the CLK signal.
DP(2-0)	63, 74, 86	I/O	*Host Data Parity Bits* – Tri-state, active-high. These signals are the parity bits for the Host data bus, D(23-0). DP2 is the parity signal for data bits D(23-16), DP1 is the parity signal for data bits D(15-8), and DP0 is the parity signal for data bits D(7-0). The SIOP supports both even and odd parity on the Host data bus. Host bus even/odd parity is programmed through bit 2 of the CTEST7 register. When the SIOP is configured for parity generation (bit 2, the SCNTLO register), these signals are placed in a high-impedance state. Parity outputs are synchronized with the CLK signal.

Table 3-1. Host Interface Signals (Continued)

Symbol	Pin #	Type	Description
BE3/	26	I/O	*Byte Enable, BE3/* – Tri-state, active-low. This signal enables data transfer in the D(31-24) data byte lane. It is driven by the Host when the SIOP is in slave mode and driven by the SIOP during a DMA transfer. In 286 mode, this signal is not defined and should be pulled high. Bit 4 of the DMODE register determines whether the chip operates in 286 mode or 386/486 mode. The output is synchronized with the CLK signal.
BE2/_BHE/	25	I/O	*Byte Enable, BE2/ or Byte High Enable/* – Tri-state, active-low. In 386/486 mode, this signal enables data transfers on the D(23-16) data byte lane. In 286 mode, it distinguishes between 8-bit and 16-bit data transfers. Bit 4 of the DMODE register determines whether the chip operates in 286 or 386 mode. The output is synchronized with the CLK signal.
BE1/_A1	24	I/O	*Byte Enable, BE1/ or Address, A1* – Tri-state, active-low. In 386/486 mode, this signal enables data transfer in the D(15-8) data byte lane. In 286 mode, this pin is active-high, address line A1, and should be connected to A1 of the 80286. Bit 4 of the DMODE register determines whether the chip operates in 286 or 386/486 mode. The output is synchronized with the CLK signal.
BE0/_A0	23	I/O	*Byte Enable, BE0/ or Address, A0* – Tri-state, active-low. In 386 mode, this signal enables data transfers on the D(7-0) data byte lane. In 286 mode, this signal is tri-state, active-high and used as address line A0, connected to the A0 pin of the 286. Bit 4 of the DMODE register determines whether the chip is operating in 286 or 386 mode. The output is synchronized with the CLK signal.
W_R/	32	I/O	*Write or Read/* – Tri-state. This signal defines the type of bus cycle being performed. When the SIOP is in slave mode, high is a write to the chip and low is a read from the chip. When the SIOP is in master mode, high is a write to the system bus and low is a read from the system bus. The output is synchronized with the CLK signal.

Table 3-1. Host Interface Signals (Continued)

Symbol	Pin #	Type	Description
DC/	30	O	*Data Control Output* – Tri-state. This signal defines the type of bus cycle being performed. A high signal indicates data is on the bus. A low signal indicates that the bus contains control information. In master mode this signal can be driven to either state when the SIOP is performing an instruction fetch operation. The assertion or deassertion of this signal during instruction fetch operations is controlled through bit 1 in the CTEST7 register. The output is synchronized with the CLK signal.
MIO/	29	O	*Memory Input/Output* – Tri-state. This signal defines the type of bus cycle being performed. When high, transfer is to a memory address. When low, transfer is to an I/O address. The output is synchronized with the CLK signal.
ADS/	28	I/O	*Address Status* – Tri-state, active-low. ADS/ indicates that address and control signals are valid and stable. In slave mode, this signal is driven by the 286/386/486. In master mode, it is driven by the SIOP. The output is synchronized with the CLK signal.
NA/	33	I	*Next Address Request* – Active-low. When the SIOP is in master mode, NA/ indicates that the system is requesting address pipelining. During address pipelining, address and status signals for the next bus cycles are driven during the current cycle. An active signal indicates that the system is ready to accept new values of BE3/, BE2/, _BHE/, BE1/_A1, BE0/_A0, A31-A2, WR/, DC/, and MIO/. It is monitored only when in master mode. This signal should not be driven active during a slave access to the SIOP.
READYI/	34	I	*Ready Transfer Acknowledge* – Active-low. In master mode, READYI/ indicates that the slave device is ready to transfer data. When READYI/ is active during a read cycle, the SIOP latches the input data and terminates the cycle. If READYI/ is active during a write cycle, the SIOP terminates the bus cycle. In slave mode, when data is read from the SIOP, this signal is monitored by the SIOP to determine when to stop driving the data bus. This allows wait states for both read's and writes in slave mode to be inserted to extend the bus cycle if needed. The 53C700 always needs a READYI/ to terminate a read or write signal.

Table 3-1. Host Interface Signals (Continued)

Symbol	Pin #	Type	Description
READYO/	27	O	*Ready Output Signal* – Active-low, totem-pole. When the SIOP is in slave mode, it asserts READYO/ to acknowledge the completion of a bus cycle. A 53C700 slave read or write cycle is a minimum of 5t states or 10 clock periods. It is not used in master mode. The output is synchronized with the CLK signal.
HOLD	40	O	*Hold Request Output* – Active-high, totem-pole. This output only signal is asserted when the SIOP needs access to the host system bus while performing a DMA transfer. If the HLDREQ input signal is asserted, this signal is asserted to allow another bus master device to gain control of the system bus using a daisy-chaining* scheme. The output is synchronized with the CLK signal.
HLDREQ	37	I	*Bus Hold Request Input* – Active-high. This signal indicates that another bus master device requests use of the host bus. It allows the system to incorporate a daisy-chaining* technique for handling system bus requests for use. If another bus master device requests the bus at the same time as the SIOP, the SIOP has priority. The signal can be asserted asynchronously by another bus master device.
HLDAI	41	I	*Bus Hold Acknowledge In* – Active-high. This signal is asserted by the Host CPU to indicate that it has given up the system bus. This signal is passed through to the HLDAO pin unless the SIOP requires use of the bus.
HLDAO	38	O	*Bus Hold Acknowledge Out* – Active-high, totem-pole. This signal is a copy of HLDAI, unless the SIOP assumes bus mastership and uses the system bus. If HLDAI is active and the SIOP does not need to use the system bus, then this signal is asserted. The output is synchronized with the CLK signal.

* See 80386/80486 Interface in Section 6 for more information on HOLD/HLDA schemes.

Table 3-1. Host Interface Signals (Continued)

Symbol	Pin #	Type	Description
RESET	88	I	*Hardware Reset* – Active-high. When asserted, all registers are set to the default values as described in the register sections. The signal is connected to the 80386 RESET line. This signal also defines the Ø1 and Ø2 clock phases.
IRQ/	87	O	*Interrupt Request* – Active-low, open drain. This signal is asserted in response to an interrupt condition or when a SCSI SCRIPTS interrupt instruction is issued. This signal has an internal pull-up resistor. The output is synchronized with the CLK signal.
HCS/	35	I	*Host Chip Select* – Active-low. This signal is generated by external address decoding to allow the SIOP's registers to be memory or I/O mapped.
CLK	36	I	*Square Wave Clock* – provides the fundamental timing for the system bus and for the SIOP chip. It should be the same signal as the CLK2 input of the 80386. The CLK signal input frequency should range from 16.67 MHz to 50 MHz for the 53C700 and up to 66 MHz for the 53C700-66 with a 40% to 60% duty cycle.
MASTER/*	93*	O	*Master Status* – 8 mA, tristate, output. Driven low when the 53C700-66 becomes bus master. This feature is enabled by bit 6 of CTEST8.
FETCH/*	94*	O	*Fetch Opcode* – 8 mA, tristate, output. Indicates that the next bus request will be for an opcode fetch. This feature is enabled by bit 6 of CTEST8.
SCLK*	95*	I	*SCSI Clock* – SCLK may be used to derive all SCSI related timings. Normal operation is to derive all SCSI timings from CLK (pin 36). The speed of this clock will be determined by the application's requirements. This feature is enabled by bit 7 of CTEST8.

* Pins 93, 94, and 95 are 53C700-66 features. Any pin(s) not used or in the 53C700, connect them to a 10 K pull up resistor.

Table 3-2. SCSI Interface Signals

Symbol	Pin #	Type	Description
SD(7-0)/, SDP/	122, 124-127 129-131, 121	I/O	*SCSI Data/* and *SCSI Data Parity/* – 48 mA, open drain, active-low.
SDIR(7-0)	133-135, 137-138 140-142	O	*SCSI Data Direction* – Active-high. In differential mode, these signals control the direction of external differential pair transceivers for the SD(7-0)/ signals. When this signal is high, the direction is from the SIOP to the SCSI bus. When it is low, the direction is from the SCSI bus to the SIOP. These signals are always valid, even in single-ended mode.
SDIRP	132	O	*SCSI Parity Direction* – Active-high. In differential mode this signal controls the direction of an external differential pair transceiver for the SDP/ signal. When the signal is high signals move from the SIOP to the SCSI bus. When it is low, signals move from the SCSI bus to the SIOP. The signal is always valid, even in single-ended mode.
ATN/	120	I/O	*SCSI Attention* – 48 mA, open drain. The initiator asserts this signal to indicate to the target that a message out phase is desired. The signal can be directly connected to the single-ended SCSI ATN line. In differential mode, the IGS output controls the direction of this signal.
REQ/	111	I/O	*SCSI Data Transfer Request* – 48 mA, open drain, active-low. This signal is asserted by the target requesting a data transfer. When this signal is active, the MSG/, C/D, and I/O phase lines are valid. This signal can be directly connected to the single-ended SCSI REQ line. In differential mode, the TGS output controls the direction of this signal.
ACK/	117	I/O	*SCSI Acknowledge* – 48 mA open drain, active-low. This signal is asserted by the initiator in response to the REQ/ signal to acknowledge a data transfer. It can be directly connected to the single-ended SCSI ACK line. In differential mode, the IGS output controls the direction of this signal.

Table 3-2. SCSI Interface Signals (Continued)

Symbol	Pin #	Type	Description
MSG/	115	I/O	*SCSI Message Phase Signal* – 48 mA, open drain. The target asserts this signal with the I/O and C/D signals to determine the information transfer phase. This signal can be directly connected to the single-ended SCSI MSG line. In differential mode, the TGS output controls the direction of this signal.
I/O	110	I/O	*SCSI Input-Output Phase* – 48 mA, open drain. This signal is asserted with the MSG/ and C/D signals by the target to determine the information transfer phase. Input (asserted) and output (deasserted) transfers are always made with respect to the initiator. This signal can be directly connected to the single-ended SCSI I/O line. In differential mode, the TGS output controls the direction of this signal.
C/D	112	I/O	*SCSI Control-Data Phase* – 48 mA, open drain. This signal is asserted with the MSG/ and I/O signals by the target to determine the information transfer phase. This signal can be directly connected to the single-ended SCSI C/D line. In differential mode, the TGS output controls the direction of this signal.

Table 3-3. SCSI Phases

MSG/	C/D	I/O	SCSI Phase
0	0	0	Data-Out
0	0	1	Data-In
0	1	0	Command
0	1	1	Status
1	0	0	Reserved for future standard
1	0	1	Reserved for future standard
1	1	0	Message-Out
1	1	1	Message-In

Key: “0” not asserted, “1” asserted

Table 3-2. SCSI Interface Signals (Continued)

Symbol	Pin #	Type	Description
BSY/	119	I/O	*SCSI Busy* – 48 mA open drain, active-low. This signal is asserted when the SCSI bus is busy. When a device wants to arbitrate to use the SCSI bus, it is driven active. Once the arbitration and selection phases are complete, the target drives this signal active. This signal can be connected directly to the single-ended SCSY BSY/ line. In differential mode, this signal is an input only and the BSYDIR signal asserts BSY/ on the SCSI bus.
RST/	116	I/O	*SCSI Reset* – 48 mA open drain, active-low. This signal is asserted to perform a SCSI bus reset. It can be directly connected to the single-ended SCSI RST line. In differential mode, it is an input only and the RSTDIR signal is used to assert RST/ on the SCSI bus. When the reset SCSI bus bit in the SCNTL1 register is set to 1, the RST/ signal is asserted and remains asserted until this bit is reset to 0.
SEL/	114	I/O	*SCSI Select* – 48 mA open drain, active-low. This signal is asserted to select or reselect another SCSI device. This signal can be directly connected to the single-ended SEL line. In differential mode, it is an input only and the SELDIR signal asserts SEL/ on the SCSI bus.
BSYDIR	108	O	*SCSI Busy/Direction* – This signal controls the assertion of the SCSI BSY/ signal in differential mode. Connect it to the driver enable of the differential pair transceiver. When this signal is high, BSY/ is asserted on the SCSI bus. When it is low, the differential pair drive is disabled. The SCSI termination resistors will deassert BSY/ by pulling it high. This signal is always valid even in single-ended mode.
RSTDIR	107	O	*SCSI Reset/Direction* – This signal controls the assertion of the SCSI RST/ signal in differential mode. Connect it to the drive enable of the differential pair transceiver. When this signal is high, RST/ is asserted on the SCSI bus. When it is low, the differential pair driver is disabled. The SCSI termination resistors will then deassert RST/ by pulling RST/ high. This signal is always valid, even in single-ended mode.

Table 3-2. SCSI Interface Signals (Continued)

Symbol	Pin #	Type	Description
SELDIR	106	O	*SCSI Select/Direction* – This signal controls the direction of the SCSI SEL/ signal in differential mode. Connect this signal to the driver enable of the differential pair transceiver. When it is high, SEL/ is asserted on the SCSI bus. When it is low, the differential pair driver is disabled. The SCSI termination resistors will then deassert SEL/by pulling SEL/ high. This signal is always valid even in single-ended mode.
TGS	103	O	*Target Group Select* – This signal enables the external transceivers to drive SCSI REQ/, MSG/, C/D, and I/O when the SIOP is operating as a target in differential mode. When this signal is high, REQ/, MSG/, C/D, and I/O are outputs. When it is low, REQ/, MSG/, C/D, and I/O are inputs. This signal is always valid, even in single-ended mode.
IGS	104	O	*Initiator Group Select* – This signal enables the external transceivers to drive SCSI ACK/ and ATN/ when the SIOP operates as an initiator in differential mode. When this signal is high, ACK/ and ATN/ are outputs. When this signal is low, ACK/ and ATN/ are inputs. This signal is always valid, even in single-ended mode.

Chapter Four
Registers

This chapter contains descriptions of all SIOP registers. CTEST8, CTEST9, SCRATCHA and SCRATCHB registers are for the 53C700-66 chip only. CTEST7 bit 5 is reserved in the 53C700 chip and used in the 53C700-66 chip.

Table 4-1. Register Addresses and Descriptions

Address (Hex)	Read/ Write	Abbreviation	Description
00	R/W	SCNTL0	SCSI Control 0
01	R/W	SCNTL1	SCSI Control 1
02	R/W	SDID	SCSI Destination ID
03	R/W	SIEN	SCSI Interrupt Enable
04	R/W	SCID	SCSI Chip ID
05	R/W	SXFER	SCSI Transfer
06	R/W	SODL	SCSI Output Data Latch
07	R/W	SOCL	SCSI Output Control Latch
08	R	SFBR	SCSI First Byte Received
09	R	SIDL	SCSI Input Data Latch
0A	R	SBDL	SCSI Bus Data Lines
0B	R/W	SBCL	SCSI Bus Control Lines
0C	R	DSTAT	DMA Status
0D	R	SSTAT0	SCSI Status 0
0E	R	SSTAT1	SCSI Status 1
0F	R	SSTAT2	SCSI Status 2
10-13		*SCRATCHA	*General purpose scratch pad A
14	R	CTEST0	Chip Test 0
15	R	CTEST1	Chip Test 1
16	R	CTEST2	Chip Test 2
17	R	CTEST3	Chip Test 3
18	R/W	CTEST4	Chip Test 4
19	R/W	CTEST5	Chip Test 5
1A	R/W	CTEST6	Chip Test 6
1B	R/W	*CTEST7	*Chip Test 7
1C-1F	R/W	TEMP	Temporary Stack
20	R/W	DFIFO	DMA FIFO
21	R/W	ISTAT	Interrupt Status
22	R/W	*CTEST8	*Chip Test 8 (53C700-66 only)
23	R	*CTEST9	*Chip Test 9 (53C700-66 only)
24-26	R/W	DBC	DMA Byte Counter
27	R/W	DCMD	DMA Command
28-2B	R/W	DNAD	DMA Next Address for Data
2C-2F	R/W	DSP	DMA SCRIPTS Pointer
30-33	R/W	DSPS	DMA SCRIPTS Pointer Save
34	R/W	DMODE	DMA Mode
35-38		RES	Reserved
39	R/W	DIEN	DMA Interrupt Enable
3A	R/W	DWT	DMA Watchdog Timer
3B	R/W	DCNTL	DMA Control
3C-3F		*SCRATCHB	*General purpose scratch pad B

* 53C700-66 registers only and CTEST7 bit 5 is for the 53C700-66 only

Table 4-2. Register Address Map

<table>
<tr><th></th><th></th><th></th><th></th><th>Address (Hex)</th></tr>
<tr><td>SIEN (R/W)</td><td>SDID (R/W)</td><td>SCNTL1 (R/W)</td><td>SCNTL0 (R/W)</td><td>00</td></tr>
<tr><td>SOCL (R/W)</td><td>SODL (R/W)</td><td>SXFER (R/W)</td><td>SCID (R/W)</td><td>04</td></tr>
<tr><td>SBCL (R/W)</td><td>SBDL (R)</td><td>SIDL (R)</td><td>SFBR (R)</td><td>08</td></tr>
<tr><td>SSTAT2 (R)</td><td>SSTAT1 (R)</td><td>SSTAT0 (R)</td><td>DSTAT (R)</td><td>0C</td></tr>
<tr><td colspan="4">*SCRATCH A **(R/W)</td><td>10</td></tr>
<tr><td>CTEST3 (R)</td><td>CTEST2 (R)</td><td>CTEST1 (R)</td><td>CTEST0 (R)</td><td>14</td></tr>
<tr><td>CTEST7 (R/W)</td><td>CTEST6 (R/W)</td><td>CTEST5 (R/W)</td><td>CTEST4 (R/W)</td><td>18</td></tr>
<tr><td colspan="4">TEMP</td><td>1C</td></tr>
<tr><td>*CTEST9 (R)</td><td>*CTEST8 (R/W)</td><td>***ISTAT (R/W)</td><td>DFIFO (R/W)</td><td>20</td></tr>
<tr><td>DCMD (R/W)</td><td colspan="3">DBC (R/W)</td><td>24</td></tr>
<tr><td colspan="4">DNAD (R/W)</td><td>28</td></tr>
<tr><td colspan="4">DSP (R/W)</td><td>2C</td></tr>
<tr><td colspan="4">DSPS (R/W)</td><td>30</td></tr>
<tr><td colspan="3">RESERVED</td><td>DMODE (R/W)</td><td>34</td></tr>
<tr><td>DCNTL (R/W)</td><td>DWT (R/W)</td><td>DIEN (R/W)</td><td>RESERVED</td><td>38</td></tr>
<tr><td colspan="4">*SCRATCH B **(R/W)</td><td>3C</td></tr>
</table>

* 53C700-66 chip only (Register CTEST7 bit 5 is for 53C700-66 only)
** Read/Write by external processor only
*** This is the only register in the SIOP that can be accessed while fetching and executing SCRIPTS

Register Descriptions

This section contains descriptions of all SIOP registers. Table 4-1 summarizes the SIOP register set Table 4-2 shows a more graphical representation of the register set. The terms "set" and "assert" are used to refer to bits that are programmed to a binary one. Similarly, the terms "deassert," "clear" and "reset" are used to refer to bits that are programmed to a binary zero. Reserved bits are designated as "RES" in each register map. These bits should always be written to zero; mask all information read from them. Reserved bit functions may be changed at any time. Unless otherwise indicated, all bits in registers are active high, i.e., the feature is enabled by setting the bit. The bottom of every register diagram shows the default register values, which are enabled after the chip is powered on or reset. Registers can be addressed as bytes, words, or longwords. Other access sizes can result in bus errors.

Note: The only register that the host CPU can access while the SIOP is executing SCRIPTS is the ISTAT register, attempts to access other registers will interfere in the operation of the chip.

SCSI Control 0 (SCNTL0)
Address 00 Read/Write

	ARB1	ARB0	STRT	WATN	EPC	EPG	AAP	TRG
	7	6	5	4	3	2	1	0
Default >>>	1	1	0	0	0	0	0	0

Bit 7 ARB1 (Arbitration Mode bit 1)

Bit 6 ARB0 (Arbitration Mode bit 0)

ARB1	ARB0	Arbitration Mode
0	0	Simple Arbitration
1	1	Full Arbitration, Selection or Reselection

Start an arbitration or selection sequence by setting the Start Sequence bit in this register. The sequence can be aborted by resetting the Start Sequence bit. Check the connected bit in the SCNTL1 register to verify that the SIOP is not connected to the SCSI bus before starting any arbitration sequences. If the connected bit is set to 1, the SIOP has been selected or reselected and is already connected to the SCSI bus.

Simple Arbitration

In this mode, the SIOP waits for a bus free condition to occur, asserts BSY/ and asserts the contents of the SCID register onto the SCSI bus. The chip ID should be written to the SCID register before setting the start sequence bit. If the SEL/ signal is asserted by another SCSI device, the SIOP will deassert BSY/, deassert its ID and set the Lost Arbitration bit in the SSTAT1 register. When operating in this mode, the firmware should read the SBDL register to check if a higher priority SCSI ID is present.

The programming sequence is as follows:

1. Clear low-level mode bit
2. Set 53C700 chip ID (SCID register and SODL register)
3. Enable simple arbitration (reset SCNTL0 bit 6 and 7)
4. Start simple arbitration sequence, (set SCNTL0 bit 5)
5. Wait for either:
   A. Arbitration in progress (SSTAT1 bit 4 = 1) or
   B. Lost arbitration (SSTAT1 bit 3 = 1) or
   C. SEL asserted (SBCL bit 4 = 0)
6. If arbitration in progress, goto step 9
7. Wait for SEL deasserted (SBCL bit 4 = 0)
8. Goto step 4
9. Wait 2400 ns
10. If lost arbitration (SSTAT1 bit 3 = 1) goto step 7
11. If our ID is the highest asserted (compare our ID with any other IDs present in SBDL) goto step 13
12. Goto step 10
13. Assert BSY (set SOCL bit 5)
14. If target mode, set I/O (set SOCL bit 0)
15. Set assert data bus bit (set SCNTL1 bit 6)
16. Set connected bit (set SCNTL1 bit 4)
17. Assert BSY and SEL (set SOCL bits 5 and 4)
18. If not lost arbitration (SSTAT1 bit 3 = 0) goto step 22
19. Clear SOCL register
20. Clear assert data bus, connected bits (clear SCNTL1 bits 6 and 4)
21. Goto step 7
22. Wait 1200 ns
23. Clear start sequence bit (clear SCNTL0 bit 5)
24. Set SODL to bitwise OR of chip ID and destination ID
25. If initiator, set ATN if desired (set SOCL bit 3)
26. Wait 90 ns
27. Deassert BSY (clear SOCL bit 5)
28. Wait 400 ns
29. Wait for either:

    A. BSY asserted (SBCL bit 5 = 1) or

    B. Timeout (250 ms recommended, determine by maximum number times through this loop)
30. If BSY was asserted (step 29-A) goto step 37
31. Clear assert data bus bit (clear SCNTL1 bit 6)
32. Wait 201 μs
33. If BSY is asserted (SBCL bit 5 = 1) goto step 37
34. Clear connected bit (clear SCNTL1 bit 4)
35. Clear SOCL register
36. Goto step 4
37. Deassert SEL (clear SOCL bit 4)
38. Clear assert data bus bit (clear SCNTL1 bit 6)
39. Proceed with information transfers (SCRIPTS or low-level)

Full Arbitration, Selection & Reselection

In this mode, the SIOP waits for a bus free condition, then it asserts BSY/ and asserts its SCSI ID (the highest priority ID stored in the SCID register) onto the SCSI bus. If the SEL/ signal is asserted by another SCSI device or if the SIOP detects a higher priority ID, the SIOP will deassert BSY/, deassert its ID, and wait until the next bus free state to try arbitration again. The SIOP repeats arbitration until it wins control of the SCSI bus.

When the SIOP wins arbitration, the Won Arbitration Bit is set in the SSTAT1 register. After winning arbitration, the SIOP performs selection by asserting the following onto the SCSI bus: SEL/, the target's ID (stored in the SDID register) and the SIOP's ID (the highest priority ID stored in the SCID register). After a selection is complete, the Function Complete bit is set to 1 in the SSTAT0 register. If a selection time-out occurs, the Selection Time-out bit is set to 1 in the SSTAT0 register.

The programming sequence is as follows:

1. SCID = chip ID
2. SDID = target ID
3. SCNTL0 = full arb, selection (bit 6, 7)
4. SCNTL0 = start sequence, full arb, selection (bit 7, 6, 5)
5. Wait for function complete or select time-out SSTAT0 = function complete (bit 6) or select time-out (bit 5)

Bit 5 START (Start Sequence)

When this bit is written to 1, one of the arbitration or selection sequences will start. The SIOP will arbitrate and/or select according to the Arbitration Mode bits. While executing SCSI SCRIPTS™, the Start Sequence is controlled by the SCRIPTS PROCESSOR. Use the Start Sequence bit in low-level mode or register level programming. The start sequence must be completed (chip connected, SCNTL1, bit 4) before initializing low-level mode (DCNTL, bit 3). Otherwise, an unexpected disconnect will occur. This bit is cleared automatically when the selection sequence is complete.

The arbitration sequence can be aborted by resetting this bit to 0. If the sequence is aborted, check the connected bit in the SCNTL1 register, bit 4, to verify that the SIOP is not connected on the SCSI bus. The connected bit is set to indicate that the SIOP has won arbitration.

Bit 4 WATN (Select with ATN/ on a Start Sequence)

When set to 1, the SCSI ATN/ signal is asserted during the selection phase. ATN/ is asserted when BSY/ is deasserted to select a target device. When executing SCSI SCRIPTS, this bit is initialized by the SCRIPTS PROCESSOR. This bit is used in low-level mode or register level programming. While attempting to select a target device and a selection time-out occurs, ATN/ is deasserted when SEL/ is deasserted. The ATN/ signal is not asserted during selection if the Select with ATN/ bit is reset to 0.

Bit 3 EPC (Enable Parity Checking)

When set to 1, the SCSI data bus is checked for odd parity when SCSI data is received. The host data bus is checked for odd parity if parity generation is disabled (SCNTL0, bit 2). Host data bus parity is checked as data is loaded into the SODL register when sending SCSI data. If a parity error is detected, bit 0 of the SSTAT0 register is set to 1 and an interrupt can be generated (SIEN, bit 0).

If the SIOP is operating in target mode, you can stop bytes with parity errors written into the DMA FIFO from being written to the SCSI bus (SXFER register, bit 7 = 0).

If the SIOP is operating in initiator mode and a parity error is detected, ATN/ can optionally be asserted (SCNTL0, bit 1). The transfer continues until the target changes phase to Message-Out indicating a parity error has been detected. The parity error is reported when the current SCRIPTS Block Move instruction has executed and/or the target changes phase to message out.

If this bit is written to 0, parity errors are not reported.

Bit 2 EPG (Enable Parity Generation/ Parity Through)

When set to 1, the SCSI parity bit will be generated by the SIOP. The host data bus parity lines, DP3 - DP0, are ignored and should not be used as parity signals. The DP3_ABRT/ signal can be used as an Abort signal (ABRT/).

If this bit is written to 0, the parity present on the host data parity lines, DP3 - DP0 will flow through the SIOP's internal FIFOs and be driven onto the SCSI bus when sending data.

Bit 1 AAP (Assert ATN/ on Parity Error)

When set to 1, the SIOP automatically asserts the SCSI ATN/ signal upon detection of a parity error. ATN/ is only asserted in initiator mode. The ATN/ signal is asserted before deasserting ACK/ during the byte transfer with a parity error. The Enable Parity Checking bit must also be set to 1 for the SIOP to assert ATN/.

If the Assert ATN/ on Parity Error bit is written to 0 or the Enable Parity Checking bit is written to 0, ATN/ will not be asserted on the SCSI bus when a parity error is received.

Bit 0 TRG (Target Mode)

When set to 1, the chip is a target device. There are instances when the chip may change modes from initiator to target and vice versa. For example, an initiator device can be selected as a target. A mode change does not affect the state of this bit. After completion of a mode change I/O operation, the SIOP defaults to the role defined by this bit.

When the Target Mode bit is reset to 0, the SIOP is an initiator device.

SCSI Control 1 (SCNTL1)
Address 01 Read/Write

EXC	ADB	ESR	CON	RST	AESP	SND	RCV
7	6	5	4	3	2	1	0

Default >>>

0	0	0	0	0	0	0	0

Bit 7 EXC (Extra Clock Cycle of Data Setup)

When set to 1, an extra SCSI clock, as defined by the DCNTL register, of data setup is added to each SCSI Send transfer. The extra clock cycle can provide additional system design margin. In the 53C700-66, the SBCL register can affect the definition of a SCSI clock period. The extra clock cycle only affects the Send transfer period. The transfer period must be adjusted when receiving data to maintain equal Send and Receive transfer rates. For example, during Send: XFERP = n + EXC. During Receive: XFERP = n + 1.

Bit 6 ADB (Assert contents of the SODL onto the SCSI data bus)

When set to 1, the SIOP asserts the SCSI data bus with the contents of the SCSI Output Data Latch (SODL).

As an initiator, to assert SODL, the phase lines (MSG/, C/D, I/O) in the SOCL register must be set to match the phase asserted by the target (SBCL). The SCSI I/O signal must be inactive to assert the SODL contents onto the SCSI bus. When the SIOP is a target, the SCSI I/O signal must be active for the SODL contents to be asserted onto the SCSI bus.

This bit should be written to 0 when executing SCSI SCRIPTS. Use it for register level programming or low-level mode.

Bit 5 ESR (Enable the SIOP to respond to Selection & Reselection)

When set to 1, the SIOP is enabled to respond to bus-initiated selections and reselections. While executing WAIT for Selection or Reselection instruction the SIOP can respond to selections and reselections in both the initiator and the target roles. If disconnect - reconnect is to be supported, write this bit to 1 as part of the initialization routine.

Bit 4 CON (Connected)

This bit is automatically set to 1 after winning arbitration or after the SIOP has responded to a bus-initiated selection or reselection. It should be written to 1 after successfully completing simple arbitration when operating in low-level mode.

If this bit is cleared after the SIOP is connected, an unexpected disconnect will occur. When set to 0, the SIOP is not connected to the SCSI bus.

Bit 3 RST (Assert SCSI RST/ signal)

Writing this bit to 1 asserts the SCSI RST/ signal. The RST/ output remains asserted until this bit is written to 0. The 25 µsec minimum assertion time defined in the SCSI specification must be timed out by the controlling microprocessor.

Bit 2 AESP (Assert Even SCSI Parity (force bad parity))

When set to 1 and the Enable Parity Generation bit is set in the SCNTL0 register, the SIOP asserts even parity on the SCSI bus. It forces a SCSI parity error on each byte sent to the SCSI bus from the SIOP. However, when parity checking is enabled (SCNTL0, bit 3), it always checks for odd parity when receiving data across the SCSI bus. Use this bit for diagnostic testing. Reset it to 0 for normal operation. Use it to generate a parity error to test error handling functions.

Bit 1 SND (Start SCSI Send operation)

Setting this bit to 1 initiates a SCSI send operation. Bytes in the DMA FIFO will be sent across the SCSI bus. It is automatically set to 1 by the SCRIPTS PROCESSOR to start a SCSI send operation when executing SCSI SCRIPTS™. It is intended for register level programming or low-level mode.

Bit 0 RCV (Start SCSI Receive operation)

Setting this bit to 1 initiates a SCSI receive operation. Bytes are received from the SCSI bus into the DMA FIFO (via the SCSI FIFO if synchronous). It is automatically set to 1 by the SCRIPTS PROCESSOR to start a SCSI receive operation when executing SCSI SCRIPTS™. Use it for register level programming or low-level mode.

SCSI Destination ID (SDID)
Address 02 Read/Write

ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
7	6	5	4	3	2	1	0
Default >>>							
0	0	0	0	0	0	0	0

Bits 7 - 0 ID7 - ID0 (SCSI ID 7 - SCSI ID0)

Use this register to select the desired SCSI device when executing a select or reselect command. Only one of these bits may be set to 1 for proper selection or reselection. When executing SCSI SCRIPTS™, the SCRIPTS Processor writes the destination SCSI ID to this register. The SCSI ID is defined by the user in a SCSI SCRIPTS™ select or reselect instruction .

SCSI Interrupt Enable (SIEN)
Address 03 Read/Write

M/A	FC	STO	SEL	SGE	UDC	RST/	PAR
7	6	5	4	3	2	1	0
Default >>>							
0	0	0	0	0	0	0	0

This register is the interrupt mask register for the interrupting conditions described in the SSTAT0 register. Each condition is maskable and has its own interrupt enable bit.

Bit 7 M/A (Initiator mode: Phase Mismatch or Target mode: ATN/ active)

In the initiator mode, this bit is the Phase Mismatch bit. When the Phase Mismatch bit is set to 1, the IRQ/ signal is asserted if the current SCSI phase does not match the expected SCSI phase defined in a SCSI SCRIPTS™ Block Move instruction. For register level programming or low-level mode, the phase lines in SOCL must match the current SCSI phase driven by the target (SBCL).

In the target mode, this bit is the ATN/ Active bit. When set to 1, the IRQ/ signal is asserted when ATN/ is detected. The Disable halt on the ATN/ bit in the SXFER register controls when the SIOP will assert the IRQ/ signal after ATN/ is received. If halt on ATN/ is disabled (bit 7 in the SXFER register = 1), the IRQ/ signal is asserted after the current SCSI transfer is complete. If halt on ATN/ is enabled (bit 7 in the SXFER register = 0), the IRQ/ signal is asserted at the time ATN/ is received. If ATN/ is received in the middle of a data transfer, the SIOP may transfer up to 3 additional bytes before halting to synchronize between internal core cells. During synchronous operation, the SIOP transfers data until there are no outstanding synchronous offsets.

If the SIOP is receiving data, any data residing in the SCSI or DMA FIFOs is sent to memory before halting. This interrupt is masked by writing this bit to 0.

Bit 6 FC (Function Complete)

When set to 1, the IRQ/ signal is asserted when a simple arbitration or full arbitration selection or reselection sequence has completed. This interrupt can be masked by resetting this bit to 0.

Bit 5 STO (Selection or Reselection Time-out)

When set to 1, the IRQ/ signal is asserted when a selection or reselection time-out occurs. A selection time-out occurs when the device being selected or reselected does not respond within the 250 msec time-out period. The interrupt can be masked by resetting this bit to 0.

Bit 4 SEL (Selected or Reselected)

When set to 1, the IRQ/ signal is asserted when the SIOP is selected or reselected by another SCSI device. The interrupt can be masked by writing this bit to 0.

Bit 3 SGE (SCSI Gross Error)

When set to 1, the IRQ/ signal is asserted when the SIOP detects a SCSI Gross Error condition. The interrupt can be masked by resetting this bit. The following conditions can cause a SCSI Gross Error condition.

1. Data Underflow - the SCSI FIFO register was read when no data was present.
2. Data Overflow - Too many bytes were written to the SCSI FIFO.
3. Offset Underflow - When the SIOP is operating in target mode, and an ACK/ pulse was received when the outstanding offset was zero.
4. Offset Overflow - the other SCSI device sent a REQ/ or ACK/ pulse with data which exceeded the maximum synchronous offset defined by the SXFER register.
5. Residual data in the Synchronous data FIFO - a transfer other than synchronous data receive was started with data left in the synchronous data FIFO.
6. A phase change occurred with an outstanding synchronous offset when the SIOP is operating as an initiator.

Bit 2 UDC (Unexpected Disconnect)

When set to 1, the IRQ/ signal is asserted when a target device unexpectedly disconnects from the SCSI bus. This bit is valid when the SIOP is in initiator mode.

When the SIOP is executing SCSI SCRIPTS™, an unexpected disconnect is any disconnect other than a legal SCSI disconnect. A legal SCSI disconnect can occur after a Disconnect Message (04h) or a Command Complete Message (00h) is received as a message in. A select time-out or loss of busy at any other time is considered an unexpected disconnect. Refer to the SCSI specification for more detailed information on SCSI disconnects. In low-level Mode, any type of disconnect will cause an interrupt. This interrupt is masked by resetting this bit.

Bit 1 RST/ (SCSI RST/ Received)

When set to 1, the IRQ/ signal is asserted when the SIOP detects an active SCSI RST/ signal. The interrupt is masked by writing this bit to 0.

Bit 0 PAR (Parity Error)

When set to 1, the IRQ/ signal is asserted if the SIOP detects a parity error while receiving SCSI or host data as the data enters the SCSI core. Parity checking must be enabled (SCNTL0 register, bit 3).

In initiator mode, an interrupt is not generated until the data transfer is complete or until the target changes phases.

In the target mode, an interrupt is generated immediately upon receipt of bad parity. If a parity error is received from the SCSI bus in the middle of a data transfer, the SIOP may transfer up to 3 additional bytes to synchronize between internal core cells. Any data received from the SCSI bus residing in the DMA FIFO is sent to memory. While sending data in target mode with pass parity enabled, the byte with the parity error will not be sent across the SCSI bus. During synchronous operation, the SIOP transfers data until there are no outstanding synchronous offsets.

The interrupt is masked when this bit is reset to 0.

SCSI Chip ID (SCID)
Address 04 Read/Write

	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
	7	6	5	4	3	2	1	0
Default >>>	0	0	0	0	0	0	0	0

Bits 7 - 0 ID 7- ID 0 (SCSI ID 7- SCSI ID 0)

This register initializes the SIOP's SCSI ID. If more than one bit is set to 1, the SIOP will respond to each corresponding SCSI ID. The SIOP always uses the highest priority SCSI ID during arbitration. For example, if an 84 hex were written to this register, the SIOP would respond when another device selects ID 7 or ID 2. When arbitrating for the SCSI bus, ID 7 would be used as the SIOP's SCSI ID.

SCSI Transfer (SXFER)
Address 05 Read/Write

DHP	TP2	TP1	TP0	MO3	MO2	MO1	MO0
7	6	5	4	3	2	1	0

Default >>>

0	0	0	0	0	0	0	0

Bit 7 DHP (Disable Halt on a Parity Error or ATN/ (Target Mode Only))

In Target mode, this bit is defined as Disable Halt on Parity Error or ATN/. When this bit is reset to 0, the SIOP halts the SCSI data transfer when a parity error is detected or when the ATN/ signal is asserted.

While receiving data if ATN/ or a parity error is received in the middle of a data transfer, the SIOP may transfer up to 3 additional bytes before halting to synchronize between internal core cells. If the SIOP is receiving data, any data residing in the DMA FIFO is sent to memory before halting.

During synchronous operation, the SIOP transfers data until there are no outstanding synchronous offsets.

While sending data with pass parity enabled, the byte with the parity error received from the host will not be sent across the SCSI bus.

When set to 1, the SIOP does not halt the SCSI transfer when ATN/ or a parity error is received until the Block Move is complete.

Bit 6 TP2 (SCSI Synchronous Transfer Period bit 2)

Bit 5 TP1 (SCSI Synchronous Transfer Period bit 1)

Bit 4 TP0 (SCSI Synchronous Transfer Period bit 0)

These bits describe the SCSI synchronous transfer period used by the SIOP when sending synchronous SCSI data in either initiator or target mode. The following table describes the possible combinations.

Synchronous Transfer Periods Used by SIOP

TP2	TP1	TP0	XFERP
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

The actual Synchronous Transfer Period used by the SIOP when transferring SCSI data is defined by the following equations:

Given XFERP from the table above and TCP defined as:

TCP = 1 / (CLK input frequency / 2)
If Bit 7 = 0 & Bit 6 = 0 in the DCNTL register (SCSI clock frequency divide for 37.51-50 MHz)

TCP = 1 / (CLK input frequency / 1.5)
If Bit 7 = 0 & Bit 6 = 1 in the DCNTL register (SCSI clock frequency divide for 25.01-37.5 MHz)

TCP = 1 / (CLK input frequency / 1)

If Bit 7 = 1 & Bit 6 = 0 in the DCNTL register (SCSI clock frequency divide for 16.67-25 MHz)

The minimum Synchronous Transfer Period:

when sending SCSI data

$= TCP \times (4 + XFERP + 1)$

If Bit 7 = 1 in the SCNTL1 register (indicates extra clock cycle of data setup)

$= TCP \times (4 + XFERP)$

If Bit 7 = 0 in the SCNTL1 register (indicates no extra clock cycle of data setup)

when receiving SCSI data

$= TCP \times (4 + XFERP)$

The following table gives example transfer periods for the 53C700 and the 53C700-66.

CLK (MHz)	SCSI CLK / DCNTL bits 7, 6	XFERP	Synch Transfer Period (ns)	Synch Transfer Rate (MB/s)
66.67	/3.0	0	180	5.55
66.67	/3.0	1	225	4.44
50.00	/2.0	0	160	6.25
50.00	/2.0	1	200	5.00
40.00	/2.0	0	200	5.00
37.50	/1.5	0	160	6.25
33.33	/1.5	0	180	5.55
25.00	/1.0	0	160	6.25
20.00	/1.0	0	200	5.00
16.67	/1.0	0	240	4.17

The following table gives example transfer periods for fast transfers with the 53C700-66.

CLK (MHz)	SCSI CLK / SBCL bits 1, 0	XFERP	Synch Transfer Period (ns)	Synch Transfer Rate (MB/s)
66.67	/1.5	0	90.00	11.11*
66.67	/1.5	1	112.50	8.88
50.00	/1.0	0	80.00	12.50*
50.00	/1.0	1	100.00	10.00
40.00	/1.0	0	100.00	10.00
37.50	/1.0	0	106.67	9.375
33.33	/1.0	0	120.00	8.33
25.00	/1.0	0	160.00	6.25
20.00	/1.0	0	200.00	5.00
16.67	/1.0	0	240.00	4.17

* Violates SCSI specifications. Slower rates are achieved by using larger XFERP values and/or different SCLK prescale values.

Bit 3 MO3 (Maximum SCSI Synchronous Offset Bit 3)

Bit 2 MO2 (Maximum SCSI Synchronous Offset Bit 2)

Bit 1 MO1 (Maximum SCSI Synchronous Offset Bit 1)

Bit 0 MO0 (Maximum SCSI Synchronous Offset Bit 0)

These bits describe the maximum SCSI synchronous offset used by the SIOP when transferring synchronous SCSI data in either initiator or target mode. The following table describes the possible combinations and their relationship to the synchronous data offset used by the SIOP. These bits determine the SIOP's method of transfer for Data phases only – Data-In & Data-Out. All other information transfers will occur asynchronously.

SCSI Synchronous Offsets Used by the SIOP

MO3	MO2	MO1	MO0	Synchronous Offset
0	0	0	0	0 – Asynchronous
0	0	0	1	1 – Synchronous
0	0	1	0	2 – Synchronous
0	0	1	1	3 – Synchronous
0	1	0	0	4 – Synchronous
0	1	0	1	5 – Synchronous
0	1	1	0	6 – Synchronous
0	1	1	1	7 – Synchronous
1	0	0	0	8 – Synchronous

Hex numbers 9 to F are reserved (not used)

SCSI Output Data Latch (SODL) Address 06 Read/Write

SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0
7	6	5	4	3	2	1	0

Default >>>

0	0	0	0	0	0	0	0

Bit 7 SD7 (SCSI Output Data Latch Bit 7)

Bit 6 SD6 (SCSI Output Data Latch Bit 6)

Bit 5 SD5 (SCSI Output Data Latch Bit 5)

Bit 4 SD4 (SCSI Output Data Latch Bit 4)

Bit 3 SD3 (SCSI Output Data Latch Bit 3)

Bit 2 SD2 (SCSI Output Data Latch Bit 2)

Bit 1 SD1 (SCSI Output Data Latch Bit 1)

Bit 0 SD0 (SCSI Output Data Latch Bit 0)

This register is used primarily for diagnostics testing or register level programming. Data written to this register is asserted onto the SCSI data bus by setting the Assert Data Bus bit (SCNTL1 register, bit 6). As an initiator, the phase lines (MSG/, C/D, I/O) in the SOCL register (bits 2-0) must be written to match the phase asserted by the target in order to assert the contents of this register. Use this register to send data via programmed I/O. Data flows through this register when sending data to the SCSI bus in any mode. It is also used to write to the synchronous SCSI FIFO when testing the chip.

This register is useful for debugging by identifying the last data byte sent to the SCSI bus by the SIOP when an interrupt occurs.

SCSI Output Control Latch (SOCL)
Address 07 Read/Write

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

Default >>>

0 0 0 0 0 0 0 0

Bit 7 REQ (Assert SCSI REQ/ signal)

Bit 6 ACK (Assert SCSI ACK/ signal)

Bit 5 BSY (Assert SCSI BSY/ signal)

Bit 4 SEL (Assert SCSI SEL/ signal)

Bit 3 ATN Assert SCSI ATN/ signal)

Bit 2 MSG (Assert SCSI MSG/ signal)

Bit 1 C/D (Assert SCSI C/D signal)

Bit 0 I/O (Assert SCSI I/O signal)

This register is used primarily for diagnostics testing or register level programming. It is controlled by the SCRIPTS PROCESSOR when executing SCSI SCRIPTS. SOCL should only be used when transferring data via programmed I/O. Some bits are set (1) or reset (0) when executing SCSI SCRIPTS. Do not write to this register once the SIOP becomes connected and starts executing SCSI SCRIPTS.

In low-level mode this register must be initialized to the correct phase before the contents of the SODL register can be asserted on the SCSI bus. No data transfer will occur if there is a SCSI phase mismatch.

SCSI First Byte Received (SFBR)
Address 08 Read Only

1B7	1B6	1B5	1B4	1B3	1B2	1B1	1B0
7	6	5	4	3	2	1	0

Default >>>

0 0 0 0 0 0 0 0

Bit 7 1B7 (First Byte Received Bit 7)

Bit 6 IB6 (First Byte Received Bit 6)

Bit 5 1B5 (First Byte Received Bit 5)

Bit 4 IB4 (First Byte Received Bit 4)

Bit 3 1B3 (First Byte Received Bit 3)

Bit 2 IB2 (First Byte Received Bit 2)

Bit 1 1B1 (First Byte Received Bit 1)

Bit 0 IB0 (First Byte Received Bit 0)

This register contains the first byte received for a Block Move instruction. For example, when the SIOP is operating in initiator mode, this register contains the first byte received for a Block Move in any of the following phases:

- Message-In
- Status
- Data-In

When in target mode, this register contains the first byte received for a Block Move in any of the following phases:

- Command
- Message-Out
- Data-Out

When the SIOP is selected or reselected, this register contains the selecting or reselecting device's SCSI ID and the SIOP's SCSI ID. The register contents will change after a Block Move instruction is executed for receiving data.

SCSI Input Data Latch (SIDL)
Address 09 Read Only

SD7	SD6	SD	SD4	SD3	SD2	SD5	SD0
7	6	5	4	3	2	1	0

Default >>>

0	0	0	0	0	0	0	0

Bit 7 SD7 (SCSI Input Data Bit 7)

Bit 6 SD6 (SCSI Input Data Bit 6)

Bit 5 SD5 (SCSI Input Data Bit 5)

Bit 4 SD4 (SCSI Input Data Bit 4)

Bit 3 SD3 (SCSI Input Data Bit 3)

Bit 2 SD2 (SCSI Input Data Bit 2)

Bit 1 SD1 (SCSI Input Data Bit 1)

Bit 0 SD0 (SCSI Input Data Bit 0)

This register is used primarily for diagnostics testing, programmed I/O operation or error recovery. Data received from the SCSI bus can be read from this register. Data can be written to the SODL register and then read back into the SIOP by reading this register to provide "loopback" testing. When receiving SCSI data, the data will flow into this register and through to the DMA FIFO. The SIDL register differs from the SBDL register, because it contains latched data where the SBDL reflects what is currently on the SCSI data bus. Data is latched on the asserting edge of REQ/ for an initiator and ACK/ if we are a target.

This register is useful for debugging by identifying the last byte received from SCSI when an interrupt occurs.

SCSI Bus Data Lines (SBDL)
Address 0A Read Only

DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
7	6	5	4	3	2	1	0

Default >>>

0	0	0	0	0	0	0	0

Bit 7 SD7 (SCSI Data Bit 7)

Bit 6 SD6 (SCSI Data Bit 6)

Bit 5 SD5 (SCSI Data Bit 5)

Bit 4 SD4 (SCSI Data Bit 4)

Bit 3 SD3 (SCSI Data Bit 3)

Bit 2 SD2 (SCSI Data Bit 2)

Bit 1 SD1 (SCSI Data Bit 1)

Bit 0 SD0 (SCSI Data Bit 0)

This register contains the SCSI data bus status. These bits are active-high. A "1" indicates an active signal on the SCSI bus. The signal status is not latched and is a true representation of exactly what is on the data bus at the time that the register is read. Use this register when data is received via programmed I/O. This register can be used for diagnostics testing or in low-level mode.

This register is useful for debugging by identifying the current data byte being driven on the SCSI bus when an interrupt occurs.

Note: Before reading this register in the slave mode, disable parity checking (SCNTL0 bit 3) because reading this register causes new parity to be latched from the SCSI bus.

SCSI Bus Control Lines (SBCL)
Address 0B Read 53C700, Read/Write -66

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

Default >>>

Bit 7 REQ (REQ/ status)

Bit 6 ACK (ACK/ status)

Bit 5 BSY (BSY/ status)

Bit 4 SEL (SEL/ status)

Bit 3 ATN (ATN/ status)

Bit 2 MSG (MSG/ status)

Bit 1 C/D (C/D status)

Bit 0 I/O (I/O status)

When read, this register returns the SCSI control line status. A bit is asserted when the corresponding SCSI Control line is set. These bits are not latched; they are a true representation of what is on the SCSI bus at the time the register is read. This register can be used for diagnostics testing or in low-level mode. Writing to bits 7-2 has no effect.

Bits 1-0 SSCF1-0 (Synchronous SCSI Clock Control Bits) 53C700-66 only

SSCF1	SSCF0	Synchronous CLK
0	0	Set by DCNTL
0	1	SCLK / 1.0
1	0	SCLK / 1.5
1	1	SCLK / 2.0

When written, these bits determine the clock prescale factor used by the synchronous portion of the SCSI core. The default is to use the same clock prescale factor as the asynchronous logic (set by CF(1-0) in DCNTL). Setting one or both of these bits allows the synchronous logic to run at a different speed than the asynchronous logic; this is necessary for fast SCSI-2.

DMA Status (DSTAT)
Address 0C Read Only

DFE 7	RES 6	RES 5	ABRT 4	SSI 3	SIR 2	WTD 1	OPC 0
Default >>>							
1	0	0	0	0	0	0	0

Reading this register will clear any DMA interrupts that may have caused the IRQ/ signal to be asserted. DMA interrupts are masked by programming the DIEN register (39h).

Bit 7 DFE (DMA FIFO Empty)

This status bit is set to 1 when the DMA FIFO (36 x 8) is empty. This status bit may be changing at the time this register is read. Use it to determine if any data resides in the FIFO when an error occurs and an interrupt is generated.

Bits 6 - 5 RES (Reserved)

Bit 4 ABRT (Aborted)

This bit is set when an abort condition occurs. An abort condition occurs because of the following: the DP3_ABRT/ input signal is asserted (Parity Generation must be enabled) or a software abort command is issued by setting Bit 7 of the ISTAT register.

Bit 3 SSI (SCRIPTS Single Step Interrupt)

This status bit is set when an interrupt condition occurs during Single Step operation. Single step mode (DCNTL register, bit 4) or Pipeline Mode (DCNTL register, bit 3) must be enabled to receive a Single Step interrupt. The following conditions can cause a SCRIPTS Single Step Interrupt:

1. Interrupt occurs after successful execution of each SCRIPTS instruction in Single Step Mode (DCNTL register, bit 4).
2. Interrupt occurs if the SIOP encounters a branch condition while executing pipelined instructions (pipeline mode, DCNTL register, bit 3).

Bit 2 SIR (SCRIPTS Interrupt Instruction Received)

This status bit is set whenever a SCSI SCRIPTS interrupt (INT) instruction is executed.

Bit 1 WTD (Watchdog Time-out Detected)

This status bit is set when the Watchdog Timer Counter (DWT register) has decremented to zero. This only applies when the SIOP is in Master mode. If the counter decrements to zero, it indicates that the memory device did not assert its READYI/ signal to terminate the cycle within the specified time-out period from ADS/ asserted.

Bit 0 OPC (Illegal Instruction Detected)

This status bit is set anytime an illegal instruction is fetched. Causes of an illegal instruction are listed in the following:

1. Corruption of the SCRIPTS instruction fetched into the chip.

   This may occur if the system does not support bus mastering. The typical symptom is for FFs to appear in the DCMD, DBC or DNAD (or DSPS) when this interrupt occurs. DCMD, DBC and DNAD (or DSPS) contain the SCRIPTS instruction fetched.

   This may also occur if the SIOP Master Mode read/write cycle time is too fast for the system bus or memory. Wait states may need to be inserted by delaying READYI/ to the SIOP. See Bus Master timings in Appendix A for more information.

2. A Block Move instruction in a target SCRIPTS (move byte_count, with DATA_IN) fetched when the chip is in initiator mode (SCNTL0, bit 0).

3. A Block Move instruction in an initiator SCRIPTS (move byte_count, when DATA_IN) fetched when the chip is in target mode (SCNTL0, bit 0).

4. A wait disconnect instruction is fetched after the chip has been disconnected then reselected. In this instance, a target disconnected and another device reselected the SIOP before the wait disconnect instruction was fetched and executed. Bit 0 in CTEST8 can be set to avoid this condition on the 53C700-66.

5. An indirect Block Move instruction is executed while operating in pipeline mode.

Note: If executing 8-bit reads of the DSTAT and SSTAT0 registers to clear interrupts, insert 10 CLKs (one or two NOPs) between the consecutive reads of the DSTAT or SSTAT0 registers to ensure that the interrupt clears properly. For example:

1. Read DSTAT (to clear the DMA interrupt)
2. Read ISTAT. This guarantees a system independent delay of 10 CLKS (one slave mode cycle).
3. Read SSTAT0 to clear the SCSI interrupt.

SCSI Status 0 (SSTAT0)
Address 0D Read Only

	M/A 7	CMP 6	STO 5	SEL 4	SGE 3	UDC 2	RST/ 1	PAR 0
Default >>>	0	0	0	0	0	0	0	0

Each of these bits correspond to a SCSI condition that causes the SIOP to generate an interrupt. The SCSI interrupts are individually masked by programming the SIEN register (address 03h).

Bit 7 M/A (Initiator mode: Phase Mismatch or Target mode: ATN/ active)

In the initiator mode, this bit is set if the SCSI phase asserted by the target does not match the SCSI phase defined in a Block Move instruction. In low-level mode, this bit is set if the SCSI phase asserted by the Target does not match the phase in the SOCL Register. The phase is sampled when REQ/ is asserted by the target.

In the target mode, this bit is set when the ATN/ signal is asserted by the initiator.

Bit 6 CMP (Function Complete)

This bit is set to 1 when a simple arbitration, or full arbitration with selection or reselection sequence has completed.

Bit 5 STO (Selection or Reselection Time-out)

This bit is set to 1 when a selection or reselection time-out occurs. A time-out occurs when the device being selected or reselected did not respond within the specified 250 msec time-out period.

Bit 4 SEL (Selected or Reselected)

This bit is set to 1 when the SIOP is selected or reselected by another SCSI device. The Enable Selection/Reselection bit must be set to 1 in the SCNTL1 register for the SIOP to respond to selection or reselection.

Bit 3 SGE (SCSI Gross Error)

This bit is set to 1 when the SIOP encounters a SCSI Gross Error condition. The following conditions can cause a SCSI Gross Error condition.

1. Data Underflow - the SCSI FIFO register was read when no data was present.
2. Data Overflow - Too many bytes were written to the SCSI FIFO.
3. Offset Underflow - When the SIOP is operating in target mode and an ACK/ pulse is received when the outstanding offset is zero.
4. Offset Overflow - the other SCSI device sent a REQ/ or ACK/ pulse with data which exceeded the maximum synchronous offset defined by the SXFER register.
5. Residual data in the synchronous data FIFO – a transfer other than synchronous data receive was started with data left in the synchronous data FIFO.
6. A phase change occurred with an outstanding synchronous offset when the SIOP is operating as an initiator.

Bit 2 UDC (Unexpected Disconnect)

When set to 1, the IRQ/ signal is asserted when a target device unexpectedly disconnects from the SCSI bus. This bit is valid when the SIOP is in initiator mode.

When the SIOP is executing SCSI SCRIPTS™, an unexpected disconnect is any disconnect other than a legal SCSI disconnect. A legal SCSI disconnect can occur after a Disconnect Message (04h) or a Command Complete Message (00h) is received as a message in. A select time-out or loss of busy at any other time is considered an unexpected disconnect. Refer to the SCSI specification for more detailed information on SCSI disconnects. In low-level Mode, any type of disconnect will cause an interrupt. This interrupt is masked by resetting this bit.

Bit 1 RST/ (SCSI RST/ Received)

This bit is set to 1 by the following conditions: the SIOP detects an active RST/ signal or the Assert RST/ bit in the SCNTL1 register is set to 1. This status bit is edge-triggered so that multiple interrupts do not occur for one assertion of the SCSI RST/ signal.

Bit 0 PAR (Parity Error)

This bit is set to 1 when the SIOP detects a parity error when sending or receiving SCSI data. The Enable Parity Checking bit (bit 3 in the SCNTL0 register) must be set for this bit to become active. A parity error can occur when receiving data from the SCSI bus or when receiving data from the host bus. From the host bus, parity is checked as it is transferred from the DMA FIFO to the SODL register. A parity error can occur from the host bus only if pass parity is allowed (bit 3 in the SCNTL0 register = 1, bit 2 in the SCNTL0 register = 0).

Note: If executing 8-bit reads of the DSTAT and SSTAT0 registers to clear interrupts, insert 10 CLKs (one or two NOPs) between the consecutive reads of the DSTAT or SSTAT0 registers to ensure that the interrupt clears properly. For example:

1. Read DSTAT (to clear the DMA interrupt)
2. Read ISTAT. This guarantees a system independent delay of 10 CLKS (one slave mode cycle).
3. Read SSTAT0 to clear the SCSI interrupt.

SCSI Status 1 (SSTAT1)
Address 0E Read Only

ILF	ORF	OLF	AIP	LOA	WOA	RST/	SDP/
7	6	5	4	3	2	1	0

Default >>>

0	0	0	0	0	0	0	0

Bit 7 ILF (SIDL Register Full)

This bit is set to 1 when the SCSI Input Data Latch register (SIDL) contains data. Data is transferred from the SCSI bus to the SCSI Input Data Latch register before being sent to the DMA FIFO and then to the host bus. The SIDL register contains SCSI data received asynchronously. Synchronous data received does not flow through this register.

This bit can be used to determine how many bytes reside in the chip when a DMA error occurs receiving data from SCSI.

Bit 6 ORF (SODR Register Full)

This bit is set to 1 when the SCSI Output Data Register (SODR, a hidden buffer register) contains data. The SODR register is used by the SCSI logic as a second storage register when sending data synchronously. It is a not accessible to the user (cannot be read or written).

This bit can be used to determine how many bytes reside in the chip when an error occurs during a SCSI data transfer.

Bit 5 OLF (SODL Register Full)

This bit is set to 1 when the SCSI Output Data Latch (SODL) contains data. The SODL register is the interface between the DMA logic and the SCSI bus. In synchronous mode, data is transferred from the host bus to the SODL register and then to the SCSI Output Data Register (SODR), (a hidden buffer register which is not accessible), before being sent to the SCSI bus. In asynchronous mode, data is transferred from the host bus to the SODL register, and then to the SCSI bus. The SODR buffer register is not used for asynchronous transfers.

This bit can be used to determine how many bytes reside in the chip when an error occurs during data transfer.

Bit 4 AIP (Arbitration In Progress)

Arbitration in Progress (AIP = 1) indicates that the SIOP has detected a bus free condition, asserted BSY and asserted its SCSI ID onto the SCSI bus.

Bit 3 LOA (Lost Arbitration)

Lost Arbitration (LOA = 1) indicates that the SIOP has detected a bus free condition, arbitrated for the SCSI bus, and lost arbitration due to another SCSI device asserting the SEL/ signal.

Bit 2 WOA (Won Arbitration)

Won Arbitration (WOA = 1) indicates that the SIOP has detected a bus free condition, arbitrated for the SCSI bus and won arbitration.

Bit 1 RST/ (SCSI RST/ Signal)

This bit represents the current status of the SCSI RST/ signal. This signal is not latched and may be changing when read.

Bit 0 SDP/ (SCSI SDP/ Parity Signal)

This bit represents the current status of the SCSI SDP/ parity signal. This signal is not latched and may be changing when read.

SCSI Status 2 (SSTAT2)
Address 0F Read Only

FF3 7	FF2 6	FF1 5	FF0 4	SDP 3	MSG 2	C/D 1	I/O 0
Default >>>							
0	0	0	0	0	0	0	0

Bit 7 FF3 (FIFO Flags bit 3)

Bit 6 FF2 (FIFO Flags bit 2)

Bit 5 FF1 (FIFO Flags bit 1)

Bit 4 FF0 (FIFO Flags bit 0)

These four bits define the number of bytes that currently reside in the SIOP's SCSI synchronous data FIFO. These bits are not latched and they will change as data moves in and out of the FIFO. The following chart describes the possible combinations and each corresponding value.

Number of Bytes Residing in Synchronous Data FIFO

FF3	FF2	FF1	FF0	Number of bytes in the SCSI FIFO
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8

Because the FIFO is 8 deep, any value over 8 will not occur.

Bit 3 SDP (Latched SDP/ SCSI Parity)

This status bit reflects the SCSI parity signal corresponding to the data latched in the SCSI Input Data Latch register (SIDL). It changes when a new byte is latched into the SIDL register. When this bit is 1, the parity signal is active. When this bit is 0, the parity signal is inactive.

Bit 2 MSG (SCSI MSG/ phase signal - latched by REQ/)

Bit 1 C/D (SCSI C/D phase signal - latched by REQ/)

Bit 0 I/O (SCSI I/O phase signal - latched by REQ/)

These SCSI phase status bits are latched on the asserting edge of REQ/ when operating in either initiator or target mode. These bits are set to 1 when the MSG, C/D or I/O signals are active.

Scratch A (SCRATCHA)
Address 10-13 Read/Write

Default >>> all zeros

This register is available only in the 53C700-66 chip

This is a general purpose user definable scratch pad register. Normal SCRIPTS operations will not destroy the contents of this register, only slave Read/Writes into the SCRATCHA register will alter its contents. This register can not be accessed by the internal SCSI I/O processor.

Chip Test 0 (CTEST0)
Address 14 Read Only

RES	RES	RES	RES	RES	RES	RTRG	DDIR
7	6	5	4	3	2	1	0

Default >>>

X	X	X	X	X	X	0	0

X = Don't Care

Bits 7-2 RES (Reserved)

Bit 1 RTRG (Real Target Mode)

This status bit indicates the operating mode of the logic inside the SIOP. It does not reflect the status of the Target Mode bit in the SCNTL0 register. For example, the Target Mode bit in the SCNTL0 register might be written to 0 indicating that the SIOP is operating in initiator mode. However, if the SIOP is selected as a target, this bit will indicate that the SIOP has been selected as a target. When this bit is 1, the SIOP is actually operating as a target, and when this bit is 0, the SIOP is actually operating as an initiator. If the SIOP is idle or disconnected, this bit will reflect the status of the Target Mode bit in the SCNTL0 register.

Bit 0 DDIR (Data Transfer Direction)

This status bit indicates which direction data is being transferred. When this bit is 1, the data will be transferred from the SCSI bus to the host bus. When this bit is 0, the data will be transferred from the host bus to the SCSI bus.

Chip Test 1 (CTEST1)
Address 15 Read Only

	FMT3	FMT2	FMT1	FMT0	FFL3	FFL2	FFL1	FFL0
	7	6	5	4	3	2	1	0
Default >>>	1	1	1	1	0	0	0	0

Bit 7 FMT3 (Byte 3 Empty in the DMA FIFO)

Bit 6 FMT2 (Byte 2 Empty in the DMA FIFO)

Bit 5 FMT1 (Byte 1 Empty in the DMA FIFO)

Bit 4 FMT0 (Byte 0 Empty in the DMA FIFO)

These status bits identify the bottom bytes in the DMA FIFO that are empty. Each bit corresponds to a byte lane in the DMA FIFO. For example, if byte lane 3 is empty, then FMT3 will be 1. Since the FMT flags indicate the status of bytes at the bottom of the FIFO, if all FMT bits are 1, the DMA FIFO is empty.

Bit 3 FFL3 (Byte 3 Full in the DMA FIFO)

Bit 2 FFL2 (Byte 2 Full in the DMA FIFO)

Bit 1 FFL1 (Byte 1 Full in the DMA FIFO)

Bit 0 FFL0 (Byte 0 Full in the DMA FIFO)

These status bits identify the top bytes in the DMA FIFO that are full. Each bit corresponds to a byte lane in the DMA FIFO. For example, if byte lane 3 is full, then FFL3 will be 1. Since the FFL flags indicate the status of bytes at the top of the FIFO, if all FFL bits are 1, the DMA FIFO is full.

Chip Test 2 (CTEST2)
Address 16 Read Only

	RES	RES	SOFF	SFP	DFP	TEOP	DREQ	RES
	7	6	5	4	3	2	1	0
Default >>>	0	0	1	0	0	0	0	1

Bit 7 RES (Reserved)

Bit 6 RES (Reserved)

Bit 5 SOFF (SCSI Offset Compare)

If the SIOP is an initiator, this bit will be 1 whenever the SCSI Synchronous offset counter is equal to zero. If the SIOP is a target, this bit will be 1 whenever the SCSI Synchronous offset counter is equal to the maximum synchronous offset defined in the SXFER register.

Bit 4 SFP (SCSI FIFO Parity bit)

This bit represents the parity bit of the SCSI Synchronous FIFO corresponding to data read out of the FIFO. Reading the CTEST3 register unloads a data byte from the bottom of the SCSI synchronous FIFO. When the CTEST3 register is read, the data parity bit is latched into this bit location.

Bit 3 DFP (DMA FIFO Parity bit)

This bit represents the parity bit of the DMA FIFO. Reading the CTEST6 register unloads one data byte from the bottom of the DMA FIFO. When the CTEST6 register is read the parity signal is latched into this bit location and the next byte falls down to the bottom of the FIFO.

Bit 2 TEOP (SCSI True End Of Process)

This bit indicates the status of the SIOP's internal TEOP signal. The TEOP signal acknowledges the completion of a transfer through the SCSI portion of the SIOP. When this bit is 1, TEOP is active. When this bit is 0, TEOP is inactive.

Bit 1 DREQ (Data Request Status)

This bit indicates the status of the SIOP's internal Data Request signal (DREQ). When this bit is 1, DREQ is active. When this bit is 0, DREQ is inactive.

Bit 0 RES (Reserved)

Chip Test 3 (CTEST3)
Address 17 Read Only

SF7	SF6	SF5	SF4	SF3	SF2	SF1	SF0
7	6	5	4	3	2	1	0
Default >>>							
0	0	0	0	0	0	0	0

Bit 7 SF7 (SCSI FIFO – Bit 7)

Bit 6 SF6 (SCSI FIFO – Bit 6)

Bit 5 SF5 (SCSI FIFO – Bit 5)

Bit 4 SF4 (SCSI FIFO – Bit 4)

Bit 3 SF3 (SCSI FIFO – Bit 3)

Bit 2 SF2 (SCSI FIFO – Bit 2)

Bit 1 SF1 (SCSI FIFO – Bit 1)

Bit 0 SF0 (SCSI FIFO – Bit 0)

Reading this register unloads the bottom byte of the eight-deep SCSI Synchronous FIFO. Reading this register also latches the parity bit for the FIFO into the SCSI FIFO Parity bit (CTEST2). The FIFO Full Bits in the SSTAT2 register can be read to determine how many bytes currently reside in the SCSI Synchronous FIFO.

Note: Reading this register when the SCSI FIFO is empty causes a SCSI Gross Error (FIFO underflow).

If this register is read with the CTEST2 register (16 or 32 bit read), bit 2 of this register may get corrupted reading back the contents of the SCSI FIFO. To guarantee that this does not happen, do an eight bit read of this register.

Chip Test 4 (CTEST4)
Address 18 Read/Write

RES	ZMOD	SZM	SLBE	SFWR	FBL2	FBL1	FBL0
7	6	5	4	3	2	1	0

Default >>>

0	0	0	0	0	0	0	0

Bit 7 RES (Reserved)

Bit 6 ZMOD (Z Mode - High-Impedance Mode)

Writing this bit to 1 causes the SIOP to place all outputs into the high-impedance state. In order to read data out of the SIOP, this bit must be reset to 0. Reset this bit or do a software reset to disable the high-impedance mode. Since all outputs are tri-stated, this may cause problems in a systems application.

Bit 5 SZM (SCSI Z Mode - SCSI High-Impedance Mode)

Setting this bit to 1 causes the SIOP to place SCSI outputs in a high-impedance state. The following outputs will be in a high-impedance state: SD7-SD0, SDP, BSY/, SEL/, RST/, REQ/, C/D, I/O, MSG/, ACK/, ATN/. The direction control lines (SDIR7-SDIR0, SDIRP, BSYDIR, RSTDIR, and SELDIR) are deasserted low and will not be in a high-impedance state. In order to transfer data on the SCSI bus, this bit must be written to 0.

Bit 4 SLBE (SCSI Loopback Enable)

Setting this bit to 1 enables "Loopback" Mode. Loopback allows the user to assert any SCSI signal. The SIOP may be an initiator or a target. It also allows the SIOP to transfer data from the SODL register back into the SIDL register. For a complete description of the tests that can be performed in loopback mode, please refer to section 6.2, Loopback Mode.

Bit 3 SFWR (SCSI FIFO Write Enable)

Setting this bit to 1 redirects data from the SODL to the SCSI FIFO. A write to the SODL register loads a byte into the SCSI FIFO. The parity bit loaded into the FIFO will be odd or even parity depending on the status of the Assert SCSI Even Parity bit in the SCNTL1 register. Resetting this bit will disable this feature.

Bit 2 FBL2 (FIFO Byte Control bit 2)

Bit 1 FBL1 (FIFO Byte Control bit 1)

Bit 0 FBL0 (FIFO Byte Control bit 0)

Byte Lane Selection for 32-bit DMA FIFO

FBL2	FBL1	FBL0	Byte Lane
0	X	X	DMA FIFO Byte Lane 0
1	0	0	DMA FIFO Byte Lane 0
1	0	1	DMA FIFO Byte Lane 1
1	1	0	DMA FIFO Byte Lane 2
1	1	1	DMA FIFO Byte Lane 3

These bits send the contents of the CTEST6 register to the appropriate byte lane of the 32-bit DMA FIFO. If the FBL2 bit is written to 1, then FBL1 & FBL0 determine which of four byte lanes can be read or written. Each of the four bytes that make up the 32-bit DMA FIFO can be accessed by writing these bits to the proper value. For normal operation, FBL2 must equal 0.

Chip Test 5 (CTEST5)
Address 19 Read/Write

	ADCK	BBCK	ROFF	MASR	DDIR	EOP	DREQ	DACK
	7	6	5	4	3	2	1	0
Default >>>								
	0	0	0	0	0	0	0	0

Bit 7 ADCK (Clock Address Incrementor)

Setting this bit to 1 increments the address pointer contained in the DNAD register. The DNAD register is incremented based on the DNAD contents, the bus width, and the host mode used (286 or 386 mode).

In 386 mode with a 16-bit bus width or in 286 mode, DNAD is incremented to a 16-bit word boundary. For example, if DNAD contains an even address (0, 2, 4, ...), it will increment by 2. In 386 mode with a 32-bit bus, it is incremented to a 32-bit longword boundary. For example, if the DNAD register contains a longword boundary address (0, 4, 8, ...), it will increment by four.

This bit automatically clears itself after incrementing the DNAD register.

Bit 6 BBCK (Clock Byte Counter)

Setting this bit to 1 decrements the byte counter contained in the DBC register. The DBC register is decremented based on the DBC contents, whether the current address is on a 16-bit word or a 32-bit longword boundary (DNAD contents) and the bus width. It will always decrement by 1, 2, 3, or 4.

This bit automatically clears itself after decrementing the DBC register.

Bit 5 ROFF (Reset SCSI Offset)

Setting this bit to 1 clears the current offset pointer in the SCSI synchronous offset counter (SSTAT2 bits 7..4). This bit is set to 1 if a SCSI Gross Error condition occurs. The offset should be cleared when a synchronous transfer does not complete successfully. This bit automatically clears itself after clearing the synchronous offset.

Bit 4 MASR (Master Control for Set or Reset pulses)

This controls the operation of bits 3 - 0. When set to 1, bits 3 - 0 assert the corresponding signals. When this bit is reset to 0, bits 3 - 0 deassert the corresponding signals.

Bit 3 DDIR (DMA Direction)

Setting this bit either asserts or deasserts the internal DMAWR direction signal depending on the current status of the MASR bit in this register. Asserting the DMAWR signal indicates that data will be transferred from the SCSI bus to the host bus. Deasserting the DMAWR signal transfers data from the host bus to the SCSI bus.

Bit 2 EOP (End of Process)

Setting this bit either asserts or deasserts the internal EOP control signal depending on the current status of the MASR bit in this register. The internal EOP signal is an output from the DMA portion of the SIOP to the SCSI portion of the SIOP. Asserting the EOP signal indicates that the last data byte has been transferred between the two portions of the chip. Deasserting the EOP signal indicates that the last data byte has not been transferred between the two portions of the chip. If the MASR bit is configured to assert this signal, this bit automatically clears itself after pulsing the EOP signal.

Bit 1 DREQ (Data Request)

Setting this bit either asserts or deasserts the internal DREQ (data request signal) depending on the current status of the MASR bit in this register. Asserting the DREQ signal indicates that the SCSI portion of the SIOP requests a data transfer with the DMA portion of the chip. Deasserting the DREQ signal indicates that data should not be transferred between the SCSI portion of the SIOP and the DMA portion. If the MASR bit is configured to assert this signal, this bit automatically clears itself after asserting the DREQ signal.

Bit 0 DACK (Data Acknowledge)

Setting this bit either asserts or deasserts the internal DACK/ data request signal dependent on the current status of the MASR bit in this register. Asserting the DACK/ signal indicates that the DMA portion of the SIOP acknowledges a data transfer with the SCSI portion of the chip. Deasserting the DACK/ signal indicates that data should not be transferred between the DMA portion of the SIOP and the SCSI portion. If the MASR bit is configured to assert this signal, this bit automatically clears itself after asserting the DACK/ signal.

Chip Test 6 (CTEST6)
Address 1A Read/Write

	DF7	DF6	DF5	DF4	DF3	DF2	DF1	DF0
	7	6	5	4	3	2	1	0
Default >>>	0	0	0	0	0	0	0	0

Bit 7 DF7 (DMA FIFO – Bit 7)

Bit 6 DF6 (DMA FIFO – Bit 6)

Bit 5 DF5 (DMA FIFO – Bit 5)

Bit 4 DF4 (DMA FIFO – Bit 4)

Bit 3 DF3 (DMA FIFO – Bit 3)

Bit 2 DF2 (DMA FIFO – Bit 2)

Bit 1 DF1 (DMA FIFO – Bit 1)

Bit 0 DF0 (DMA FIFO – Bit 0)

A write to this register writes data to the appropriate byte lane of the DMA FIFO as determined by the FBL bits in the CTEST4 register. Reading this register unloads data from the appropriate byte lane of the DMA FIFO as determined by the FBL bits in the CTEST4 register. Data written to the FIFO is loaded into the top of the FIFO. Data is read out of the FIFO from the bottom. When data is read from the DMA FIFO, the parity bit for that byte is latched and stored in the DMA FIFO parity bit in the CTEST2 register.

Do not read or write to this register before starting or restarting a SCSI SCRIPTS. If the DMA FIFO is not cleared before executing a SCRIPTS Block Move instruction, data left in the FIFO will be transferred.

Chip Test 7 (CTEST7)
Address 1B Read/Write

RES 7	RES 6	FM 5	STD 4	DFP 3	EVP 2	DC 1	DIFF 0
0	0	0	0	0	0	0	0

Default >>>

Bit 7 RES (Reserved)

Bit 6 RES (Reserved)

Bit 5 RES (Reserved) 53C700 only

Bit 5 FM (Fetch Mode) 53C700-66 only

This bit, in conjunction with the Fetch/Master enable bit will modify the function of the Fetch output. When the Fetch/Master enable is set and this bit is set, the Fetch/ pin will deassert during indirect read operations. Fetch/ will only be active during the opcode portion of an instruction fetch. This bit will not become active until the Fetch/Master enable is asserted, causing the read back capability to be disabled.

Bit 4 STD (Selection Time-out Disable Bit)

This bit disables the selection time-out timer. A selection or reselection time-out will not be reported when this bit is set.

Bit 3 DFP (DMA FIFO Parity bit)

This bit represents the parity bit of the DMA FIFO when reading data out of the DMA FIFO via programmed I/O (CTEST6). In order to transfer data to/from the DMA FIFO, perform a read or a write to the CTEST6 register. When loading data into the FIFO, write this bit to the FIFO as the parity bit for each byte loaded. Set this bit with the status of the parity bit to be written to the FIFO before writing the byte to the FIFO. For details of performing a diagnostic test of the DMA FIFO, please refer to the Diagnostics DMA FIFO Test section of this manual.

Bit 2 EVP (Even Parity)

Setting this bit to 1 causes the SIOP to generate even parity when sending data to the host bus. It will also check the host bus for even parity if parity checking is enabled (SCNTL0, bit 3). Setting this bit to 0 causes odd parity to be generated on the host side.

Bit 1 DC (DC/ Output Signal Low For Instruction Fetches)

Setting this bit to 1 causes the SIOP to drive the DC/ signal low when fetching SCRIPTS instructions from memory. This allows the user the option of storing SIOP instructions in a cache or forcing them to be read directly out of memory. However, the DC/ signal does not become valid until HOLDAI has been received and the SIOP has started the first bus master cycle.

When this bit is reset to 0, the DC/ signal is driven high. The DC/ signal is always driven high when moving data to/from memory and can only be driven low during instruction fetch cycles.

Bit 0 DIFF (Differential Mode)

Setting this bit to 1 enables the SIOP to interface with external differential pair transceivers. The function of the SCSI BSY/, SEL/ and RST/, is different for differential mode. For more information on differences between the two modes, refer to the pin descriptions for these signals in Chapter 3. Resetting this bit enables single-ended mode. This bit should be set to 1 during initialization if the differential pair interface is to be used.

Temporary Stack (TEMP)
Address 1C-1F Read/Write

Default >>> all zeros

This 32-bit register stores the instruction address pointer for a CALL. The address pointer stored in this register is loaded into the DSP register upon execution of a RETURN. This address points to the next instruction to be executed. Do not write to the TEMP register while the SIOP is executing SCSI SCRIPTS.

DMA FIFO (DFIFO)
Address 20 Read/Write

FLF 7	CLF 6	BO5 5	BO4 4	BO3 3	BO2 2	BO1 1	BO0 0
0	0	0	0	0	0	0	0

Default >>>

The DFIFO register can only be accessed by 8-bit reads or writes. A 16- or 32-bit read or write of this register would also include the ISTAT register. To protect the SIOP from any internal bus contentions while executing SCRIPTS, any other registers accessed during a read or a write of the ISTAT register will be disabled and appear as FF.

Bit 7 FLF (Flush DMA FIFO)

When set to 1, data residing in the DMA FIFO is transferred to memory starting at the address in the DNAD register. Once the SIOP has successfully transferred the data (DSTAT, bit 7 equals one), this bit should be written to 0.

Bit 6 CLF (Clear DMA and SCSI FIFOs)

When set to 1, the SCSI and DMA FIFO pointers are cleared. In addition, the SIDL, SODL, and SODR full bits in the SSTAT1 register are reset to 0. This bit automatically resets to 0 after the SIOP has successfully cleared the appropriate FIFO pointers and registers .

Bits 5-0 BO5-0 (Bits 5-0 FIFO Byte Offset Counter)

These six bits indicate the amount of data transferred between the SCSI core and the DMA core. Use it to determine the number of bytes in the DMA FIFO when an error occurs. These bits will change when data is transferred between the two cores. Once the chip has stopped transferring data, these bits are stable.

Data may remain in the chip when an interrupt occurs in the middle of a SCRIPTS Block Move instruction. The most common circumstance is a disconnect by a target device in the middle of a block move. To determine how many bytes reside in the DMA FIFO, perform the following steps:

When sending SCSI data,

1. Read this DFIFO register
2. Mask the upper 2 bits by ANDing with 3F hex
3. Read the lower 8 bits of the DBC register
4. Mask the upper 2 bits by ANDing with 3F hex
5. Subtract the 6-bit value of the DBC register from the 6-bit value of the DFIFO register
6. Mask any carry bits by ANDing the result with 3F hex
7. The final result will be between 0 and 32 bytes

When receiving SCSI data,

1. Read the lower 8 bits of the DBC register
2. Mask the upper 2 bits by ANDing with 3F hex
3. Read the DFIFO register
4. Mask the upper 2 bits by ANDing with 3F hex
5. Subtract the 6-bit value of the DFIFO register from the 6-bit value of the DBC register
6. Mask any carry bits by ANDing the result with 3F hex
7. The final result will be between 0 and 32 bytes

If a SCSI interrupt occurs when receiving data, the data left in the DMA FIFO will be transferred to host memory. There should be no bytes left to recover in the FIFO. If, however, a DMA interrupt (DMA Watchdog Interrupt) occurs or some catastrophic error on the host side, and the chip is unable to continue the data transfer to host memory, there may be bytes remaining in the FIFO.

Interrupt Status (ISTAT)
Address 21 Read/Write

ABRT 7	RES 6	RES 5	RES 4	CON 3	PRE 2	SIP 1	DIP 0
Default >>>							
0	0	0	0	0	1	0	0

This is the only register in the SIOP that can be accessed while fetching and executing SCRIPTS.

It can be read or written at any time without interfering with SCRIPTS operation of the SIOP. Use it to poll for interrupts if hardware interrupts are not enabled.

Multiple DMA and/or SCSI interrupts may occur simultaneously. This will cause more then one interrupt status bit to be set in DSTAT or SSTAT0. There may also be stacked interrupts which occur one immediately following another. To check for stacked interrupts, read this register after clearing an interrupt. The IRQ signal will be asserted for each stacked interrupt that occurs.

The typical sequence for servicing interrupts is shown in the following:

1. Read ISTAT
2. If the SIP bit is set, read SSTAT0 to clear the bit and get the interrupt status.
3. If the DIP bit is set, read DSTAT to clear the bit and get the interrupt status.
4. If the SIP and DIP bits are both set, read SSTAT0 and DSTAT as a word to clear the bits and get the interrupt status. If executing 8-bit reads of the DSTAT and SSTAT0 registers to clear interrupts, insert 10 CLKs (one or two NOPs) between the consecutive reads to ensure that the interrupt clears properly.
5. Repeat this sequence until no interrupts are pending in ISTAT.

To protect the SIOP from any internal bus contention while executing SCRIPTS, any other registers accessed during a read or write of this register will be disabled and appear as FFs. For example, a 32-bit read of address 20 will include Chip Test 9 and Chip Test 8 registers (23h and 22h), ISTAT register (21h) and the DFIFO register (20h). The ISTAT register will read valid data, but the DFIFO, CTEST8 and CTEST9 registers will show FFs. Therefore, the DFIFO register can only be accessed by an 8-bit read or write and the CTEST8 and CTEST9 registers can be accessed by either an 8-bit or 16-bit read or write.

Bit 7 ABRT (Abort Operation)

This bit is set to 1 to abort the current operation being executed by the SIOP. If this bit is set to 1 and the DMA interrupt is received, reset this bit 0 before clearing the interrupt (reading the DSTAT register) to prevent further Abort interrupts from being generated. The sequence to abort is described below.

1. Write this bit to 1.
2. Wait for an interrupt.
3. Read the ISTAT register
4. If the DMA Interrupt Pending bit is 1, then write 00h value to this register.
5. Read the DSTAT register to verify the abort interrupt was received and to see if any other interrupting conditions have occurred.
6. If executing a software abort in a multi-threaded environment, the chip can be selected or reselected while the abort is executing. See "Abort Example" in Chapter 6.

Bits 6-4 RES (Reserved)

Bit 3 CON (Connect/disconnect)

This status bit is set to 1 when the SIOP has won arbitration on the SCSI bus. It is reset to 0 when the SIOP is disconnected from the SCSI bus.

This bit is automatically set anytime the SIOP becomes connected as an initiator or a target. Connected is defined by winning arbitration on the SCSI bus or when the SIOP has responded to a selection or reselection.

Bit 2 PRE (Pointer Register Empty)

This status bit is set to 1 when the DSPS and DSP registers are empty. In pipeline mode, poll this register to determine when the SIOP is ready to accept another instruction. This bit is always set unless using pipeline mode.

Bit 1 SIP (SCSI Interrupt Pending)

This status bit is set to 1 when an interrupt condition is detected in the SCSI portion of the SIOP. To determine which condition(s) have occurred, read the SSTAT0 register. It indicates that one of the following SCSI interrupt conditions has occurred.

1. Phase Mismatch (Initiator Mode) or ATN/ active (Target Mode)
2. Function Complete
3. Selection or Reselection Time-out occurred
4. The SIOP was selected or reselected
5. SCSI Gross Error occurred
6. Unexpected Disconnect occurred
7. SCSI Reset detected active
8. Parity Error received

Bit 0 DIP (DMA Interrupt Pending)

This status bit is set to 1 when an interrupt condition is detected in the DMA portion of the SIOP. To determine which condition(s) have occurred, read the DSTAT register. It indicates that one of the following DMA interrupt conditions has occurred.

1. Abort condition detected
2. SCRIPTS single step interrupt received
3. SCSI SCRIPTS Interrupt instruction
4. Watchdog timer counter decremented to zero, indicating that a host memory time-out occurred
5. Illegal SCRIPTS instruction detected

Note: If executing 8-bit reads of the DSTAT and SSTAT0 registers to clear interrupts, insert one or two NOPs (10 CLKs) between the consecutive reads of the DSTAT or SSTAT0 registers to ensure that the interrupt clears properly. For example:

1. Read DSTAT (to clear the DMA interrupt)
2. Read ISTAT (10 CLKs for one slave mode, register read cycle)
3. Read SSTAT0 to clear the SCSI interrupt.

Chip Test 8 (CTEST8)
Address 22 Read/Write

EAS	EFM	GRP	EAN	HSC	SRA	DAS	LDE
7	6	5	4	3	2	1	0

Default >>>

0	0	0	0	0	0	0	0

This register is available only in the 53C700-66 chip

Bit 7 EAS (Enable Alternate SCSI Clock)

When set, (SCLK/ pin 95) is used as an alternate SCSI clock to the SCSI core. This feature allows the use of different clock frequencies for the DMA and SCSI cores. After reset or when the bit is cleared, the CLK pin is used for both the DMA and SCSI core.

Bit 6 EFM (Enable Fetch and Master Outputs)

When set, FETCH/ (pin 94) will be driven low to indicate that the 53C700-66 is fetching an opcode. When the 53C700-66 becomes a bus master MASTER/ (pin 93) will be driven low. After reset or when the bit is cleared, the FETCH/ and MASTER/ outputs will be disabled. See bit 5 of CTEST7 for additional information.

Bit 5 GRP (Generate Receive Parity for Pass Through)

When set, and the 53C700-66 is in parity pass through mode, parity received on the SCSI bus will not pass through to the DMA FIFO. Parity will be generated as data enters the DMA FIFO eliminating the possibility of bad SCSI parity passing through to the host bus. A SCSI parity error interrupt will be generated but a system parity problem will not be created. After reset or when the bit is cleared, and parity pass through mode is enabled, parity received on the SCSI bus will pass through the 53C700-66 unmodified.

Bit 4 EAN (Enable Active Negation)

Asserting this bit causes the SCSI Request, Acknowledge, Data, and Parity to be actively deasserted, in addition to relying on external pull-ups, when the 53C700-66 is driving these signals. Active deassertion of these signals will occur only when the 53C700-66 is in an information transfer phase. When operating in a differential environment or at fast SCSI timings, Active Negation should be enabled to improved setup and hold times. After reset or when the bit is cleared, Active Negation is disabled.

Bit 3 HSC (Halt SCSI Clock)

Asserting this bit causes the internal divided SCSI clock to come to a stop in a glitchless manner. This bit may be used for test purposes or to lower IDD during a power down mode. Note: SCSI registers must be reinitialized at power-up.

Bit 2 SRA (Shorten REQ/ACK Filtering)

The SCSI core contains a special digital filter on the REQ/ and ACK/ pins which will cause glitches on deasserting edges to be disregarded. Asserting this bit will provide less filtering on the deasserting edge of the REQ/ and ACK/ signals. Note: This bit must be set during fast SCSI (>5 M transfers per second) operations, if not set, a valid assertion could be treated as a glitch.

Bit 1 DAS (Disable Auto Switching)

This feature will allow the user to disable the automatic switch from initiator to target or from target to initiator in the existing SIOP, eliminating the possibility of the user not recognizing that the SIOP has changed modes.

Bit 0 LDE (Last Disconnect Enable)

If this bit is set, the status of a pending SCSI disconnect is maintained by the SCSI core, eliminating the possibility of not recognizing a selection or reselection while waiting to fetch a WAIT DISCONNECT opcode.

Chip Test 9 (CTEST9)
Address 23 Read Only

VER 7	VER 6	VER 5	VER 4	VER 3	VER 2	VER 1	VER 0
Default >>>							
1	1	1	1	1	1	1	1
	or						
0	0	0	0	0	0	0	0

This register is available only in the 53C700-66 chip

Bits 7-0 Version Control

This register determines the 53C700 product type. All ones (FF) indicates this chip is the 53C700 standard part. All other values indicate this chip is the 53C700-66 version. All zeros indicate this chip is the first version of the 53C700-66.

Note: To identify a 53C700 or a 53C700-66 part perform the following steps:

1. Write ones (FF) to CTEST9 to precharge the internal data bus
2. Read CTEST9
3. If the contents are all ones (FF), the chip is a 53C700 standard part
4. If the contents are all zeros, the chip is the first version of the 53C700-66

DMA Byte Counter (DBC)
Address 24-26 Read/Write

Default >>> all zeros

This 24-bit register indicates the number of bytes to be transferred in a Block Move instruction. While sending data to the SCSI bus, the counter is decremented as data is moved into the DMA FIFO from memory. While receiving data from the SCSI bus, the counter is decremented as data is written to memory from the SIOP. The DBC counter is decremented each time that the ADS/ signal is pulsed by the SIOP. It is decremented by an amount equal to the number of bytes that were transferred.

The maximum number of bytes that can be transferred in any one Block Move command is 16,777,215 bytes. The maximum value that can be loaded into the DBC register is FFFFFFh. If the instruction is Block Move and a value of 000000h is loaded into the DBC register, an illegal instruction interrupt will occur.

DMA Command (DCMD)
Address 27 Read/Write

Default >>> all zeros

This contains the 8-bit opcode of a SCRIPTS instruction that has been fetched. The opcode is the first 8-bit field of a 64-bit SCRIPTS instruction. For a complete description of the opcodes, please refer to the instruction set of the SIOP in Chapter 5.

DMA Next Address for Data (DNAD)
Address 28-2B Read/Write

Default >>> all zeros

This 32-bit register contains the second longword of a SCRIPTS Block Move instruction. Block Move instructions use this register to point to the address where data is to be moved. For a select, reselect, jump, call or return instruction, it contains a copy from the DSPS register of the second longword . This register should not be read or written while executing SCSI SCRIPTS.

DMA SCRIPTS Pointer (DSP)
Address 2C-2F Read/Write

Default >>> all zeros

To execute SCSI SCRIPTS, the address of the first SCSI SCRIPTS instruction should be written to this register. In normal SCRIPTS operation, once the start address of the SCSI SCRIPTS is written to this register, the SCRIPTS instructions are automatically fetched and executed until an interrupt condition occurs. The DSP register is incremented immediately after the current instruction is fetched. The register will therefore point to the next instruction while the current instruction is executing.

In single step mode, there is a SCRIPTS single step interrupt after each instruction is executed. After the first instruction is executed, the DSP register does not need to be written with the next address, but the Start DMA bit (bit 2, DCNTL register) must be set each time the single step interrupt occurs to fetch and execute the next SCRIPTS instruction.

In pipeline mode (DCNTL register, bit 1), this register becomes the DCMD and DBC register.

The write to the upper byte starts a SCRIPTS instruction fetch. When writing this register 8-bits at a time or 16 bits at a time, the upper byte should be written last.

DMA SCRIPTS Pointer Save (DSPS)
Address 30-33 Read/Write

Default >>> all zeros

This 32-bit register contains the second longword of a Select, Reselect, Jump, Call, Return, or Interrupt SCRIPTS instruction that has been fetched. When executing pipelined instructions (Pipeline mode, DMODE register, bit 1) this register should be loaded with the second longword of the pipelined command. It should not be read or written while executing SCSI SCRIPTS.

DMA Mode (DMODE)
Address 34 Read/Write

BL1 7	BL0 6	BW16 5	286 4	IO/M 3	FAM 2	PIPE 1	MAN 0
Default >>>							
0	0	0	0	0	0	0	0

Bit 7 BL1 (Burst Length Bit 1)

Bit 6 BL0 (Burst Length Bit 0)

Burst Lengths Transferred across the 80386 Interface

BL1	BL0	Burst Length
0	0	1 Transfer
0	1	2 Transfers
1	0	4 Transfers
1	1	8 Transfers

These two control bits determine the maximum data burst length transferred across the 80386 interface. The actual number of bytes transferred across the bus is equal to the host bus width (16-bit or 32-bit) times the burst length.

Once the SIOP has won control of the host bus, it will stay on the bus until the data burst is complete.

Bit 5 BW16 (Host Bus Width Equal to 16 bits)

When set to 1, during Block Move instructions the SIOP transfers data 16-bits at a time. This allows the SIOP to operate with 16-bit memory. This bit does NOT cause SCSI SCRIPTS to be loaded 16-bits at a time. The SCSI SCRIPTS 16 bit in the DCNTL register controls how SCSI SCRIPTS are loaded.

Bit 4 286 (286 Mode)

When set to 1, the SIOP operates in 80286 mode. It will connect directly to the Intel 80286 microprocessor. The following signals change function:

BE2/ becomes BHE/,

BE1/ becomes A1, and

BE0/ becomes A0.

Block Move instructions transfer data 16-bits at a time and SCRIPTS instructions are fetched 16-bits at a time. Initialize this bit before reading or writing any other register if the SIOP needs to operate in 80286 mode. This bit must be set again any time a software reset is issued (DCNTL, bit 0).

Bit 3 IO/M (I/O Mapped or Memory Mapped)

This bit determines if data is to be transferred to/from a memory-mapped address or an I/O-mapped address when the SIOP becomes a bus master. Writing this bit to 1 will drive the MIO/ signal low, transferring data to an I/O-mapped device. Writing this bit to 0 will drive the MIO/ signal high, transferring data to a memory-mapped device. This bit does not have an effect on instruction fetch operations, it only applies to data being transferred to/from memory. This bit has no affect on how the SIOP's addresses are mapped, this is determined by external address decode logic.

Bit 2 FAM (Fixed Address Mode)

Writing this bit to 1 disables the DMA next address pointer from incrementing after each data transfer. The DMA next address pointer is located in the DNAD register. Use fixed addressing to transfer data to/from one port address, i.e. a serial port. If this bit is 0, the next address pointer increments after each data transfer.

Bit 1 PIPE (Pipeline Mode *)

Setting this bit to 1 disables the automatic fetch and execution of SCSI SCRIPTS™ from memory. In this mode, the DSP and DSPS registers have different functions. The DSP register operates as the first 32-bit word of a pipelined instruction. The DSPS register operates as the second 32-bit word of a pipelined instruction. The execution of pipelined commands are as follows:

1. Write this pipeline mode bit to 1.
2. Load the DSPS register with the second 32-bit word of the instruction.

**3. Load the DSP register with the first 32-bit word of the instruction.

4. Write the start DMA bit (DCNTL register, bit 2).
5. Poll the Pipeline register Empty bit in the ISTAT register until it is 1.
6. Load the DSPS register with the second 32-bit word of the next instruction.
7. Load the DSP register with the first 32-bit word of the next instruction.
8. Go to step 4.

* Pipeline Mode will not be offered in the next generation of the 53C700 family.

** If the DSP is not written in a single 32-bit cycle, the high byte (or word) must be written last.

Bit 0 MAN (Manual Start Mode)

Writing this bit to 1 disables the SIOP from automatically fetching and executing SCSI SCRIPTS after the DSP register is written. For this case, the Start DMA bit in the DCNTL register must be set to 1 for the SIOP to start fetching and executing instructions. Writing this bit to 0 causes the SIOP to automatically fetch and execute SCSI SCRIPTS after the DSP Register is written.

DMA Interrupt Enable (DIEN)
Register 39 Read/Write

RES	RES	RES	ABRT	SSI	SIR	WTD	OPC
7	6	5	4	3	2	1	0

Default >>>

0	0	0	0	0	0	0	0

Bits 7-5 RES (Reserved)

Bit 4 ABRT (Enable Aborted Interrupt)

Writing this bit to 1 asserts the IRQ/ signal on an abort condition. Abort conditions can occur in two ways: the DP3_ABRT/ input signal is asserted or a software abort command is issued by writing 1 to Bit 7 of the ISTAT register. Writing 0 to this bit disables the assertion of IRQ/ when an abort condition occurs.

Bit 3 SSI (Enable SCRIPTS Single Step Interrupt)

Writing this bit to 1 asserts the IRQ/ signal when the SCRIPTS Single Step Interrupt occurs. Resetting this bit to 0 disables the assertion of IRQ/ when a SCRIPTS Single Step Interrupt condition occurs. The following conditions cause a Single Step interrupt .

1. If the Single Step Mode bit in the DCNTL register is equal to 1, then there will be a SCRIPTS Single Step Interrupt after executing each instruction.
2. If the SIOP encounters a branch condition while executing pipelined instructions (Pipeline mode, DMODE register, bit 1).

Bit 2 SIR (Enable SCRIPTS Interrupt Instruction Received Interrupt)

Writing this bit to 1 asserts the IRQ/ signal when the SCRIPTS Interrupt Instruction Received bit is set to 1 in the DSTAT register. The SCRIPTS Interrupt Instruction Received status bit is set when an interrupt instruction occurs during execution of SCSI SCRIPTS. Writing 0 to this bit disables the assertion of IRQ/ when a SCRIPTS Interrupt instruction is received.

Bit 1 WTD (Enable Watchdog Time-out Interrupt)

Writing this bit to 1 asserts the IRQ/ signal whenever the Watchdog Timer Counter has decremented to zero.

If the counter decrements to zero, it indicates that the memory device did not assert the READYI/ signal within the specified time-out period from the SIOP assertion of ADS/. Resetting this bit to 0 disables the assertion of IRQ/ when a Watchdog Time-out condition occurs.

Bit 0 OPC (Enable Illegal Instruction Interrupt)

Writing this bit to 1 asserts the IRQ/ signal anytime that an illegal instruction is decoded. This bit can be set when the SIOP operates in either SCSI SCRIPTS mode or Single Step mode or pipeline mode. Writing 0 to this bit disables the assertion of IRQ/ when an Illegal Instruction condition occurs.

DMA Watchdog Timer (DWT)
Address 3A Read/Write

Default >>> all zeros

The DMA Watchdog Timer Register provides a time-out mechanism during data transfers between the SIOP and memory. This register determines the amount of time that the SIOP will wait for the assertion of the READYI/ signal after pulsing the ADS/ signal. Write the time-out value to this register during initialization. Every time the SIOP transfers data to/from memory, the value stored in this register is loaded into the counter. Disable the time-out feature by writing a 00h to this register.

The unit time base for this register is 16 CLK input periods. For example, at 50 MHz (clock period = 20 nsec), the time base for this register is 16 x 20 nsec = 320 nsec. If a time-out of 50 µsec was desired, then at 50 MHz this register should be loaded with a value of 9D hex.

DMA Control (DCNTL)
Address 3B Read/Write

CF1	CF0	S16	SSM	LLM	STD	RES	RST
7	6	5	4	3	2	1	0

Default >>>

0	0	0	0	0	0	0	0

Bit 7 CF1 (Clock Frequency bit 1)

Bit 6 CF0 (Clock Frequency bit 0)

Set these two bits according to the input clock frequency of the SIOP. The following table describes how to program these two bits. It is important that these bits be set to the proper state to guarantee that the SIOP meets the SCSI timings defined by the ANSI specification. The center column of the following tables is an internal divide by for SCSI core clock.

53C700

CF1	CF0	SCSI Core Clock	SCLK Frequency
0	0	SCLK / 2	37.51 - 50.00 MHz
0	1	SCLK / 1.5	25.01 - 37.50 MHz
1	0	SCLK / 1	16.67 - 25.00 MHz
1	1	Reserved	Reserved

53C700-66

CF1	CF0	SCSI Core Clock	SCLK Frequency
0	0	SCLK / 2	37.51 - 50.00 MHz
0	1	SCLK / 1.5	25.01 - 37.50 MHz
1	0	SCLK / 1	16.67 - 25.00 MHz
1	1	SCLK / 3	50.01 - 66.67 MHz

Bit 5 S16 (SCSI SCRIPTS™ Loaded in 16-bit Mode)

When set to 1, SCSI SCRIPTS instructions are fetched 16-bits at a time. SCSI SCRIPTS instruction fetches involve four 16-bit transfers. This bit applies only to SCSI SCRIPTS operations and has no effect on data transfers for Block Move instructions. When set to 0, SCSI SCRIPTS instructions are fetched 32-bits at a time.

Bit 4 SSM (Single Step Mode)

Writing this bit to 1 halts the SIOP after completing each instruction. The SCRIPTS Single Step Interrupt bit in the DSTAT register becomes 1 after each instruction is executed. If the SCRIPTS Single Step interrupt is enabled (DIEN register, bit 3), the IRQ/ signal will be asserted after each instruction is executed.

To (re)start the SIOP in Single Step mode, read the DSTAT register to clear the SCRIPTS Single Step Interrupt and then set the START DMA bit (bit 2) in this register. Continue this for each instruction to be executed.

If this bit is 0, then the SIOP will not stop after each instruction; instead it continues fetching and executing instructions until an interrupt condition occurs. For normal SCSI SCRIPTS operation, this bit should be 0.

Bit 3 LLM (Enable SCSI Low-Level Mode)

Setting this bit to 1 places the SIOP in low-level Mode. This bit can only be set after connecting to the SCSI bus. When selecting or reselecting in the low-level mode, this bit must be set to zero (0). Execute the Arbitration and Selection Modes by writing 1 to the Start Sequence bit as described in the SCNTL0 register. Perform SCSI bus transfers manually by asserting and polling SCSI signals. For more information on low-level programming refer to the NCR 53C700 SCSI I/O Processor Programmer's Guide.

Setting this bit to 0 disables low-level Mode.

Bit 2 STD (Start DMA Operation)

The SIOP fetches a SCSI SCRIPTS instruction from the address contained in the DNAD register when this bit is set to 1. This bit is required if the SIOP is in one of the following modes:

1. Manual Start Mode - Bit 0 in the DMODE register equals 1
2. Single Step Mode - Bit 4 in the DCNTL register equals 1
3. Pipeline Mode - Bit 1 in the DMODE register equals 1

The Start DMA bit needs to be written to 1 to start execution of each instruction. If the SIOP is in Manual Start Mode, Single Step Mode, or Pipeline Mode after the Start DMA bit is set to 1, it should not be written to 1 again until an interrupt occurs.

Bit 1 RES (Reserved)

Bit 0 RST (Software Reset)

Writing this bit to 1 resets the SIOP. All registers are cleared to their default values (except the 286 bit in DMODE) and all SCSI signals are deasserted. Writing this bit to 1 does not cause the SCSI RST/ signal to become asserted. This bit is not self-clearing and must be written to 0 in order to clear the reset condition.

Scratch B (SCRATCHB)
Address 3C-3F Read/Write

Default >>> all zeros

This register is available only in the 53C700-66 chip

This is a general purpose user definable scratch pad register. Normal SCRIPTS operations will not destroy the contents of this register, only slave Read/Writes into the SCRATCH B register will alter its contents. This register can not be accessed by the internal SCSI I/O processor.

Chapter Five
Command Set

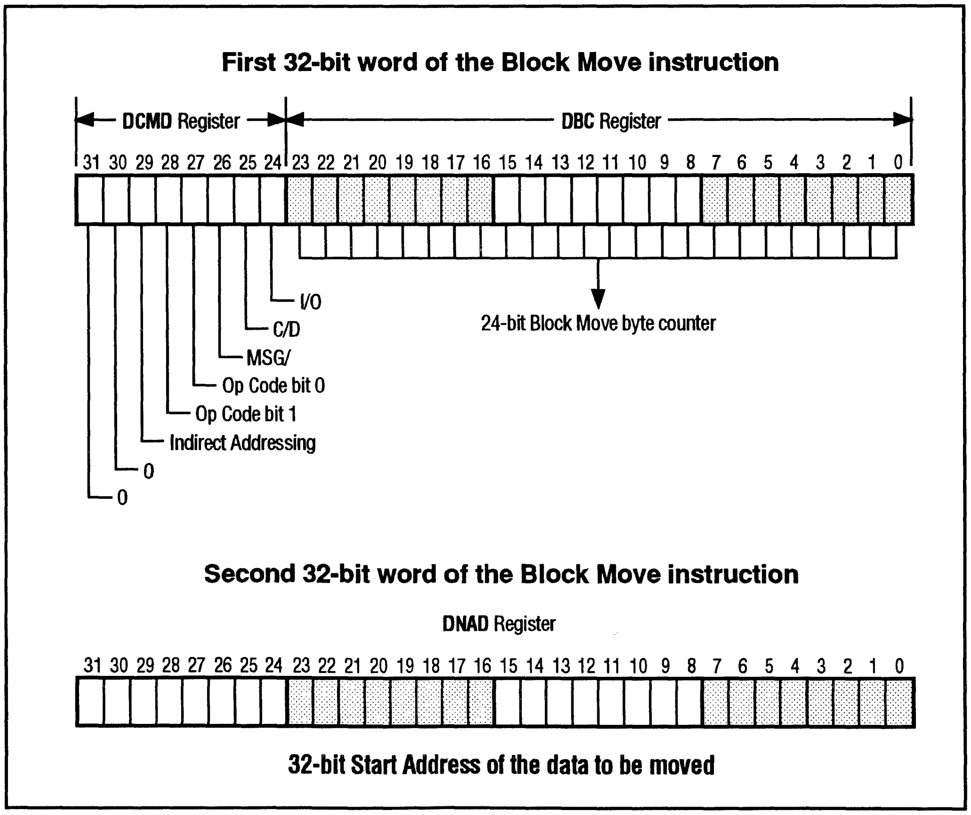
The SCSI I/O processor fetches and executes its own instructions by becoming a bus master and loading two 32-bit longwords; the first 32-bits goes into the DMA Command (DCMD) and the DMA Byte Counter (DBC) registers and the second 32-bits loads into the DMA Next Address for Data (DNAD) or the DMA SCRIPTS Pointer Save (DSPS) register.

The SIOP implements three types of instructions:

- Block Move instructions,
- I/O instructions, and
- Transfer Control instructions.

Block Move Instructions

Figure 5-1. Block Move Instruction Register



Indirect Addressing Field (Bit 29)

When this bit is cleared (set to 0), user data is moved to or from the 32-bit data start address for the Block Move instruction. The value is loaded into the chip's address register and incremented as data is transferred.

When set to 1, the 32-bit user data start address for the Block Move instruction is the address of a pointer to the actual data buffer address. The value at the 32-bit start address is loaded into the chip's DNAD register via a second longword (four-byte transfers across the host computer bus).

This option implies three DMA longword transfers, rather than only two transfers.

Once the data buffer address is loaded, it is executed as if the chip was operating in the direct mode. This indirect feature allows a table of data buffer addresses to be specified. Using the NCR SCSI SCRIPTS compiler, the table offset is placed in the script at compile time. Then at the actual data transfer time, the offsets are added to the base address of the data address table by the external processor. The logical I/O driver builds a structure of addresses for an I/O rather than treating each address individually. This feature makes it possible to locate SCSI SCRIPTS in a PROM.

Opcode Field (Bits 28, 27)

This two-bit field defines the instruction to be executed. The Opcode Field bits have a different meaning depending on whether the SIOP is operating in the initiator or the target mode.

Target Mode

OPC1	OPC0	Instruction Defined
0	0	MOVE – Block Move

Note: Opcode 1, 2, and 3 are Reserved – Illegal Instruction, Interrupt will occur.

MOVE Instruction

1. If the Indirect Addressing bit is 1, the SIOP fetches the starting address from the location pointed to by the DNAD register and stores it in the DNAD register.
2. The SIOP verifies that any previous Perform Reselection command has been completed or that the SIOP has been selected as a target before starting to execute this instruction.
3. The SIOP asserts the SCSI phase signals (MSG/, C/D, & I/O) as defined by the Phase Field bits in the instruction.
4. If the instruction is for the command phase (MSG/ = 0, C/D = 1, & I/O = 1), the SIOP waits for the first command byte to be received and decodes its SCSI Group Code.

   A. If the SCSI Group Code is either Group 0, Group 1, Group 2, or Group 5, then the SIOP overwrites the DBC register with the length of the Command Descriptor Block, 6, 10, or 12 bytes.

   B. If any other Group Code is received, the DBC register is not modified and the SIOP will request the number of bytes specified in the DBC register.

   C. If the Group Code is not one of the Group Codes defined above in "A." and the DBC register contains 000000h, then an Illegal Instruction Interrupt is generated.

5. The SIOP transfers the number of bytes specified in the DBC register starting at the address specified in the DNAD register.
6. If the SCSI ATN/ signal is asserted by the initiator or a parity error occurred during the transfer, the transfer can optionally be halted and an interrupt generated. The Disable Halt on Parity Error or ATN bit in the SXFER register controls whether an interrupt will be generated.

Initiator Mode

OPC1	OPC0	Instruction Defined
0	0	Reserved, DO NOT use
0	1	MOVE – Wait Block Move instruction

Note: Opcodes 2 and 3 are Reserved – An Illegal Instruction, Interrupt will occur.

MOVE Instruction

1. If the Indirect Addressing bit is 1, the SIOP fetches the starting address from the location pointed to by the DNAD register, and stores it in the DNAD register.
2. The SIOP verifies that any previous Perform Selection command has been completed or that the SIOP has been reselected as an initiator before executing this instruction.
3. The SIOP waits for a previously unserviced phase to occur. A previously unserviced phase is defined as any phase with REQ/ asserted. It means that the SIOP has not transferred data for the corresponding phase by responding with an ACK/ to a REQ/ received by the target.
4. The SIOP compares the SCSI phase bits in the DCMD register with the latched SCSI phase lines stored in the SSTAT2 register. These phase lines are latched when REQ/ becomes asserted.
5. If the SCSI phase bits match the value stored in the SSTAT2 register, the SIOP will transfer the number of bytes specified in the DBC register starting at the address pointed to by the DNAD register.
6. If the SCSI phase bits do not match the value stored in the SSTAT2 register, the SIOP generates a Phase Mismatch Interrupt and the command is not executed.
7. If the SCSI phase is Message-In, the SIOP will not deassert the last ACK/ of the transfer. To deassert the last ACK/, a CLEAR ACK instruction should be executed. This allows the host processor to evaluate the Message-In byte(s).
8. During the Message-Out phase, following the assertion of ATN, the SIOP will automatically deassert ATN during the last handshake of the transfer, but before ACK/ is asserted.

Phase Field (Bits 26-24, MSG, C/D, & I/O)

This three-bit field defines the desired SCSI information transfer phase. When the SIOP operates in initiator mode, these bits are compared with the Latched SCSI phase bits in the SSTAT2 register. When the SIOP operates in target mode, the SIOP asserts the phase defined in this field. The following table describes the possible combinations and their corresponding SCSI phase.

Information Transfer Phases

MSG	C/D	I/O	SCSI Phase
0	0	0	Data-Out
0	0	1	Data-In
0	1	0	Command
0	1	1	Status
1	0	0	Reserved for future use
1	0	1	Reserved for future use
1	1	0	Message-Out
1	1	1	Message-In

Key: "0" equals not asserted, "1" equals asserted

Transfer Counter Field (Bits 23-0, DBC register)

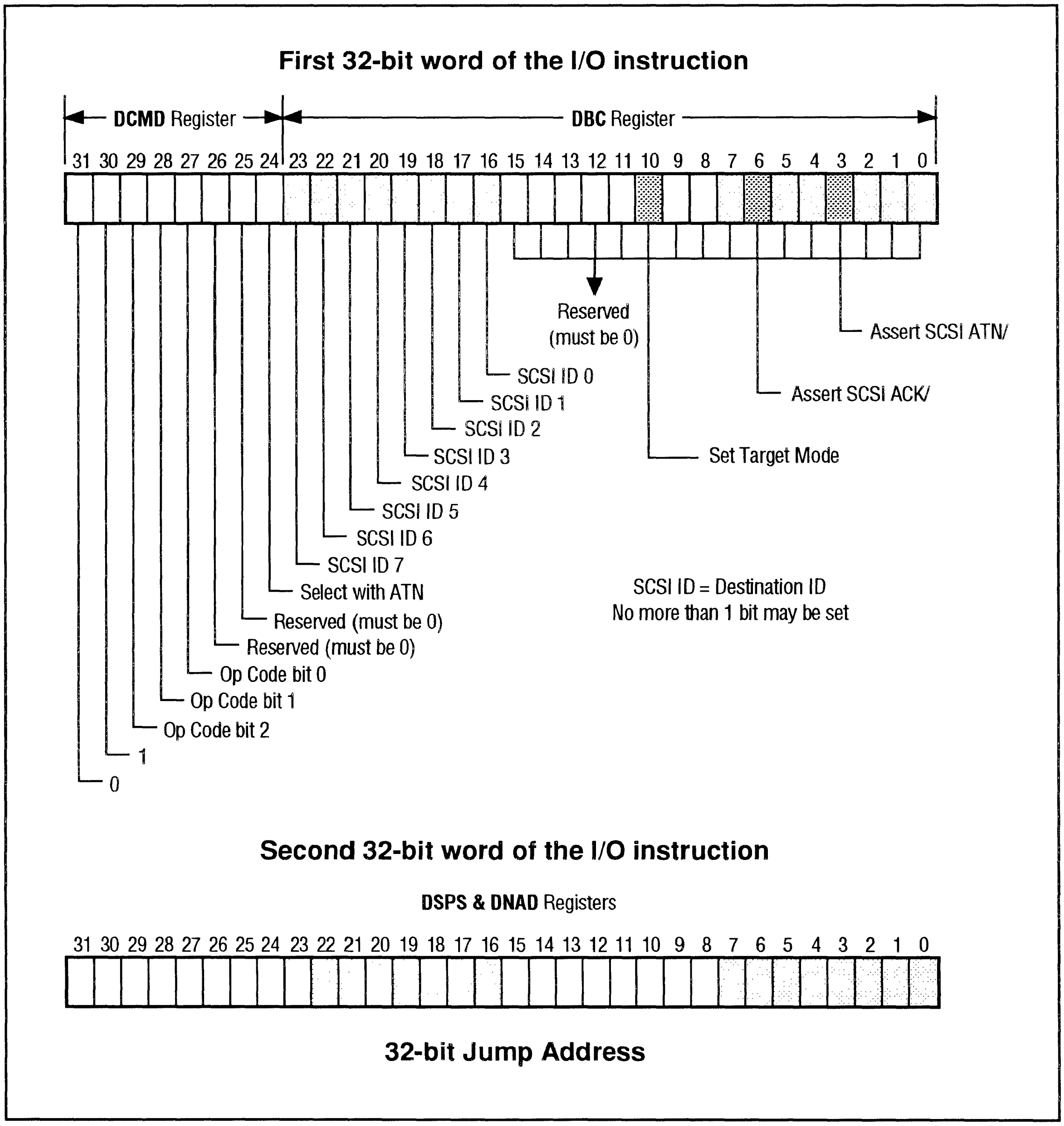
A twenty-four-bit field specifying the number of data bytes to be moved between the SIOP and system memory. The field is stored in the DBC register. When the SIOP transfers data to/from memory, the DBC register is decremented by the number of bytes transferred. In addition, the address in the DNAD register is incremented by the number of bytes transferred. This process is repeated until the DBC register has been decremented to zero. At that time, the SIOP fetches the next instruction. Once the SIOP has started executing SCSI SCRIPTS instructions, do not write to the DBC register.

Start Address Field (Bits 31-0, DNAD register)

This 32-bit field specifies the starting address of the data to be moved to/from memory. The field is stored in the DNAD register. When the SIOP transfers data to/from memory, the DNAD register is incremented by the number of bytes transferred. Once the SIOP has started executing SCSI SCRIPTS instructions, do not write to the DNAD register.

I/O Instructions

Figure 5-2. I/O Instruction Register



Note: In future generations of the SIOP family, the second 32-bit word of the I/O instruction will be loaded into the DSPS only.

Opcode Field (Bits 29, 28, 27)

This three-bit field specifies the event required to occur before continuing execution. The Opcode Field bits have different meanings, dependent on whether the SIOP is in initiator or target mode.

Target Mode

OPC2	OPC1	OPC2	Instruction Defined
0	0	0	RESELECT – Reselect instruction
0	0	1	DISCONNECT – Disconnect instruction
0	1	0	WAIT SELECT – Wait for Selection instruction
0	1	1	SET – Set or Assert instruction
1	0	0	CLEAR – Clear or Deassert instruction

Note: Opcode 5, 6, and 7 are Reserved – An Illegal Instruction Interrupt will occur

RESELECT Instruction (Opcode 0)

1. The SIOP arbitrates for the SCSI bus by asserting the SCSI ID stored in the SCID register. If the SIOP loses arbitration, then it tries again during the next available arbitration cycle without reporting any lost arbitration status.
2. If the SIOP wins arbitration, it attempts to reselect the SCSI device whose ID is defined in the Destination ID field of the instruction. Once the SIOP has won arbitration, it fetches the next instruction from the address pointed to by the DSP register.
3. If the SIOP gets selected or reselected before winning arbitration, it fetches the next instruction from the 32-bit address contained in the second longword of the current instruction. This is located in the DSPS register. The SIOP automatically configures itself to be in the initiator mode if reselected, or the target mode if selected.

DISCONNECT Instruction (Opcode 1)

The SIOP disconnects from the SCSI bus by deasserting all SCSI signal outputs. The SCSI direction control signals are deasserted which disables the differential pair output drivers.

WAIT SELECT Instruction (Opcode 2)

1. If the SIOP is already selected, it fetches the next instruction from the address pointed to by the DSP register.
2. If reselected, the SIOP fetches the next instruction from the 32-bit address contained in the second longword of the current instruction. This is located in the DSPS register. The SIOP is automatically configured into initiator mode when reselected.

SET Instruction (Opcode 3)

When the ACK/ and/or ATN/ bits are set to 1, the corresponding bits in the SOCL register are set. Do not use this instruction in target mode.

CLEAR Instruction (Opcode 4)

When the ACK/ and/or ATN/ bits are set to 1, the corresponding bits are cleared in the SOCL register. Do not use this instruction in target mode.

Initiator Mode

OPC2	OPC1	OPC2	Instruction Defined
0	0	0	SELECT – Select instruction
0	0	1	WAIT DISCONNECT – Wait for Disconnect instruction
0	1	0	WAIT RESELECT – Wait for Reselection instruction
0	1	1	SET – Set or Assert instruction
1	0	0	CLEAR – Clear or Deassert instruction

Note: Opcode 5, 6, and 7 are Reserved – An Illegal Instruction Interrupt will occur

SELECT Instruction (Opcode 0)

1. The SIOP arbitrates for the SCSI bus by asserting the SCSI ID stored in the SCID register. If the SIOP loses arbitration, it tries again during the next available arbitration cycle without reporting any lost arbitration status.
2. If the SIOP wins arbitration, it attempts to select the SCSI device whose ID is defined in the instruction's Destination ID field. It then fetches the next instruction from the address pointed to by the DSP register.
3. If the SIOP is selected or reselected before winning arbitration, it fetches the next instruction from the 32-bit address contained in the second longword of the instruction. This is located in the DSPS register. The SIOP automatically configures itself to initiator mode if it was reselected, or to target mode if it was selected.
4. If the Select with ATN/ field is 1, the ATN/ signal is asserted during the selection phase.

WAIT DISCONNECT Instruction (Opcode 1)

The SIOP waits for the target to perform a "legal" disconnect from the SCSI bus. A "legal" disconnect occurs when BSY/ and SEL/ are inactive for a minimum of a Bus Free Delay (400 ns), after the SIOP has received a Disconnect Message or a Command Complete Message.

WAIT RESELECT Instruction (Opcode 2)

1. If the SIOP is selected before being reselected, it fetches the next instruction from the 32-bit address contained in the second longword of the instruction. This is located in the DSPS register. The SIOP automatically configures itself into target mode when selected.
2. If the SIOP is reselected, it fetches the next instruction from the address pointed to by the DSP register.

SET Instruction (Opcode 3)

When the Assert ACK/ and/or Assert ATN/ are 1, the corresponding bits are set in the SOCL register.

This instruction is not valid in target mode.

CLEAR Instruction (Opcode 4)

If the SIOP is operating in initiator mode, then the appropriate bit (ACK/ or ATN/) is reset to 0 in the SOCL register.

This instruction is not valid in target mode.

Select with ATN/ Field (Bit 24)

This bit specifies whether ATN/ was asserted during the selection phase when the SIOP is executing a SELECT instruction. When operating in initiator mode, set it to 1 for the SELECT instruction. If this bit is set to 1 on any other I/O instruction, an Illegal Instruction Interrupt is generated.

SCSI Destination ID Field (Bits 23-16)

This eight-bit field specifies the destination SCSI ID for an I/O instruction. Set only one bit in this field to 1.

SET Target Role (Bit 9)

To enable the SIOP as a target device set bit 10 to 1. This sets bit 0 of the SCNTL0 register to 1. The SIOP remains in target device mode until this bit or bit 0 in the SCNTL0 register is reset to 0.

Assert ACK/ (Bit 6) & Assert ATN/ (Bit 3) Fields

Use these bits during the Set or Clear command. Bit 10, on places the chip in the target/initiator role. Bit 6, on sets/resets the SCSI acknowledge. Bit 3, on sets/resets the SCSI attention.

Writing any of these bits to 1 sets the SIOP, or resets the corresponding bits in the SOCL register. Use the SET instruction to assert ACK/ and/or ATN/ on the SCSI bus. Also, use set Acknowledge to handshake bytes across the SCSI bus.

Use the CLEAR instruction to deassert ACK/ and/or ATN/ on the SCSI bus after the last target message-in byte has been verified for each separate message data Block Move command. The initiator has the opportunity to set attention before acknowledging the last message byte of a Block Move command. On each byte, if a parity error is detected on the message in operation, the ASSERT SCSI ATN is issued before the clear acknowledge is issued to accept the message. Issue clear attention after the target has serviced the request for a message out by the initiator.

ACK/ and ATN/ are not asserted on the SCSI bus unless the SIOP is operating as an initiator or the SCSI Loopback Enable bit is 1 in the CTEST4 register.

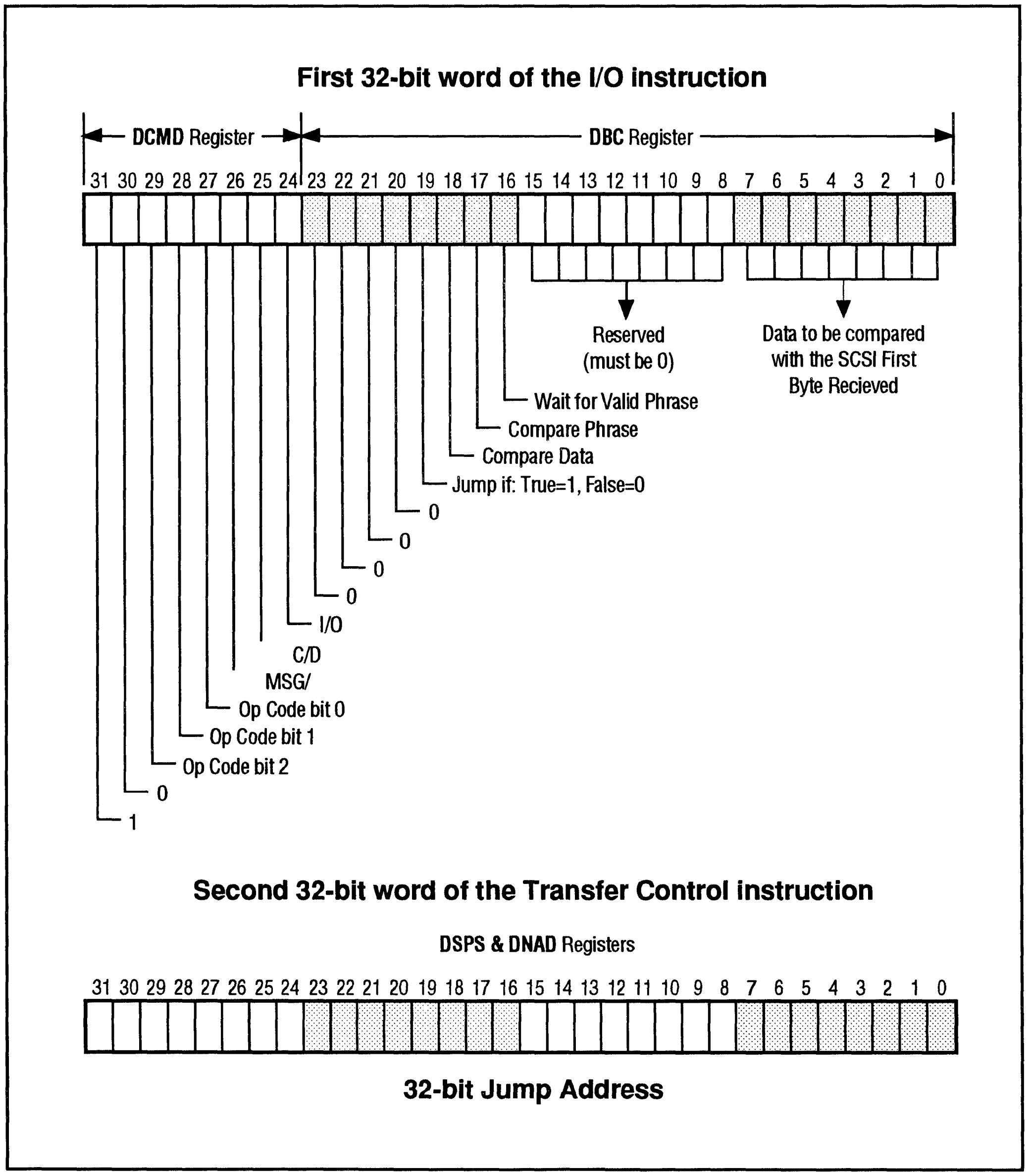
Jump Address Field

This thirty-two-bit field specifies the address of the instruction to fetch when the SIOP encounters a jump condition. The SIOP fetches instructions from the address pointed to by this field whenever the SIOP encounters a SCSI condition that is different from the condition specified in the instruction.

For example, during the execution of a SELECT instruction in initiator mode, if the SIOP is reselected, then the next instruction is fetched from the address pointed to by the jump address field. For a complete description of the different jump conditions, refer to the description of each instruction.

Transfer Control Instructions

Figure 5-3. Transfer Control Instruction Register



Note: In future generations of the SIOP family, the second 32-bit word of the Transfer Control instruction will be loaded into the DSPS only.

Opcode Field (Bits 29, 28, 27)

This field specifies the type of transfer control instruction to be executed. All transfer control instructions can be conditional. They can be dependent on a comparison of the SCSI information transfer phase with the Phase Field and/or a comparison of the First Byte Received with the Data Compare field. Each instruction operates in the initiator or the target mode.

Transfer Control Instructions - Opcode Field Definitions

OPC2	OPC1	OPC2	Instruction Defined
0	0	0	JUMP – Jump instruction
0	0	1	CALL – Call instruction
0	1	0	RETURN – Return instruction
0	1	1	INT – Interrupt instruction

Note: Opcode hex 4-7 are Reserved – If used, an Illegal Instruction Interrupt will occur.

JUMP Instruction (Opcode 0)

1. The SIOP compares the phase and/or data as defined by the Phase Compare, Data Compare and True/False bit fields. If the comparisons are true, the SIOP loads the DSP register with the contents of the second longword of the current instruction which is the 32-bit jump address. The DSP register now contains the address of the next instruction.
2. If the comparisons are false, the SIOP fetches the next instruction from the address pointed to by the DSP register leaving the instruction pointer unchanged.

CALL Instruction (Opcode 1)

1. The SIOP compares the phase and/or data as defined by the Phase Compare, Data Compare, and True/False bit fields. If the comparisons are true, the SIOP saves the current DSP value in TEMP and then loads the DSP contents of the second longword of the current instruction which is the 32-bit call address. The DSP register now contains the address of the next instruction.

   A. When the SIOP executes a CALL instruction, the instruction pointer contained in the DSP register is stored in the TEMP register.

   B. When a RETURN instruction is executed, the value stored in the TEMP register is returned to the DSP register.

2. If the comparisons are false, the SIOP fetches the next instruction from the address pointed to by the DSP register.

RETURN Instruction (Opcode 2)

1. The SIOP compares the phase and/or data as defined by the Phase Compare, Data Compare, and True/False bit fields. If the comparisons are true, then the SIOP loads the DSP register with the contents of the return address stored in the TEMP register. That address value becomes the address of the next instruction.

   A. When the SIOP executes a CALL instruction, the current instruction pointer contained in the DSP register is stored in the TEMP register.

   B. When a RETURN instruction is executed, the value stored in the TEMP register is returned to the DSP register.

   C. The SIOP does not check to see whether the CALL instruction has already been executed. It will not generate an interrupt if a RETURN instruction is executed without previously executing a CALL instruction.

2. If the comparisons are false, then the SIOP fetches the next instruction from the address pointed to by the DSP register.

INT Instruction (Opcode 3)

1. The SIOP compares the phase and/or data as defined by the Phase Compare, Data Compare, and True/False bit fields. If the comparisons are true, then the SIOP generates an interrupt by asserting the IRQ/ signal.
2. The second longword of the INT instruction is a 32-bit field that can contain a unique interrupt service vector. This value is loaded into the DSPS register. When servicing the interrupt, this unique status code allows the ISR to quickly identify the point at which the interrupt occurred.

   After any interrupt or NOP code, the second longword will be in the DSPS register.
3. The SIOP halts. The interrupt must be serviced and the DSP register must be written to start any further operation.

Phase Field (Bits 26, 25, 24)

This three-bit field corresponds to the three SCSI bus phase signals which is compared with the phase lines latched when REQ/ is asserted. Comparisons can be performed to determine the SCSI phase actually being driven on the SCSI bus. For each phase signal, 1 = active and 0 = inactive.

The following table describes the possible combinations and their corresponding SCSI phase. These bits are only valid when the SIOP is operating in initiator mode. When the SIOP is operating in the target mode, these bits are not valid and should be written to 0.

Information Transfer Phases

MSG	C/D	I/O	SCSI Phase
0	0	0	Data-Out
0	0	1	Data-In
0	1	0	Command
0	1	1	Status
1	0	0	Reserved for future use
1	0	1	Reserved for future use
1	1	0	Message-Out
1	1	1	Message-In

Key: “0” equals not asserted, “1” equals asserted

Jump if True/False Field (Bit 19)

This field determines if the SIOP should branch when a comparison is true or when a comparison is false.

If this bit is 1 and the comparison is true, then SIOP executes the Transfer Control instruction (JUMP, CALL, RETURN, or INT) and the SIOP fetches the next instruction from the 32-bit address contained in the second longword of the current instruction. This is located in the DSPS register. The instruction pointer will contain this new address.

If this bit is 1 and the comparison is false, the SIOP fetches the next instruction from the address pointed to by the DSP register.

If this bit is 0 and the comparison is false, the SIOP executes the Transfer Control instruction (JUMP, CALL, RETURN, or INT). Then the SIOP fetches the next instruction from the address pointed to by the DSP register.

Compare Data (Bit 18)

When this bit is 1, then the first byte received from the SCSI data bus is compared with the Data to be Compared Field in the Transfer Control instruction. Use this bit with the Compare Phase Field. The Wait for a valid phase controls when this compare will occur. The Jump if True/False bit determines the condition (true or false) to branch on.

This bit applies to both Phase Compares and Data Compares. If both the Phase Compare and Data Compare bits are set to 1, then both compares must be true to branch on a true condition. Both compares must be false to branch on a false condition.

Compare Phase Field (Bit 17)

When the SIOP is in the initiator mode, this field controls compares to be performed on the SCSI Information Transfer phase information. When this bit is 1, the SCSI phase signals latched by REQ/ are compared to the Phase Field in the Transfer Control instruction. If the phase signals latched by REQ/ are identical to the Phase Field, then the comparison is true.

If the True/False Field is 1, then the Wait for a Valid Phase controls when the compare will occur.

If the Wait for Valid Phase bit is 1, then the SIOP waits for a previously unserviced phase before comparing the SCSI phases.

If the Wait for Valid Phase bit is 0, then the SIOP compares the SCSI phases immediately. When the SIOP is operating in target mode and this bit is 1, this field will test for an active SCSI ATN/ signal.

Compare Data Field (Bit 16)

If this bit is 1, then the first byte received from the SCSI data bus is compared to the Data to be Compared Field in the Transfer Control instruction. The Wait for a Valid Phase controls when the compare will occur.

If the Wait for Valid Phase bit is 1, then the SIOP waits for a previously unserviced phase before comparing the data.

If the Wait for Valid Phase bit is 0, then the SIOP compares the SCSI data immediately. This bit can be used with the Compare Phase Field.

If both the Compare Data Field and Compare Phase Field bits are set, then the compare includes both the SCSI phase and the data byte.

If the True/False bit is 1 and the phase and data are compared and determined to be identical, then the SIOP fetches the next instruction from the address pointed to by the 32-bit jump address field.

If the True/False bit is 1 and either the phase or the data are different, then the SIOP fetches the next instruction from the address pointed to by the DSP register.

If the True/False bit is 0, and either the phase or the data do not match the compare fields, then the SIOP fetches the next instruction from the address pointed to by the DSP register.

If the True/False bit is 0, and both the phase and the data are different, then the SIOP fetches the next instruction from the address pointed to by 32-bit jump address field.

Mask for Compare Data (Bits 15-8)

The mask bits allow selective comparison of bits within the data bytes using SCRIPTS. During the compare, any bits that are on cause the corresponding bit in the data byte to be ignored for the comparison. A user can code a binary sort to quickly determine the value of a byte.

For instance, a mask of '7F' and data compare of '80' allows the SCRIPTS processor to determine whether or not the high order bit is on.

Data to be Compared Field (Bits 7-0)

This 8-bit field is the data compared to the SCSI first byte received register. Use this bit with the Compare Data field to compare for a particular data pattern.

Jump Address Field (Bits 31-0, DNAD register)

This 32-bit field contains the address of the next instruction to fetch when the compare operations are successful. For example, if a JUMP instruction is issued, the Compare Data & True/False bits are 1, and the SCSI First Byte Received is equal to the Data to be Compared field, then the SIOP fetches the next instruction from this 32-bit address.

Chapter Six

How to Use the 53C700/53C700-66

This Chapter contains the following examples of how to use the SIOP chip. This information applies to both chips except where it says 53C700-66 only.

- How to Start the SIOP in the SCSI SCRIPTS Mode
- How to Execute Normal SCRIPTS
- How to Execute Single-Step SCRIPTS
- Steps Necessary to Start SCRIPTS
- How to Test the SIOP in the Loopback Mode
- How to Implement Parity Options
- How to Test the DMA FIFO
- How to Test the SCSI FIFO
- How to Abort an Operation
- How to Disconnect the SIOP
- How to Select a Target
- How to Reselect an Initiator
- How to Respond to Multiple SCSI IDs
- How to Use Single-Ended SCSI Interface
- How to Use Differential SCSI Interface
- How to Terminate the SIOP Device

How to Start the SIOP in the SCSI SCRIPTS Mode

To start the SIOP in SCSI SCRIPTS mode, first load the DSP register with the address location containing the first SCSI SCRIPTS instruction. The SIOP fetches the first instruction from the address pointed to by the DSP register. Once the instruction is received, the DSP register is incremented by 8 and it points to the next SCSI SCRIPTS address. The SIOP continues to fetch and execute instructions from system memory until either an interrupt condition occurs or an interrupt instruction is executed. Once an interrupt is generated, the SIOP halts all operations until the interrupt is serviced. Once the SIOP has halted, write the address of the next instruction in the DSP register to restart the automatic fetch and execution of the instructions.

How to Execute Normal SCRIPTS

To start SCRIPTS instructions, write the SCRIPTS address to the DSP register. Then wait for an interrupt or poll the ISTAT register.

How to Execute Single-Step SCRIPTS

To execute single-step mode (one instruction at a time) set bit 4 to 1 in the DCNTL register. To start SCRIPTS instructions, write the SCRIPTS address to the DSP register. Then wait for a single-step interrupt. Execute subsequent instructions by setting the start DMA bit (bit 2) to 1 in the DCNTL register. Repeat until the end of the SCRIPTS instructions.

Steps Necessary to Start SCRIPTS

The following list gives the programming steps for initializing the SIOP to start fetch and execution of SCRIPTS instructions.

1. Assert hardware or software RESET (DCNTL register, bit 0)

Bits	Description
2	Fixed address mode
3	I/O or memory mapped
4	286 mode (16-bit device, set this bit on the first register access after a hardware reset)
5	Bus width 16
7-6	Host burst length

2. Program DMODE (34h)

3. Program SCNTLO (00h)

Bits	Description
0	Target mode
1	Assert ATN/ on parity error
2	Enable parity generation
3	Enable parity checking

4. Program SCNTL1 (01h)

Bits	Description
5	Enable selection and reselection

5. Program SIEN (03h)

Bits	Description
0	Enable parity error interrupt
1	Enable SCSI RST/ received inter rupt
2	Enable unexpected disconnect interrupt
3	Enable SCSI gross error interrupt
4	Enable selected or reselected interrupt
5	Enable selection or reselection time-out interrupt
6	Enable function complete interrupt
7	Enable phase mismatch or ATN/ active interrupt

6. Program SCID (04h)

Bits	Description
7-0	Chip's ID

7. Program SXFER (05h) *

Bits	Description
3-0	Synchronous offset
6-4	Synchronous transfer period
7	Disable halt on a parity error or ATN/

8. Program CTEST7 (1Bh)

Bits	Description
0	Enable differential mode
1	Enable DC/ low for instruction fetch
2	Enable even parity when sending data to host bus

8a. Program CTEST8 (22h)

53C700-66 chip only

Bits	Description
4	Enable Active Negation
2	Shorten Request/ Acknowledge filter

9. Program DIEN (39h)

Bits	Description
0	Enable illegal instruction interrupt
1	Enable watchdog timeout interrupt
2	Enable SCRIPTS interrupt instruction
	received interrupt
3	Enable SCRIPTS pipeline/step interrupt
4	Enable aborted interrupt

10. Program DWT (3Ah)

Bits	Description
7-0	Enter watchdog timer period

11. Program DCNTL (3Bh)

Bits	Description
4	Single-step mode
5	Scripts loaded in 16-bit mode (only for 386 mode)
6-7	Clock frequency divide bits

12. Program DSP (2Fh-2Ch) **

Bits	Description
31-0	Start address of script

* Note: Synchronous offset and transfer period are normally set after synchronous negotiation

** Once SCRIPTS is running ISTAT is the only register that may be read. Other registers may be accessed once the scripts has been interrupted.

How to Test the SIOP in the Loopback Mode

SIOP loopback mode allows testing of both initiator and target operations. When the Loopback Enable bit is 1 in the CTEST4 register, the SIOP allows control of all SCSI signals, whether the SIOP is operating in initiator or target mode. Perform the following steps to implement loopback mode.

1. Set the Loopback Enable bit in the CTEST4 register to 1.
2. Set-up the desired arbitration mode as defined in the SCNTL0 register.
3. Set the Start Sequence bit in the SCNTL0 register
to 1 .

4. Poll the SBCL register to determine when SEL/ is active and BSY/ is inactive.
5. Poll the SBDL register to determine which SCSI ID bits are being driven.
6. In response to selection, set the BSY/ bit (bit 5) in the SOCL register to 1.
7. Poll the SEL/ bit in the SBCL register to determine when SEL/ becomes inactive.
8. To assert the desired phase, set the MSG/, C/D, and I/O bits to the desired phase in the SOCL register.
9. To assert REQ/, keep the phase bits the same and set the REQ/ bit in the SOCL register to 1. To accommodate the 400 ns bus settle delay, set REQ/ after setting the phase signals,
10. The initiator role can be implemented by single stepping SCSI SCRIPTS and the SIOP can loopback as a target or vice versa.

How to Implement Parity Options

The SIOP implements a flexible parity scheme that allows control of the type of parity, whether parity is checked, and whether a bad parity byte is deliberately sent to the SCSI bus to test parity error recovery procedures. The parity options are controlled by the following bits:

1. Assert ATN/ on Parity Errors (AAP) – Bit 1 in the SCNTL0 register.

   This control bit allows the SIOP to automatically assert SCSI ATN/ when it detects a parity error while the SIOP is operating as an initiator.

2. Enable Parity Generation (EPG) – Bit 2 in the SCNTL0 register.

   This bit controls whether the SIOP generates parity sent to the SCSI bus or allows parity to "flow through" the chip to/from the SCSI bus and system bus.

3. Enable Parity Checking (EPC) – Bit 3 in the SCNTL0 register.

   This bit determines if the SIOP will check for parity errors. The SIOP checks for odd or even parity depending on the status of the Assert Even SCSI Parity bit.

4. Assert Even SCSI Parity (PAR) – Bit 2 in the SCNTL1 register.

   This bit determines if the SIOP checks for and then asserts even or odd parity SCNTL1 register, bit 2).

5. Disable Halt on ATN/ or a Parity Error Target Mode Only (DHP) – Bit 7 in the SXFER register.

   This bit determines if the SIOP will halt operations when a parity error is detected in target mode.

6. Enable Parity Error Interrupt (PAR) – Bit 0 in the SIEN register.

   This bit determines if the SIOP will generate an interrupt when it detects a parity error.

7. Parity Error (PAR) – Bit 0 in the SSTAT0 register.

   This status bit is 1 whenever the SIOP has detected a parity error from either the SCSI bus or the system bus.

8. Status of SCSI Parity Signal (SDP/) – Bit 0 in the SSTAT1 register.

   This status bit represents the live SCSI parity signal (SDP/). When SDP/ is active, it is 1.

9. Latched SCSI Parity Signal (SDP) – Bit 3 in the SSTAT2 register.

   This status bit represents the parity signal (SDP/) after the First Byte Received is latched in the chip for a particular phase. When SDP/ is active, it is 1.

10. DMA FIFO Parity bit (DFP) – Bit 3 in the CTEST2 register.

    This status bit represents the parity bit in the DMA FIFO after data is read from the FIFO by reading the CTEST6 register.

11. DMA FIFO Parity bit (DFP) – Bit 3 in the CTEST7 register.

    This write-only bit is written to the DMA FIFO after writing data to the DMA FIFO by writing the CTEST6 register.

12. SCSI FIFO Parity bit (SFP) – Bit 4 in the CTEST2 register.

    This status bit represents the parity bit in the SCSI FIFO after data is read from the FIFO by reading the CTEST3 register.

How to Control Parity

Table 6-1. Parity Control Signals and Descriptions

EPG (Parity Generation)	EPC (Parity Checking)	AESP (Even SCSI Parity)	EVP (Even Host Parity)	Description
0	0	x	0	Parity pass through (DP3-DP0). No parity checking. Odd parity passed through the chip (DP3-DP0).
0	0	x	1	Parity pass through (DP3-DP0). No parity checking. Even parity asserted to Host bus when receiving from SCSI (done by inverting SCSI parity). Odd parity asserted on SCSI bus when sending to SCSI (inverts Host bus parity).
0	1	0	0	Parity pass through (DP3-DP0). Parity checking (always checks odd SCSI parity). Odd parity throughout chip.
0	1	0	1	Parity pass through (DP3-DP0). Parity checking (always checks odd SCSI parity). Odd SCSI parity asserted on SCSI bus when sending data (inverts Host bus parity). Even parity asserted on Host bus when receiving data from SCSI (inverts parity received from SCSI bus).
0	1	1	0	Parity pass through (DP3-DP0). Parity checking (always checks odd SCSI parity). Asserts even parity on SCSI bus when sending data to SCSI bus (inverts parity from host bus).
0	1	1	1	Parity pass through (DP3-DP0). Parity checking (assuming odd parity received from Host bus, this configuration will always generate a parity error when sending data to the SCSI bus). Assert even parity on SCSI bus when sending data to SCSI bus (inverts parity from Host bus). Assert even parity on Host bus when receiving data from SCSI bus (inverts parity from SCSI bus).
1	0	0	x	Parity generation (DP3-DP0 ignored). No parity checking. Odd parity generated on SCSI bus.

Table 6-1. Parity Control Signals and Descriptions (Continued)

EPG (Parity Generation)	EPC (Parity Checking)	AESP (Even SCSI Parity)	EVP (Even Host Parity)	Description
1	0	1	x	Parity generation (DP3-DP0 ignored). No parity checking. Even parity generated on SCSI bus.
1	1	0	x	Parity generation (DP3-DP0 ignored). Parity checking. Odd parity generated and checked on SCSI bus.
1	1	1	x	Parity generation (DP3-DP0 ignored). Parity checking. Even parity generated onto SCSI bus. Odd parity checked when receiving data from SCSI bus.

Key:			
	EPG = Enable Parity Generation	(SCNTL0 register, bit 2)	1 = asserted
	EPC = Enable Parity Checking	(SCNTL0 register, bit 3)	0 = deasserted
	AESP = Assert SCSI Even Parity	(SCNTL1 register, bit 2)	x = don't care
	EVP = Assert Even Host Parity	(CTEST7 register, bit 2)	

What Parity Errors and Interrupts Occur?

The following table describes the options available when a parity error occurs. This table only applies to the case where the Enable Parity Checking bit is 1 (SCNTL0 register, bit 2).

Table 6-2. Parity Errors and Interrupts

DHP	EPI	Description
0	0	Will NOT halt when a parity error occurs in target or initiator mode
0	1	Will interrupt when a parity error occurs in target or initiator mode
1	0	Will halt when a parity error occurs in target mode, will NOT generate an interrupt
1	1	Will halt when a parity error occurs in target mode, will generate an interrupt in target or initiator mode

Key:		
	DHP = Disable Halt on ATN/ or a Parity Error	(SXFER register, bit 7)
	EPI = Enable Parity Interrupt	(SIEN register, bit 0)

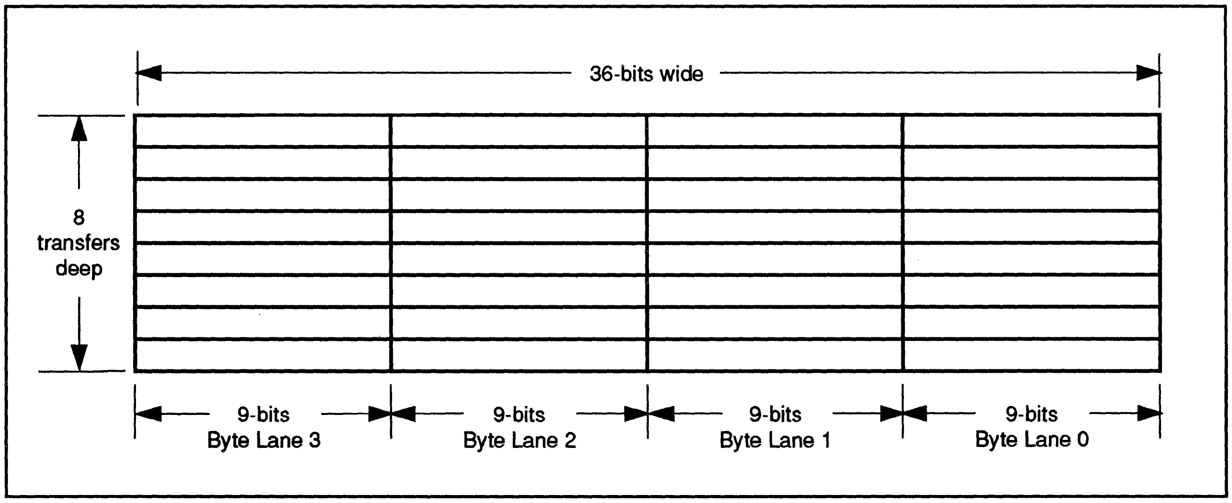
How to Test the DMA FIFO

The DMA FIFO is more complex than the SCSI FIFO. The DMA FIFO is a 36-bit wide x 8 deep FIFO.

It can be divided into 4 sections, each being 9-bits wide and 8 transfers deep.

Each of these four sections are labeled as "Byte Lanes." Each can be individually tested by writing known data into the FIFO and reading that same data back out of the FIFO.

Figure 6-1. DMA FIFO Sections



To write data into the DMA FIFO, load the data 9-bits per instruction. Data is written to the top of the FIFO and is read from the bottom of the FIFO. Three control bits in the CTEST4 register allow access to any one of the four "Byte Lanes."

Parity is written to the FIFO through bit 3 of the CTEST7 register. Set this bit to the desired value before each write operation to the FIFO.

To the appropriate "Byte Lane", write the following three bits as shown in Table 6-3.

Table 6-3. Byte Lane Selection

FBL2	FBL1	FBL0	Description
0	X	X	Access disabled (set to this value before executing SCSI SCRIPTS)
1	0	0	Byte Lane 0
1	0	1	Byte Lane 1
1	1	0	Byte Lane 2
1	1	1	Byte Lane 3

X = Don't Care

How to Load the DMA FIFO

To completely load the DMA FIFO with known data and read back the data parity, perform the following steps.

1. Write an 04h to the CTEST4 register to setup access to Byte Lane 0 in the DMA FIFO.

   ```
   outportb (CTEST4, 0x04);
   ```

2. Write the parity bit to the CTEST7 register bit 3 and the 8-bit data value to the CTEST6 register to write the desired data pattern to the DMA FIFO.

   ```
   for (i=0; i<8; ++i)
     {
     outportb (CTEST7, parity);
     /* parity values are 0x08
     equals parity of 1 */
     /* 0x00 equals a parity of 0 */
     outportb (CTEST6, i);
     /* incrementing pattern */
     }
   ```

3. Read the parity bit in the CTEST2 register to read data back out of the FIFO.

   ```
   for (i=0; i<8; ++i)
     {
     byte_lane0 [i] = inportb
     (CTEST6);
     /* read back in data out of
     the FIFO  */
     parity 0 [i] = (inportb
     (CTEST2) & 0x08);
     /* mask all but the parity
     bit - 0x08=1 */
     /* 0x00=0 */
     }
   ```

4. Repeat the above sequence for Byte Lanes 1 through 3.
5. Disable DMA FIFO access by writing zero to the CTEST4 register.

   ```
   outportb (CTEST4, 0x00);
   ```

How to Test the SCSI FIFO

Use the SCSI FIFO Write Enable bit in the CTEST4 register to load the SCSI synchronous data FIFO with data using any microprocessor. To load the SCSI FIFO with a known data pattern, write this bit to 1. The data is loaded into the SCSI FIFO by writing to the SODL register. The microprocessor reads the CTEST3 register to read data out of the SCSI FIFO. Reading bit 4, the SCSI FIFO parity bit in the CTEST2 register checks parity when reading data out of the FIFO after reading CTEST3. The parity bit is stored in the CTEST2 register during a CTEST3 register read.

Write parity to the FIFO in one of two ways.

1. Parity can flow into the SIOP on the parity signals if the Enable Parity Generation bit in the SCNTL0 register equals 0. The microprocessor drives the parity signal for the corresponding 8-bit data signals.

   For example, writing the FIFO to the SIOP on "Byte Lane 2" (D23-D16) should make DP2 drive the parity information.

2. If the Parity Generation bit is equal to 1, then the SIOP forces the parity bit to even or odd parity. Set the Assert Even SCSI Parity bit in the SCNTL1 register to 0 to load the SCSI FIFO with odd parity. If this bit is equal to 1, then the SCSI FIFO will be loaded with even parity.

Follow the steps below to completely load the SCSI FIFO with known data and be able to read back the data parity.

1. Write the control bits to determine the method and type of parity to be loaded into the SCSI FIFO.

EPG	AESP	Parity Type & Loading Method
0	X	Parity is loaded on the hardware signals DP3-DP0
1	0	Odd parity is automatically loaded when the SODL register is written
1	1	Even parity is automatically loaded when the SODL register is written

Key:

EPG= Enable Parity Generation bit in the SCNTL0 register

AESP= Assert Even SCSI Parity bit in the SCNTL1 register

X = Don't Care

```
outportb (SCNTL0, config0_info);

outportb (SCNTL1, config1_info);
```

2. Write the SCSI FIFO Write Enable bit to 1 in the CTEST4 register to enable the SCSI FIFO to accept data. When the FIFO test is complete, rewrite this bit to 0.

```
outportb (CTEST4, 0x08);
```

3. Load the SCSI FIFO with the desired data value by writing a known data pattern to the SODL register.

```
for (i=0; i<8; ++i)
  {
  outportb (SODL, i);
  /* incrementing pattern */
  }
```

4. Read the data back by reading the CTEST3 register.

```
for (i=0; i<8; ++i)
  {
  test_data_in [i] = inportb
  (CTEST3);
  /* should be the incrementing
  pattern  */
  test_parity_in [i] = inportb
  (CTEST2);
  /* bit 4 of this register is
  the parity bit for */
  /* the byte just read out of
  CTEST3    */
  }
```

5. Reset the SCSI FIFO Write Enable bit.

```
outportb (CTEST4, 0x00);
```

How to Abort an Operation

Performing an abort stops the SCRIPTS execution, it does not reset the chip.

Hardware Abort

If DP3_ABRT/ is used to perform a hardware abort, parity generation must be enabled through the SCNTLO register, bit 2. Read the DSTAT register to clear an abort interrupt.

Software Abort

Use the ISTAT register, bit 7 for a software abort, also clear this bit before clearing the interrupt to prevent multiple interrupts.

Before the abort is performed, the current data burst is completed. So when receiving data from SCSI, the DMA FIFO is sent to the Host before an abort interrupt. If the abort occurs during a block move, the number of bytes left to transfer can be determined by the following:

- **SCSI Asynchronous Receive**

The number of bytes left to transfer is in the DBC register.

```
    # of bytes   = DBC
```

Read the SSTAT1 register, bit 7, to determine if a byte is left in the SIDL register. If bit 7 is set, then there is a byte left in the SIDL register. This byte has been counted by the sender, but not by the SIOP because the DBC is not decremented until the byte is written to memory. To recover this byte, read it from the SIDL register. Decrement the number in the DBC by one so that the sender and receiver agree on the number of bytes that have been transferred.

- **SCSI Synchronous Receive**

The number of bytes left to transfer is in the DBC register.

```
    # of bytes   = DBC
```

Read the SSTAT2 register and test bits 7-4 (the binary representation of the number of valid bytes in the SCSI FIFO) to determine if any bytes are left in the SCSI FIFO. These bytes have been counted by the sender, but not by the SIOP because the DBC is not decremented until the bytes are written to memory. To recover these bytes, read them from the CTEST3 register. Decrement the number in the DBC by the number in SSTAT2 so that the sender and receiver agree on the number of bytes that have been transferred.

- **SCSI Asynchronous Send**

```
    # of bytes = DBC + # bytes in
      DMA FIFO

    # bytes in DMA FIFO = [DFIFO
      (bits 5-0) - DBC (bits 5-0)]
      AND 3F (mask for lower 6 bits)
```

Check the SODL register full (SSTAT1, bit 5). If it is full, add 1 to the number of bytes.

- **SCSI Synchronous Send**

```
    # of bytes = DBC + # bytes in
      DMA FIFO

    # bytes in DMA FIFO = [DFIFO
      (bits 5-0) - DBC (bits 5-0)]
      AND 3F (mask for lower 6 bits)
```

Check the SODL register full (SSTAT1, bit 5). If it is full, add 1 to the number of bytes.

Check the SODR Register full (SSTAT1, bit 6). If it is full, add 1 to the number of bytes.

Note: This is the same algorithm used to recover bytes after an unexpected disconnect during a Block Move.

Clear the SCSI and DMA FIFOs (DFIFO register, bit 6 before starting another data transfer).

To continue operation, clear the abort interrupt and write the DSP register to start another SCRIPT.

Abort Example (Software)

This is an example of issuing a software abort to start another SCSI I/O while executing WAIT RESELECT (as an initiator) or WAIT SELECT (as a target). This is a common situation with multi-threaded I/O performing disconnects and reconnects.

Check the ISTAT register, bit 3 to determine whether the SIOP is connected.

- If connected, (ISTAT register, bit 3=1) exit and wait for the current I/O to complete or schedule it for the next I/O.
- If not connected, (ISTAT register, bit 3=0) do the following:
  1. Issue an abort by asserting ISTAT register, bit 7.
  2. Wait for the abort interrupt.
  3. Check for connection again (ISTAT register, bit 3).

4. If the SIOP is supporting both the initiator and the target roles, the following apply.

- If connected, the SIOP provides a bit indicating that it has been selected or reselected (SSTAT0, bit 4). However, it is not possible to tell at the register level if the SIOP has been selected or reselected. The work-around for this dilemma is to start a SCRIPTS "SELECT id, alt_addra" instruction. If the SIOP has been selected or reselected, the jump to alt_addra is taken. The SCRIPTS next to the alt_addra label will look like the following:

```
alt_addra:WAIT RESELECT,alt_addrb
INT 10:

   /* Interrupt 10 indicates that
   the SIOP has been reselected */
```

The interrupt instruction after the WAIT RESELECT instruction indicates that the SIOP has been reselected. If the SIOP has been selected, the jump to alt_addrb label will look like the following:

```
alt_addrb:WAIT SELECT, alt_addrc

 /* The SIOP has been selected */
```

The interrupt instruction after the WAIT SELECT instruction indicates that the SIOP has been selected.

If the instruction you were trying to abort was a WAIT RESELECT and the above routine determines you were RESELECTED, OR if the instruction you were trying to abort was a WAIT SELECT and the above routine determines you were SELECTED then do the following:

It is possible that while you were executing the first WAIT RESELECT or WAIT SELECT instruction that you were RESELECTED or SELECTED before the abort took place. Once you are RESELECTED or SELECTED the SCRIPTS will continue on to the next instruction, which is most likely a Block Move instruction to receive the identify message. By the time you try to abort the SCRIPTS, it could have proceeded past the WAIT RESELECT or WAIT SELECT instruction. Therefore, you would actually be aborting that Block Move instruction, not the WAIT RESELECT or WAIT SELECT instruction. You can tell which instruction was aborted by subtracting 8 from the DSP register. This will tell you which instruction was aborted. In this case you will have to find out how much data had been transferred and restart at the correct place. You can use the routine given at the beginning of the Software Abort section to determine how much data has been transferred during an aborted Block Move instruction.

- If not connected, start SCRIPTS for a new I/O.

5. If the SIOP is not supporting both initiator and target roles, the following apply.

It is possible that while you were executing the first WAIT RESELECT or WAIT SELECT instruction that you were RESELECTED or SELECTED before the abort took place. Once you are RESELECTED or SELECTED the SCRIPTS will continue on to the next instruction, which is most likely a Block Move instruction to receive the identify message. By the time you try to abort the SCRIPTS, it could have proceeded past the WAIT RESELECT or WAIT SELECT instruction. Therefore, you would actually be aborting that Block Move instruction, not the WAIT RESELECT or WAIT SELECT instruction. You can tell which instruction was aborted by subtracting 8 from the DSP register. This will tell you which instruction was aborted. In this case you will have to find out how much data had been transferred and restart at the correct place. You can use the routine given at the beginning of the Software Abort section to determine how much data has been transferred during an aborted Block Move instruction.

How to Disconnect the SIOP

If there is an unexpected disconnect during a Block Move, use the following steps to save the state of SIOP at disconnect.

1. Read the address of the current SCRIPTS instruction from the DSP register. Subtract 8 from the address to get the address of the instruction executing when disconnect occurred.

2. Determine the number of bytes left to transfer.

- **SCSI Asynchronous Receive**

  The number of bytes left to transfer is in the DBC register.

  ```
  # of bytes = DBC
  ```

  Read the SSTAT1 register, bit 7, to determine if a byte is left in the SIDL register. If bit 7 is set, then there is a byte left in the SIDL register. This byte has been counted by the sender, but not by the SIOP because the DBC is not decremented until the byte is written to memory. To recover this byte, read it from the SIDL register. Decrement the number in the DBC by one so that the sender and receiver agree on the number of bytes that have been transferred.

- **SCSI Synchronous Receive**

  The number of bytes left to transfer is in the DBC register.

  ```
  # of bytes = DBC
  ```

  Read the SSTAT2 register and test bits 7-4 (the binary representation of the number of valid bytes in the SCSI FIFO) to determine if any bytes are left in the SCSI FIFO. These bytes have been counted by the sender, but not by the SIOP because the DBC is not decremented until the bytes are written to memory. To recover these bytes, read them from the CTEST3 register. Decrement the number in the DBC by the number in SSTAT2 so that the sender and receiver agree on the number of bytes that have been transferred.

- **SCSI Asynchronous Send**

  ```
  # of bytes = DBC + # bytes in DMA
    FIFO

  # bytes in DMA FIFO = [DFIFO
    (bits 5-0) - DBC (bits 5-0)]
    AND 3F (mask for lower 6 bits)
  ```

Check the SODL register full (SSTAT1, bit 5). If it is full, add 1 to the number of bytes.

- **SCSI Synchronous Send**

  ```
  # of bytes   = DBC + # bytes in
    DMA FIFO

  # bytes in DMA FIFO = [DFIFO
    (bits 5-0) - DBC (bits 5-0)]
    AND 3F (mask for lower 6 bits)
  ```

Check the SODL register full (SSTAT1, bit 5). If it is full, add 1 to the number of bytes.

Check the SODR Register full (SSTAT1, bit 6). If it is full, add 1 to the number of bytes.

Note: This is the same algorithm used to recover bytes after an abort.

3. Clear SCSI and DMA FIFOs (DFIFO register, bit 6).

4. Update the SCRIPTS to handle reselect by the disconnecting device.

How to Select a Target

In multi-tasking or multi-threaded SCSI I/O operations, it is common to become selected or reselected when trying to perform selection. This situation occurs when a SCSI controller operating in the initiator mode tries to select a target.

If the SIOP is executing

```
    SELECT id, alt_addr
```

and it becomes selected or reselected, the jump to the alt_addr is taken.

If the SIOP is executing

```
    WAIT RESELECT alt_addr1
```

and it is selected, the jump to the alt_addr1 is taken.

Typically, the alt_addr for

```
    SELECT id, alt_addr
```

would be a **WAIT RESELECT** instruction.

```
    alt_addr:
      WAIT  RESELECT alt_addr1
      JUMP  address, if id MASK data
      . . .
```

If the SIOP has been reselected, the SCRIPTS can then compare for id and jump to the corresponding device I/O script.

If the SIOP has been selected, the jump to alt_addr1 is then taken where a target SCRIPTS can be executed.

```
    ;Target SCRIPT
    alt_addr1:
      WAIT SELECT     alt_addr2
      . . .
```

How to Reselect an Initiator

In multi-tasking or multi-threaded SCSI I/O operations, it is common to become selected or reselected when trying to perform a reselection. This situation occurs when a SCSI controller operating in target mode tries to reselect an initiator.

If the SIOP is executing

```
    RESELECT id, alt_addr
```

and it is selected or reselected, the jump to alt_addr is taken.

If the SIOP is executing

```
    WAIT SELECT alt_addr1
```

and it is reselected, the jump to alt_addr1 is taken.

Typically the alt_addr for

```
    RESELECT id, alt_addr
```

would be a **WAIT SELECT** instruction.

```
    alt_addr:
      WAIT SELECT alt_addr1
      Jump address, if id MASK data
      . . .
```

If the SIOP has been selected, the IDs can be compared and then jump to the corresponding device I/O SCRIPT.

If the SIOP is reselected, the jump to alt_addr1 is taken where an initiator SCRIPT can be executed.

```
    alt_addr1:
      WAIT RESELECT alt_addr2
      . . .
```

How to Respond to Multiple SCSI IDs

In certain applications, it is desirable to respond to multiple SCSI IDs. This is most common for devices operating as targets but can also occur for initiators. The 53C700 family operates the same way for either case but the language describing the SCSI action is different. Therefore, this algorithm will describe the target case and refer to the initiator case in *italics*. This procedure applies to both the 53C700 and 53C700-66, but for simplicity the name "53C700" is used to refer to both of them.

The 53C700 family is able to respond to multiple IDs during Selection simply by setting multiple ID bits in the SCID register. However, if it becomes necessary for the 53C700 to Reselect an initiator, it will arbitrate using the highest priority ID bit set in the SCID register.

For example, if SCID is set to a value of 0F hex, the 53C700 may be selected as target ID 0, 1, 2 or 3, but will ONLY arbitrate for Reselection as ID three. To arbitrate as another ID, it is necessary to change the SCID. This leads to difficulties if another device tries to select the 53C700 as a higher ID during the time the SCID contains the lower ID. If this can occur in your system, the 53C700 must perform the Reselection in low level mode, as follows:

1. Make sure the chip has completed any information transfers and that a disconnect has taken place (DISCONNECT has completed).
2. Disable response to Selection/Reselection (ESR) with a write to SCNTL1. The same register write should reset the Connected (CON) bit.
3. Wait 400 ns (Bus Settle Delay).
4. Check SOCL bit 5 (BSY). If this bit is set, the chip has been selected. In this case, the CON bit must be set by writing SCNTL1, and the ESR bit may be set if desired. A SCRIPT starting with WAIT SELECT may then be initiated.
5. If SEL bit in SBCL is set, goto step (27).
6. If BSY bit in SOCL is set, goto step (4).
7. Clear the low-level mode bit, disable parity checking.
8. Set one arbitration ID in the SCID register.
9. Enable simple arbitration (reset SCNTL0 bits 6 and 7).
10. Start simple arbitration sequence (set SCNTL0 bit 5).
11. Wait for one of the following to occur:
    a. Arbitration in progress (SSTAT1 bit 4 = 1), goto step (21).
    b. Lost arbitration (SSTAT1 bit 3 = 1), goto step (12).
    c. SEL asserted (SBCL bit 4 = 1), goto step (12).
12. Stop any pending arbitration (reset SCNTL0 bit 5).
13. While SEL is asserted (SBCL bit 4 = 1): if BSY is deasserted (SBCL bit 5 = 0) and any response ID is set in SBDL, goto step (15).

13. Goto step (10) (restart arbitration).

15. Enable parity checking if desired.
16. Write response ID(s) to SCID register, then set ESR bit in SCNTL1.
17. The chip is being Selected or Reselected; start WAIT SELECT SCRIPT.
18. Wait 1400 ns.
19. If not connected (ISTAT bit 3 = 1), goto step (13).
20. Exit.
21. Wait 2400 ns (Arbitration Delay).
22. If lost arbitration (SSTAT1 bit 3 = 1), goto step (13).
23. If the chip ID is the highest asserted(compare our ID with any other IDs present in SBDL), go to step (25).
24. Goto step (22).

25. Enable parity checking if desired.
26. The chip will win arbitration; start RESELECT SCRIPT and exit. Enable Selection/Reselection if desired.
27. If BSY bit in SOCL is set, goto step (4).
28. If SEL bit in SBCL is clear, goto step (6).
29. If any response IDs are set in SBDL, goto step (16).
30. Goto step 27.

Note: When the response IDs are checked in step (13), it is possible that an error (for example parity error or too many IDs asserted) may exist on the SCSI bus. Rather than check for all possible errors in low level mode, a WAIT SELECT SCRIPT is started. If errors do exist, the chip will not respond. To detect this, a waiting period of 1400 ns is inserted. The CONNECTED bit in ISTAT is checked to see if we have responded. If CONNECTED is not set, the chip did not respond because of a SCSI error and the WAIT SELECT SCRIPT must be aborted, step (19). When the abort is complete, the chip will have neither won arbitration, nor responded to Selection. Step (20), "Exit", assumes that this entire process will restart itself at step (1).

How to Use Single-Ended SCSI Interface

The SIOP can be used in both single-ended and differential applications. In single-ended mode, all SCSI signals are active-low. The SIOP contains the open-drain output drivers that can be connected directly to the single-ended SCSI bus. Each output is isolated from the VDD power supply to ensure that the SIOP has no effect on an active SCSI bus when its VDD is powered down. Additionally, some signal filtering has been added to the inputs of REQ/ and ACK/ to reduce the possibility of signal reflections corrupting the transfer.

Figure 6-2. Single-Ended SCSI Interface

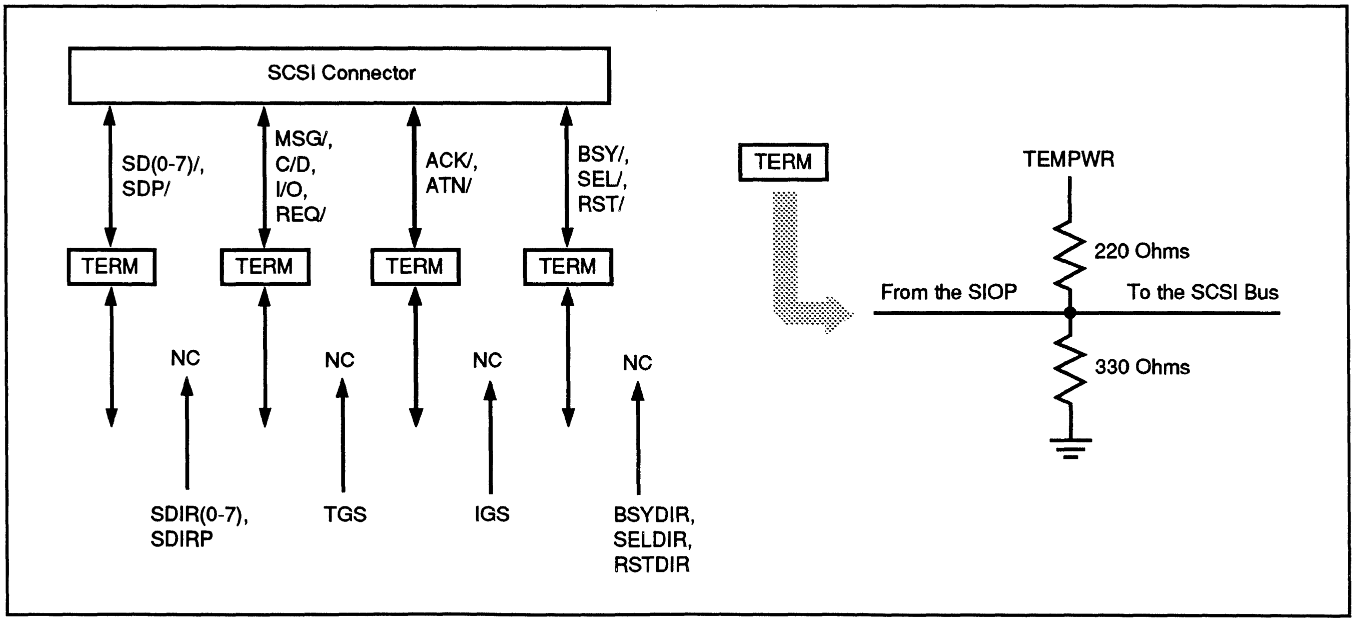
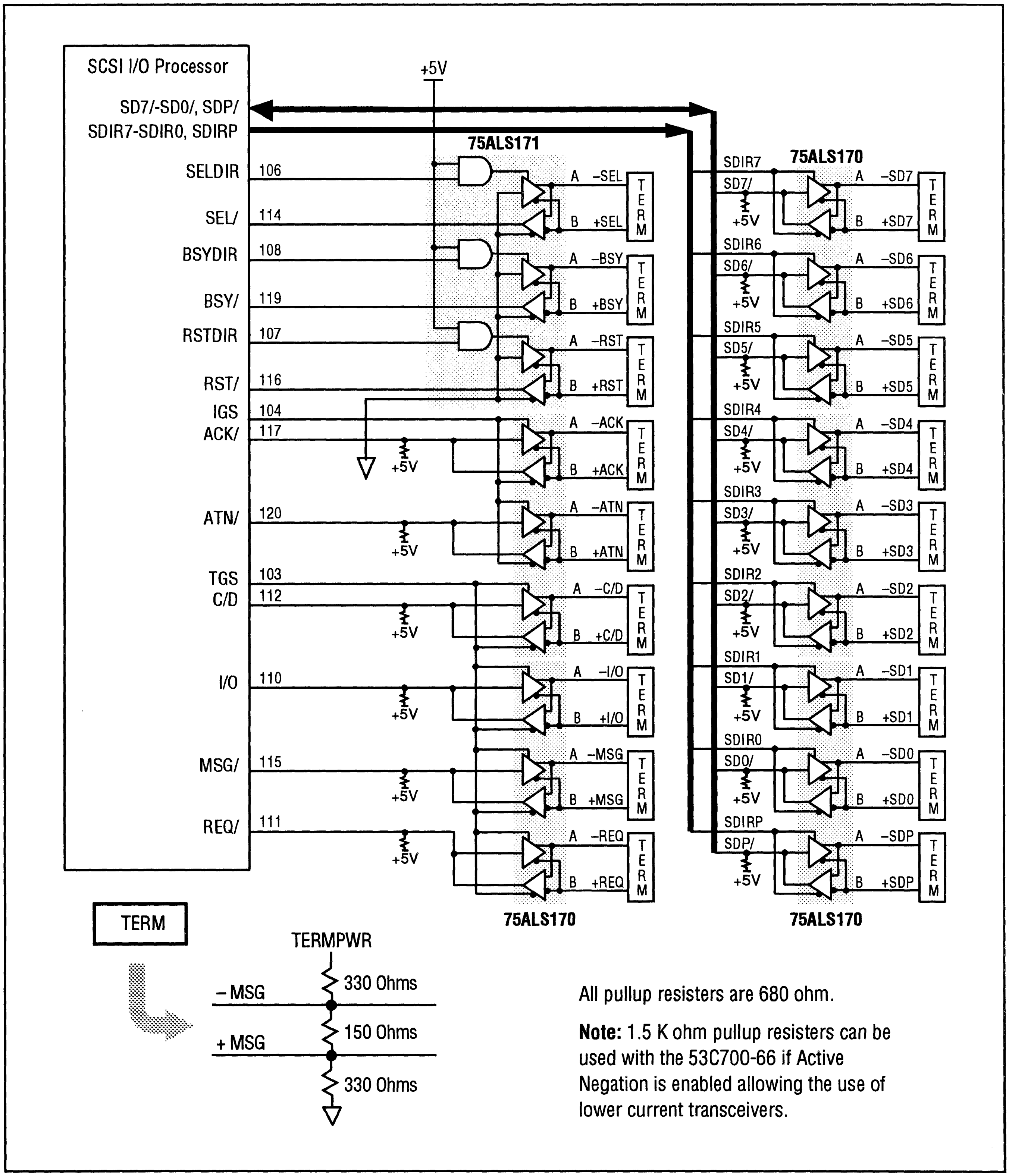


Figure 6-3. Differential SCSI Interface



All pullup resisters are 680 ohm.

Note: 1.5 K ohm pullup resisters can be used with the 53C700-66 if Active Negation is enabled allowing the use of lower current transceivers.

How to use Differential SCSI Interface

In differential mode, the SDIR(7-0), SDIRP, IGS, TGS, RSTDIR, BSYDIR, and SELDIR signals control the direction of external differential-pair transceivers. See Figure 6-3 for the suggested differential wiring diagram. The wiring diagram shows 75ALS170 3-channel transceivers and one 75ALS171 3-channel transceiver, though other single and multi-channel devices may be used (DS36954 4-channel transceiver, for instance). The suggested value for the 15 pull-up resistors in the diagram is 680 ohm. If Active Negation is enabled and the chip is operating in differential mode, the value of the pull-up resistors should be 1.5 K ohms. Resistors are necessary on SCSI REQ, ACK, Data and Parity signals regardless of the state of Active Negation.

How to Terminate the SIOP Device

The terminators provide the biasing needed to "pull" the inactive signal to an appropriate inactive voltage level ~ 3.0 V. The terminators do not need to be present on every SCSI board; but the terminators must be installed at each end of the SCSI cable. The terminator location is dependent on the equipment set-up. Most SCSI boards should provide a means of accommodating terminators. The terminator receptacles should be sockets, so that if not needed the terminators may be removed. No system should ever have more than 2 sets of terminators installed and activate. If more than 2 sets of terminators are active, then the impedance and inactive SCSI voltage levels may not be correct.

Chapter Seven
Electrical Specifications

DC Characteristics

Absolute Maximum Stress Ratings

Symbol	Parameter	Min	Max	Units	Conditions
T_{STG}	Storage temperature	– 55	150	°C	-
V_{DD}	Supply voltage	– 0.5	7.0	V	-
V_{IN}	Input voltage	VSS– 0.5	VDD + 0.5	V	-
ESD*	Electrostatic discharge sensitivity	-	2 K	V	-

* Test using the human body model, 100 pF at 1.5 K ohms

Operating Conditions

Symbol	Parameter	Min		Max		Units
		700	700-66	700	700-66	
V_{DD}	Supply voltage	4.75	4.75	5.25	5.25	V
I_{DD}	Supply current (Dynamic)	-	-	50.0	75.0	mA
I_{DD}	Supply current (Static)	-		5.0	1.0	mA
T_A	Operating temp free-air	0.0	0.0	70.0	70.0	°C
Θ_{JA}	Thermal Resistance from junction to ambient air	51.8	51.8	-	-	°C/W

SCSI Signals – SD(7-0)/, SDP/, REQ/, MSG/, I/O, C/D, ATN/, ACK/, BSY/, SEL/, RST/ – Open Drain

Symbol	Parameter	Min	Max	Units	Conditions
V_{IH}	Input high voltage	2.0	VDD + 0.5	V	-
V_{IL}	Input low voltage	VSS – 0.5	0.8	V	-
V_{OH}	Output high voltage *	2.5		V	IOH = -2.5 mA
V_{OL}	Output low voltage	VSS	0.4	V	IOL = 48 mA
V_{HYS}	Hysteresis	300	-	mV	-
I_{IN}	Input leakage current	– 10	10	µA	-
–	Input leakage – SCSI RST **	– 400	10	µA	-
I_{OZ}	Output leakage current	– 10	10	µA	-
C_{IN}	Input capacitance	-	10	pF	-

* Denotes additional specification on the SD(7-0)/, SDP/, and ACK/ pads with active deassertion (CTEST8, bit 4) enabled. This applies to the 53C700-66 only.

** This is only true on the 53C700-66

SCSI Direction Control Signals – SDIR(7-0), SDIRP, BSYDIR, SELDIR, RSTDIR, TGS, IGS – Totem Pole

Symbol	Parameter	Min	Max	Units	Conditions
V_{OH}	Output high voltage	2.4	VDD	V	IOH = 8
V_{OL}	Output low voltage	VSS	0.4	V	IOL = 8
I_{OH}	Output high current	– 4.0	-	mA	VOH = VDD – 0.5 V
I_{OL}	Output low current	8.0	-	mA	VOL = 0.4 V
I_{OZ}	Output leakage current	– 10	10	µA	-
C_{IN}	Input capacitance	-	10	pF	-

Input Signals – HCS/, HLDAI, NA/, RESET, HLDREQ, READYI/, SCLK

Symbol	Parameter	Min	Max	Units	Conditions
V_{IH}	Input high voltage	2.0	VDD + 0.5	V	-
V_{IL}	Input low voltage	VSS – 0.5	0.8	V	-
I_{IN}	Input leakage current	– 10	10	µA	-
C_{IN}	Input capacitance	-	10	pF	-

CLK Input Signal

Symbol	Parameter	Min	Max	Units	Conditions
V_{IH}	Input high voltage	3.85	VDD + 0.5	V	-
V_{IL}	Input low voltage	VSS – 0.5	1.0	V	-
I_{IN}	Input leakage current	– 10	10	µA	-
C_{IN}	Input capacitance	-	10	pF	-

Output Signals – HOLD, HLDAO, MIO/, DC/, READYO/, MASTER/, FETCH/ – Totem Pole

Symbol	Parameter	Min	Max	Units	Conditions
V_{OH}	Output high voltage	2.4	VDD	V	IOH = – 8.0 mA
V_{OL}	Output low voltage	VSS	0.4	V	IOL = 8.0 mA
I_{OH}	Output high current	– 4.0	-	mA	VOH = VDD – 0.5 V
I_{OL}	Output low current	8.0	-	mA	VOL = 0.4 V
I_{OZ}	Output leakage current	– 10	10	µA	-
C_{OUT}	Pin capacitance	-	10	pF	-

Output Signal – IRQ/ – Open Drain

Symbol	Parameter	Min	Max	Units	Conditions
V_{OL}	Output low voltage	VSS	0.4	V	IOL = 8.0 mA
I_{OL}	Output low current	8.0	-	mA	VOL = 0.4 V
I_{OZ}	Output leakage current	– 400	10	µA	-
C_{OUT}	Pin capacitance	-	10	pF	-

Bi-Directional Signals – R_W/, ADS, A5-A2, D31-D0, DP3_ABRT/, DP2-DP0, BE0/_A0, BE1/_A1, BE2/_BHE/, BE3/ – Tri-state

Symbol	Parameter	Min	Max	Units	Conditions
V_{IH}	Input high voltage	2.0	VDD + 0.5	V	-
V_{IL}	Input low voltage	VSS – 0.5	0.8	V	-
V_{OH}	Output high voltage	2.4	VDD	V	IOH = – 8.0 mA
V_{OL}	Output low voltage	VSS	0.4	V	IOL = 8.0 mA
I_{OH}	Output high current	– 4.0	-	mA	VOH = VDD – 0.5 V
I_{OL}	Output low current	8.0	-	mA	VOL = 0.4 V
I_{IN}	Input leakage current	– 10.0	10.0	µA	-
I_{OZ}	Output leakage current	– 10.0	10.0	µA	-
C_{IN}	Input capacitance	-	10.0	pF	-

Bi-Directional Signals – A31-A6 – Tri-state

Symbol	Parameter	Min	Max	Units	Conditions
V_{OH}	Output high voltage	2.4	VDD	V	IOH = – 8.0 mA
V_{OL}	Output low voltage	VSS	0.4	V	IOL = 8.0 mA
I_{OH}	Output high current	– 4.0	-	mA	VOH = VDD – 0.5 V
I_{OL}	Output low current	8.0	-	mA	VOL = 0.4 V
I_{OZ}	Output leakage current	– 10.0	10.0	µA	-
C_{OUT}	Input capacitance	-	10.0	pF	-

NCR TolerANT Active Negation Technology Electrical Characteristics (53C700-66 Only)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Units
V_{OH}	Output high voltage	I_{OH} = 2.5 mA	2.5	3.1	3.5	V
V_{OL}	Output low voltage	I_{OL} = 48 mA	0.1	0.2	0.5	V
V_{IH}	Input high voltage		2.0		7.0	V
V_{IL}	Input low voltage	Referenced to V_{SS}	-0.5		0.8	V
V_{IK}	Input clamp voltage	V_{DD} = min; I_I = -20 mA Figure 6-27	-0.66	-0.74	-0.77	V
V_{TH}	Threshold, high to low		1.1	1.2	1.3	V
V_{TL}	Threshold, low to high		1.5	1.6	1.7	V
V_{TH} - V_{TL}	Hysteresis	Figure 6-26	300	350	400	mV
I_{OH}[1]	Output high current	V_{OH} = 2.5 Volts	2.5	15	24	mA
I_{OL}	Output low current	V_{OL} = 0.5 Volts	100	150	200	mA
I_{OSH}[1]	Short-circuit output high current	Output driving low, pin shorted to V_{DD} supply[2]			625	mA
I_{OSL}	Short-circuit output current, low	Output driving high, pin shorted to V_{SS} supply			95	mA
I_{LH}	Input high leakage	-0.5 < V_{DD} < 5.25 V_{PIN} = 2.7 V		0.05	10	µA
I_{LL}	Input low leakage	-0.5 < V_{DD} < 5.25 V_{PIN} = 0.5V		-0.05	-10	µA
R_I	Input resistance	SCSI pins [3]		20		MΩ
C_P	Capacitance per pin	Quad Flat Pack Package	6	8	10	pF

***Note:** These values are guaranteed by periodic characterization.* [1] Active Negation outputs only: Data, Parity, REQ, ACK; [2] Single pin only; irreversible damage may occur if sustained; [3] SCSI RESET pin has 10kΩ (nominal) pull-up resistor

NCR TolerANT Active Negation Technology Electrical Characteristics (53C700-66 Only), Continued

Symbol	Parameter	Test Conditions	Min	Typ	Max	Units
t_R[1]	Rise time, 10% to 90 %	Figure 6-24	9.7	15.0	18.5	ns
t_F	Fall time, 90% to 10%	Figure 6-24	5.2	8.1	14.7	ns
dV_H/dt	Slew rate, low to high	Figure 6-24	0.15	0.23	0.49	V/ns
dV_L/dt	Slew rate, high to low	Figure 6-24	0.19	0.37	0.67	V/ns
	Electrostatic Discharge	Mil Std 883C; 3015-7	2			kV
	Latch-up		100			mA
	Filter Delay	Figure 6-25	20	25	30	ns
	Extended Filter Delay	Figure 6-25	40	50	60	ns

***Note:** These values are guaranteed by periodic characterization.* [1] Active Negation outputs only: Data, Parity, REQ, ACK; [2] Single pin only; irreversible damage may occur if sustained; [3] SCSI RESET pin has 10kΩ (nominal) pull-up resistor

Figure 7-1. Rise and Fall Time Test Conditions (53C700-66 Only)

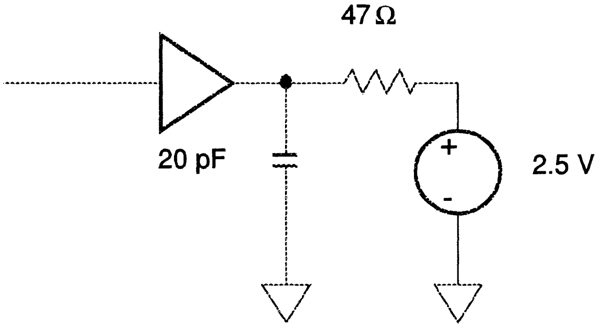
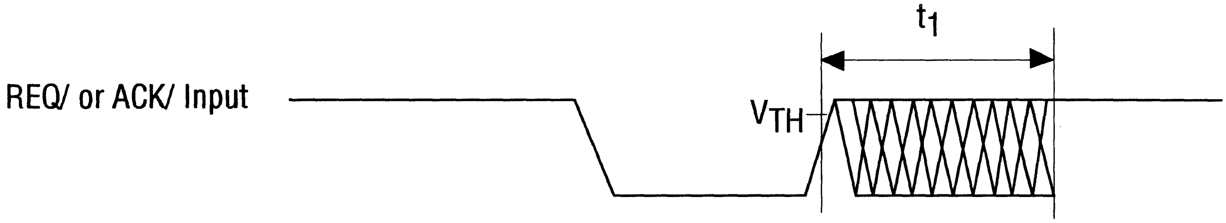


Figure 7-2. SCSI Input Filtering (53C700-66 Only)



t1 = input filtering period, resistor-programmable to either 30 or 60 ns

Figure 7-3. Hysteresis of SCSI Receiver (53C700-66 Only)

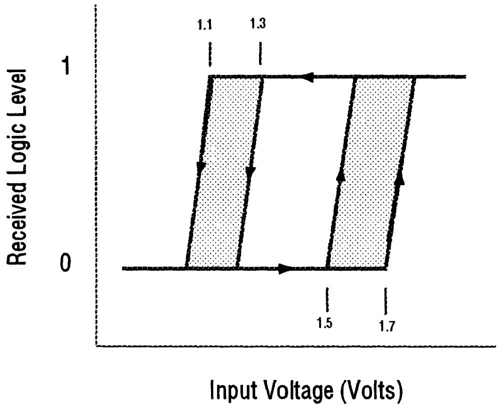


Figure 7-4. Input Current as a Function of Input Voltage (53C700-66 Only)

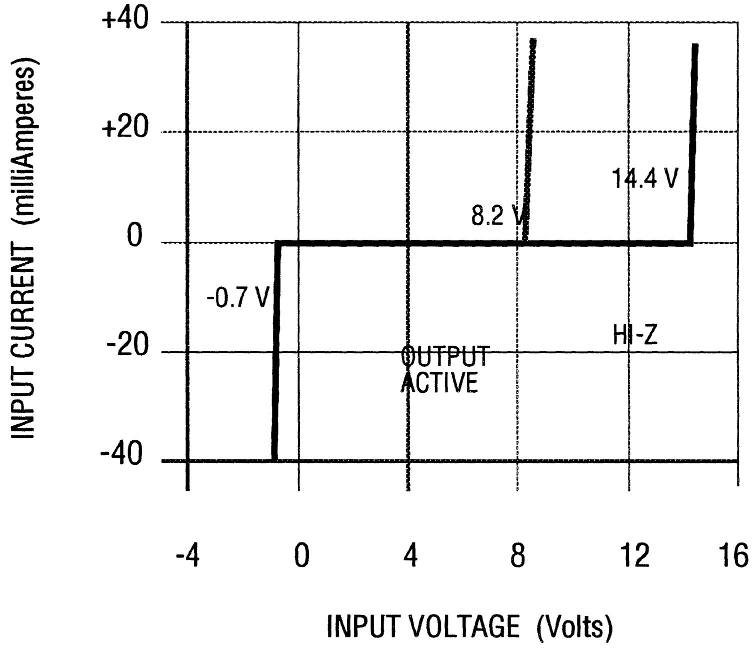
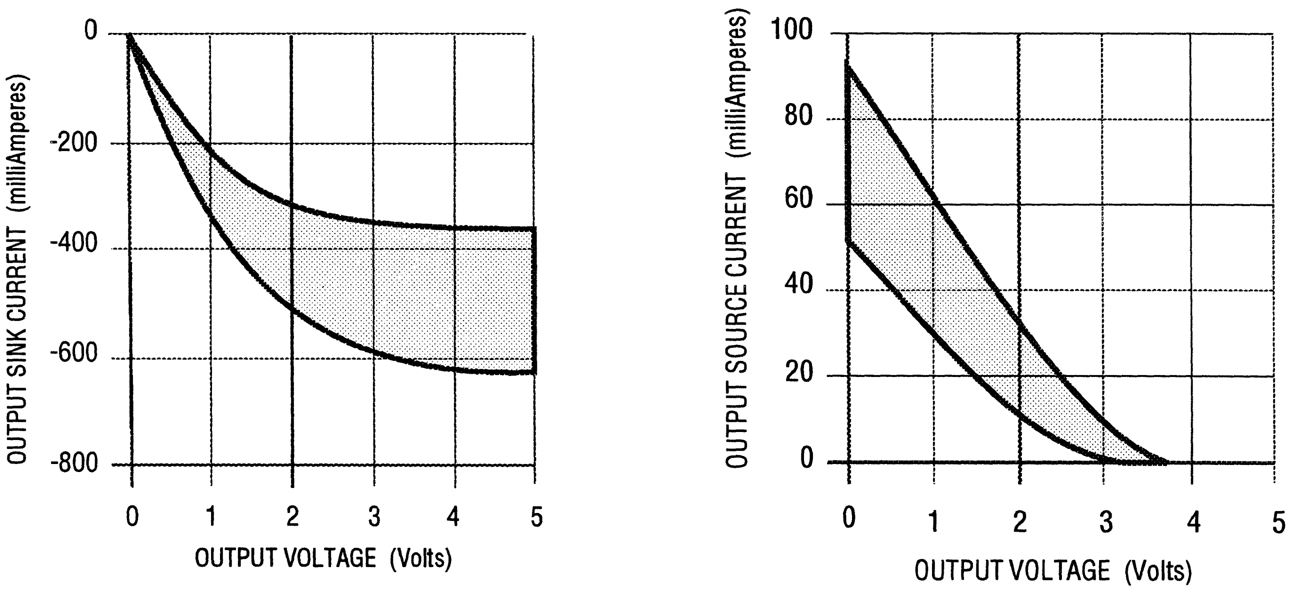


Figure 7-5. Output Current as a Function of Output Voltage (53C700-66 Only)



Output Sink Current as a Function of Output Voltage (I_{OL})

Output Source Current as a Function of Output Voltage (I_{OH})

AC Electrical Characteristics

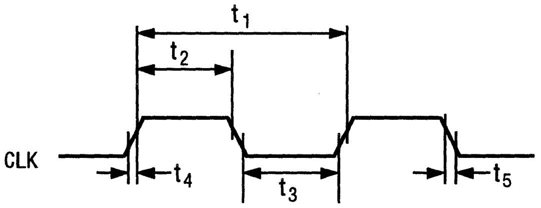
The AC characteristics described in this section apply to the voltage range (VDD) of 4.75 volts to 5.25 volts and the temperature range of 0oC to 70oC. Chip output timings are based on simulation under worst case conditions (4.75 volts, 70oC) and the following pad termination:

Pad Conditions

Signal Name	Output Load
A31-A2, D31-D0, DP3_ABRT/, DP2-DP0, BE3/, BE2/_BHE/, BE1/_A1, BE0/_A0, MASTER/, W_R/, DC/, MIO, ADS/, FETCH/, READIO/, HOLD, HLDAI, IRQ/	50 pf
SD7/-SD0/, SDP, SDIR7-SDIR0, SDIRP, ATN/, MSG/, C/D, I/O, REQ/, ACK/, BSY/, RST/, SEL/, BSYDIR, RSTDIR, SELDIR, TGS, IGS	100 pf

Timings

Figure 7-6. Clock Timing



53C700 Clock Timing

Symbol	Parameter	Min	Max	Units	Conditions
t_1	CLK period	20	60	ns	-
t_2	CLK high (at 3.5 V)	8	-	ns	-
t_3	CLK low (at 1.5 V)	8	-	ns	-
t_4	CLK rise time	1	-	V/ns	-
t_5	CLK fall time	1	-	V/ns	-

53C700-66 Clock Timing

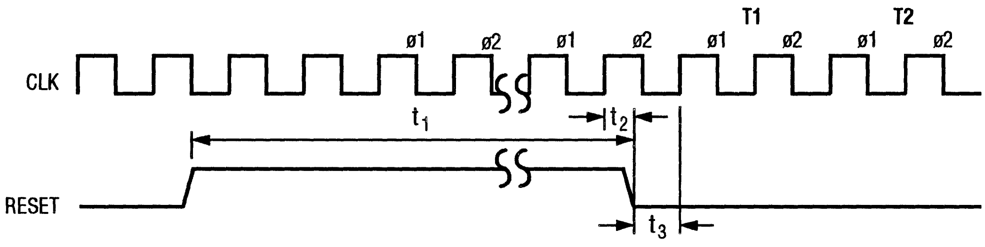
Symbol	Parameter	Min	Max	Units	Conditions
t_1	CLK period	15	60	ns	-
t_2	CLK high (at 3.5 V)	6	-	ns	-
t_3	CLK low (at 1.5 V)	6	-	ns	-
t_4	CLK rise time	1	-	V/ns	-
t_5	CLK fall time	1	-	V/ns	-

53C700-66 SCSI Clock Timing

Symbol	Parameter	Min	Max	Units	Conditions
t_1	SCSI clock cycle time (t_{SCLK})*	15	60	ns	-
t_2	SCLK low time**	7	-	ns	-
t_3	SCLK high time**	7	-	ns	-
t_4	SCLK slew rate	1	-	V/ns	-

CTEST8 bit 7 enables this clock.
* This parameter must be met to insure SCSI timings are within specification.
** Duty cycle not to exceed 60/40.

Figure 7-7. Reset and Clock Synchronization



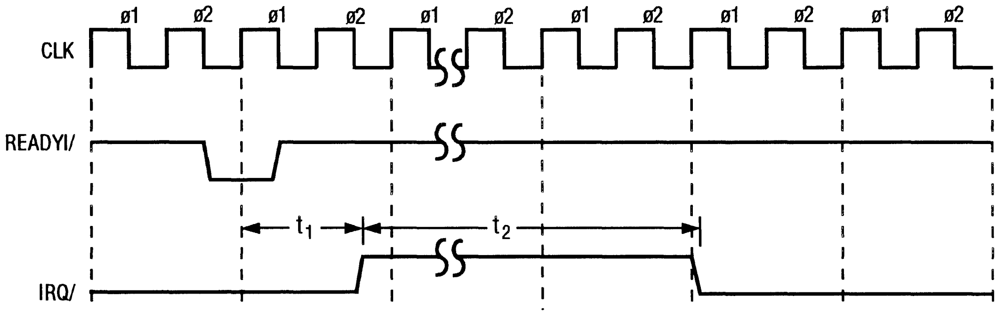
53C700 Reset and Clock Synchronization

Symbol	Parameter	Min	Max	Units	Conditions
t_1	Reset pulse width	5	-	CLK	-
t_2	Reset hold from CLK rising edge	3	-	ns	-
t_3	Reset inactive to CLK rising edge	10	-	ns	-

53C700-66 Reset and Clock Synchronization

Symbol	Parameter	Min	Max	Units	Conditions
t_1	Reset pulse width	5	-	CLK	-
t_2	Reset hold from CLK rising edge	3	-	ns	-
t_3	Reset inactive to CLK rising edge	5	-	ns	-

Figure 7-8. Interrupt Output Timong



Note: IRQ/ is deasserted after a read of the DSTAT or SSTAT0 register. This occurs within t_1 from the rising CLK edge with READYI/ active. The READYI/ timing is shown in the Slave Mode Read Cycle. When multiple or stacked interrupts occur, IRQ/ is asserted for a minimum of three CLKs.

53C700 Interrupt Output Timing

Symbol	Parameter	Min	Max	Units	Conditions
t_1	IRQ/ deasserted from CLK rising	5	120*	ns	-
t_2	IRQ/ deasserted to IRQ/ asserted	6	-	CLK	-

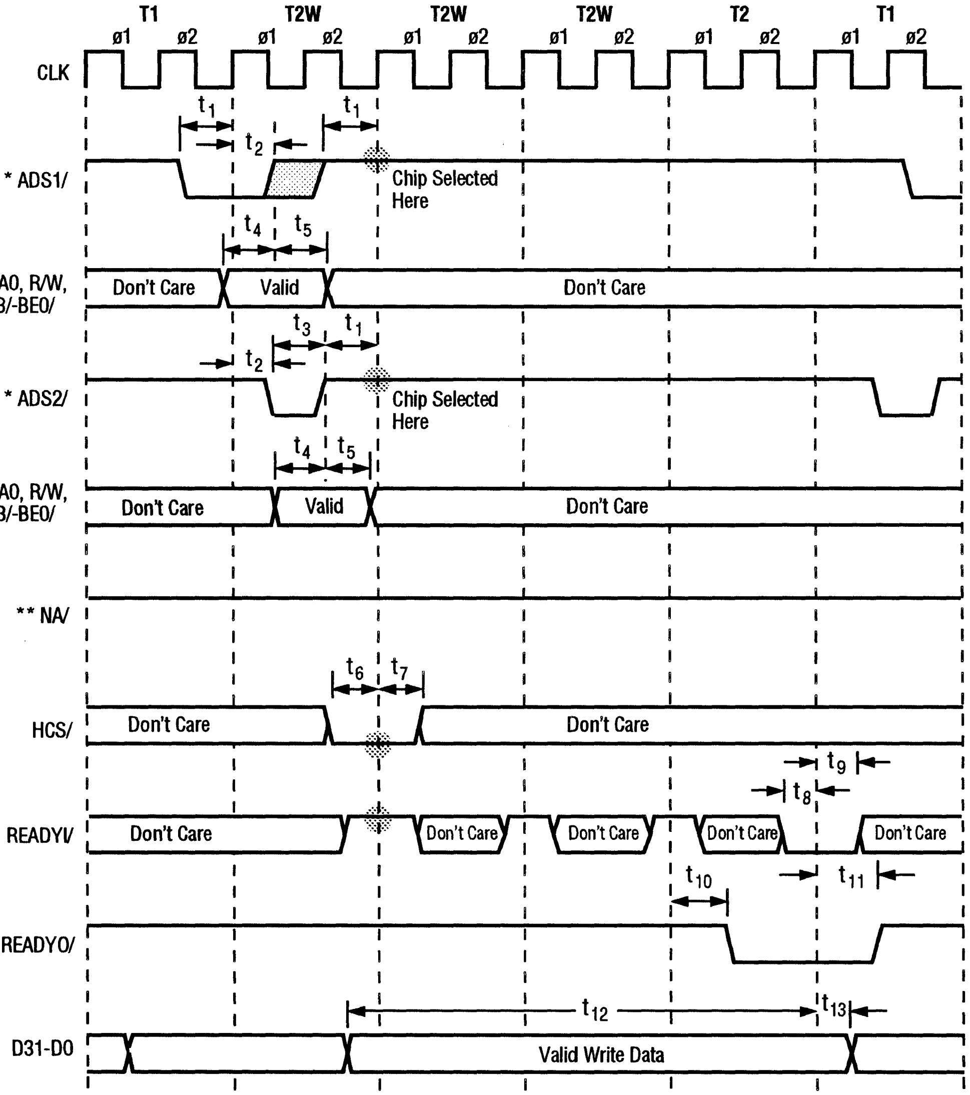
* This can be shortened by adding an external pullup resistor.

53C700-66 Interrupt Output Timing

Symbol	Parameter	Min	Max	Units	Conditions
t_1	IRQ/ deasserted from CLK rising	5	120*	ns	-
t_2	IRQ/ deasserted to IRQ/ asserted	6	-	CLK	-

* This can be shortened by adding an external pullup resistor.

Figure 7-9. Slave Mode Write Cycle (register write)



* The SCSI I/O processor is enabled as a slave when HCS/ is sampled active on the first ø1 rising clock edge after an ADS/ low active pulse.

* ADS1/ and ADS2/ are two possible timings of ADS/.

** NA/ is not sampled during a slave cycle. The 53C700 does not support next addressing in slave mode.

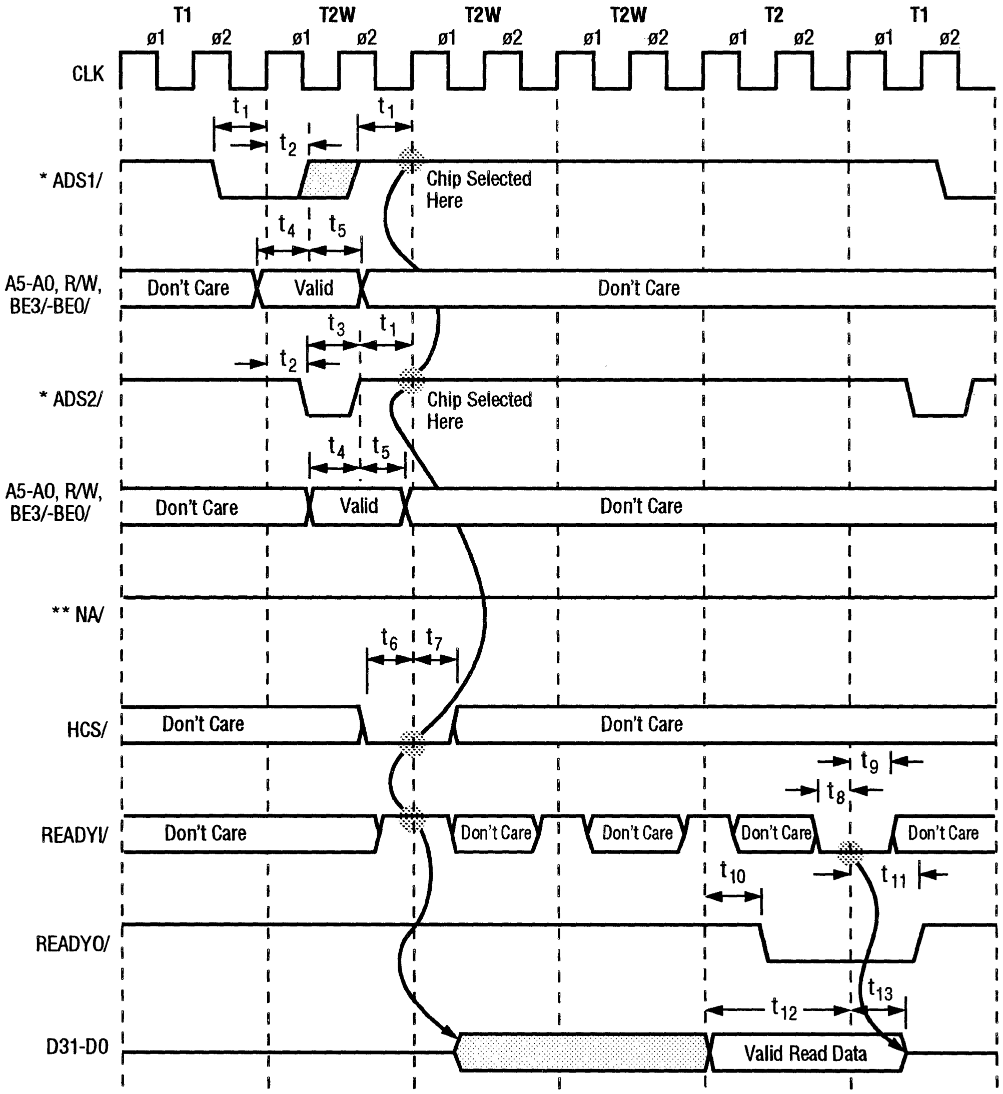
53C700 Slave Mode Write Cycle (register write)

Symbol	Parameter	Min	Max	Units
t_1	ADS1/ setup or ADS2/ deasserted to CLK rising edge	7	-	ns
t_2	ADS1/ hold or ADS2/ asserted from CLK rising edge	5	-	ns
t_3	ADS2/ pulse width	15	-	ns
t_4	Control and address setup to ADS/ deasserted	10	-	ns
t_5	Control and address hold from ADS/ deasserted	10	-	ns
t_6	HCS/ setup to CLK rising edge	7	-	ns
t_7	HCS/ hold from CLK rising edge	5	-	ns
t_8	READYI/ setup to CLK rising edge	9	-	ns
t_9	READYI/ hold from CLK rising edge	6	-	ns
t_{10}	READYO/ valid from CLK rising edge	2	17	ns
t_{11}	READYO/ hold from CLK rising edge	3	19	ns
t_{12}	Data setup to CLK rising edge	6	-	CLK
t_{13}	Data hold from CLK rising edge	20	-	ns

53C700-66 Slave Mode Write Cycle (register write)

Symbol	Parameter	Min	Max	Units
t_1	ADS1/ setup or ADS2/ deasserted to CLK rising edge	7	-	ns
t_2	ADS1/ hold or ADS2/ asserted from CLK rising edge	5	-	ns
t_3	ADS2/ pulse width	15	-	ns
t_4	Control and address setup to ADS/ deasserted	10	-	ns
t_5	Control and address hold from ADS/ deasserted	10	-	ns
t_6	HCS/ setup to CLK rising edge	4	-	ns
t_7	HCS/ hold from CLK rising edge	4	-	ns
t_8	READYI/ setup to CLK rising edge	7	-	ns
t_9	READYI/ hold from CLK rising edge	5	-	ns
t_{10}	READYO/ valid from CLK rising edge	2	15	ns
t_{11}	READYO/ hold from CLK rising edge	3	15	ns
t_{12}	Data setup to CLK rising edge	6	-	CLK
t_{13}	Data hold from CLK rising edge	20	-	ns

Figure 7-10. Slave Mode Read Cycle (register read)



* The SCSI I/O processor is enabled as a slave when HCS/ is sampled active on the first ø1 rising clock edge after an ADS/ low active pulse.

* ADS1/ and ADS2/ are two possible timings of ADS/.

** NA/ is not sampled during a slave cycle. The 53C700 does not support next addressing in slave mode.

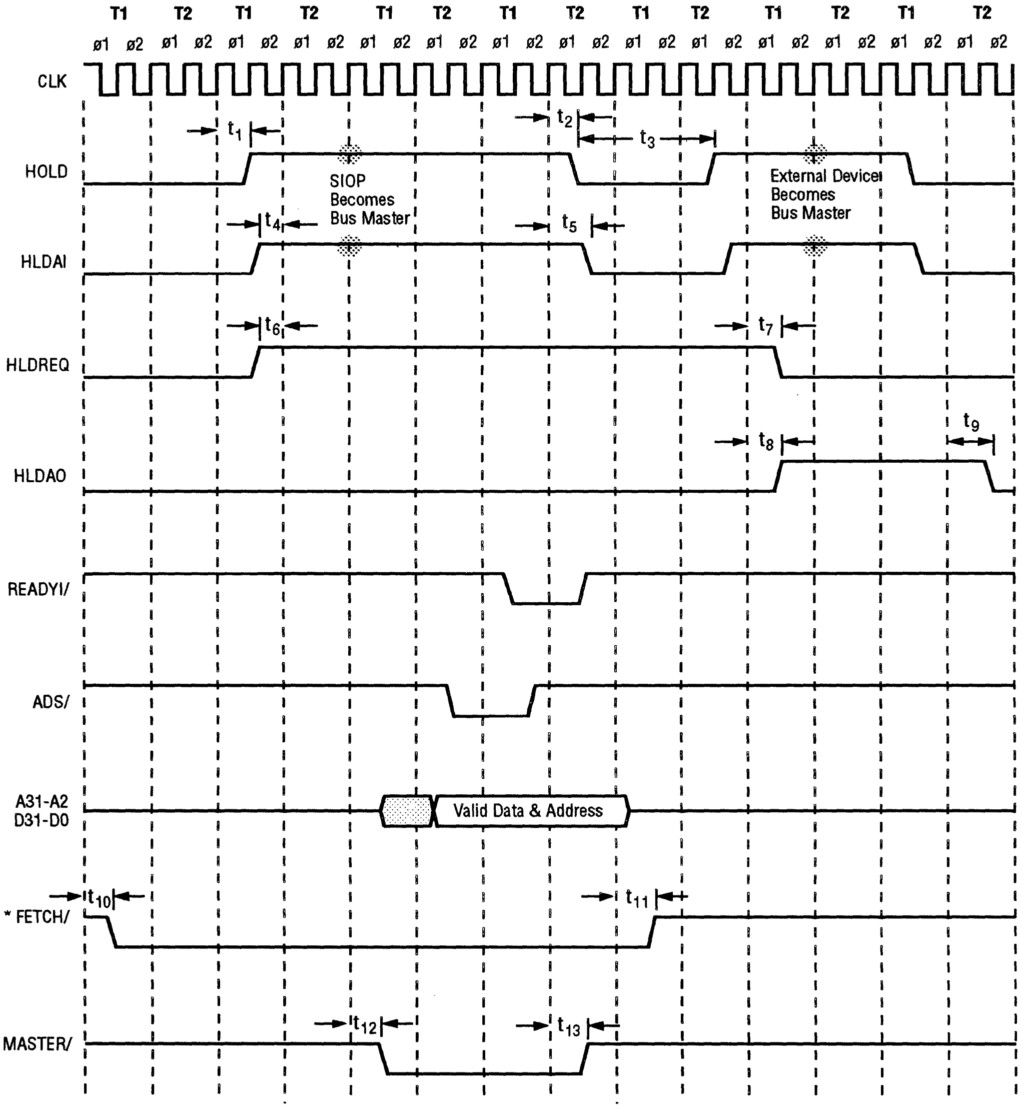
53C700 Slave Mode Read Cycle (register read)

Symbol	Parameter	Min	Max	Units
t_1	ADS1/ setup or ADS2/ deasserted to CLK rising edge	7	-	ns
t_2	ADS1/ hold or ADS2/ asserted from CLK rising edge	5	-	ns
t_3	ADS/ pulse width	15	-	ns
t_4	Control and address setup to ADS/ deasserted	10	-	ns
t_5	Control and address hold from ADS/ deasserted	10	-	ns
t_6	HCS/ setup to CLK rising edge	7	-	ns
t_7	HCS/ hold from CLK rising edge	5	-	ns
t_8	READYI/ setup to CLK rising edge	9	-	ns
t_9	READYI/ hold from CLK rising edge	6	-	ns
t_{10}	READYO/ valid from CLK rising edge	2	17	ns
t_{11}	READYO/ hold from CLK rising edge	3	19	ns
t_{12}	Data setup to CLK rising edge	2	-	CLK
t_{13}	Data hold from CLK rising edge	5	25	ns

53C700-66 Slave Mode Read Cycle (register read)

Symbol	Parameter	Min	Max	Units
t_1	ADS1/ setup or ADS2/ deasserted to CLK rising edge	7	-	ns
t_2	ADS1/ hold or ADS2/ asserted from CLK rising edge	5	-	ns
t_3	ADS/ pulse width	15	-	ns
t_4	Control and address setup to ADS/ deasserted	10	-	ns
t_5	Control and address hold from ADS/ deasserted	10	-	ns
t_6	HCS/ setup to CLK rising edge	4	-	ns
t_7	HCS/ hold from CLK rising edge	4	-	ns
t_8	READYI/ setup to CLK rising edge	7	-	ns
t_9	READYI/ hold from CLK rising edge	5	-	ns
t_{10}	READYO/ valid from CLK rising edge	2	15	ns
t_{11}	READYO/ hold from CLK rising edge	3	15	ns
t_{12}	Data setup to CLK rising edge	2	-	CLK
t_{13}	Data hold from CLK rising edge	5	25	ns

Figure 7-11. Bus Arbitration



* Output signal is valid only for 53C700-66

Note: The SIOP becomes bus master on the next ø1 CLK edge after HOLDAI is sampled active. HLDAI and HLDEEQ are asynchronous inputs; timings given are for minimum synchronous delays. The SIOP waits five to eight T states from deasserting HOLD to asserting HOLD (bus master). This allows for automatic fairness. If an external device requests the bus via HLDREQ, the SIOP waits only two T states from deasserting HOLD to asserting HOLD.

53C700 Bus Arbitration

Symbol	Parameter	Min	Max	Units
t_1	HOLD valid from CLK rising edge	3	25	ns
t_2	HOLD hold from CLK rising edge	4	30	ns
t_3	HOLD inactive to HOLD active	2	-	CLK
t_4	HLDAI setup to CLK rising edge	3	-	ns
t_5	HLDAI hold from CLK rising edge	3	-	ns
t_6	HLDREQ setup to CLK rising edge	3	-	ns
t_7	HLDREQ hold from CLK rising edge	3	-	ns
t_8	HLDAO active from CLK rising edget	3	25	ns
t_9	HLDAO inactive from CLK rising edge	3	25	ns

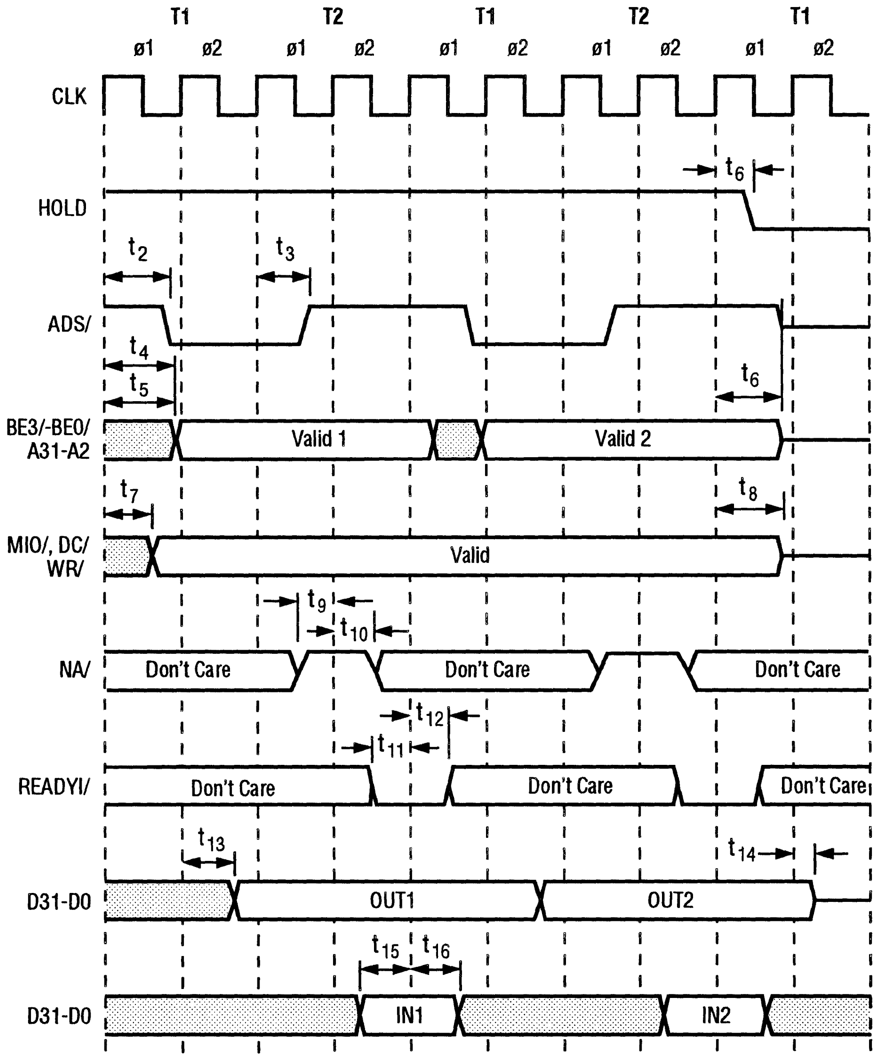
Note: The 53C700 will periodically assert the HOLD signal and receive a SCSI interrupt at the same time. When this happens the chip will wait for the HLDAI signal to complete the normal bus arbitration handshake. The chip no longer wants host bus access, it deasserts the HOLD, MASTER/, and all control lines, after one BCLK, and does NOT assert ADS/, the signal that indicates a valid bus cycle is starting. The chip will next generate an interrupt, which the system may then service.

53C700-66 Bus Arbitration

Symbol	Parameter	Min	Max	Units
t_1	HOLD valid from CLK rising edge	3	19	ns
t_2	HOLD hold from CLK rising edge	4	19	ns
t_3	HOLD inactive to HOLD active	2	-	CLK
t_4	HLDAI setup to CLK rising edge	5	-	ns
t_5	HLDAI hold from CLK rising edge	5	-	ns
t_6	HLDREQ setup to CLK rising edge	5	-	ns
t_7	HLDREQ hold from CLK rising edge	5	-	ns
t_8	HLDAO active from CLK rising edget	3	19	ns
t_9	HLDAO inactive from CLK rising edge	3	19	ns
t_{10}	FETCH/ low from CLK rising edge	8	38	ns
t_{11}	FETCH/ high from CLK rising edge	9	41	ns
t_{12}	MASTER/ low from CLK rising edge	7	33	ns
t_{13}	MASTER/ high from CLK rising edge	7	35	ns

Note: The 53C700 will periodically assert the HOLD signal and receive a SCSI interrupt at the same time. When this happens the chip will wait for the HLDAI signal to complete the normal bus arbitration handshake. The chip no longer wants host bus access, it deasserts the HOLD, MASTER/, and all control lines, after one BCLK, and does NOT assert ADS/, the signal that indicates a valid bus cycle is starting. The chip will next generate an interrupt, which the system may then service.

Figure 7-12. Bus Master Memory Read and Write Cycle - Non-Pipelined Next Address



53C700 Bus Master Memory Read and Write Cycle – Non-Pipelined Next Address

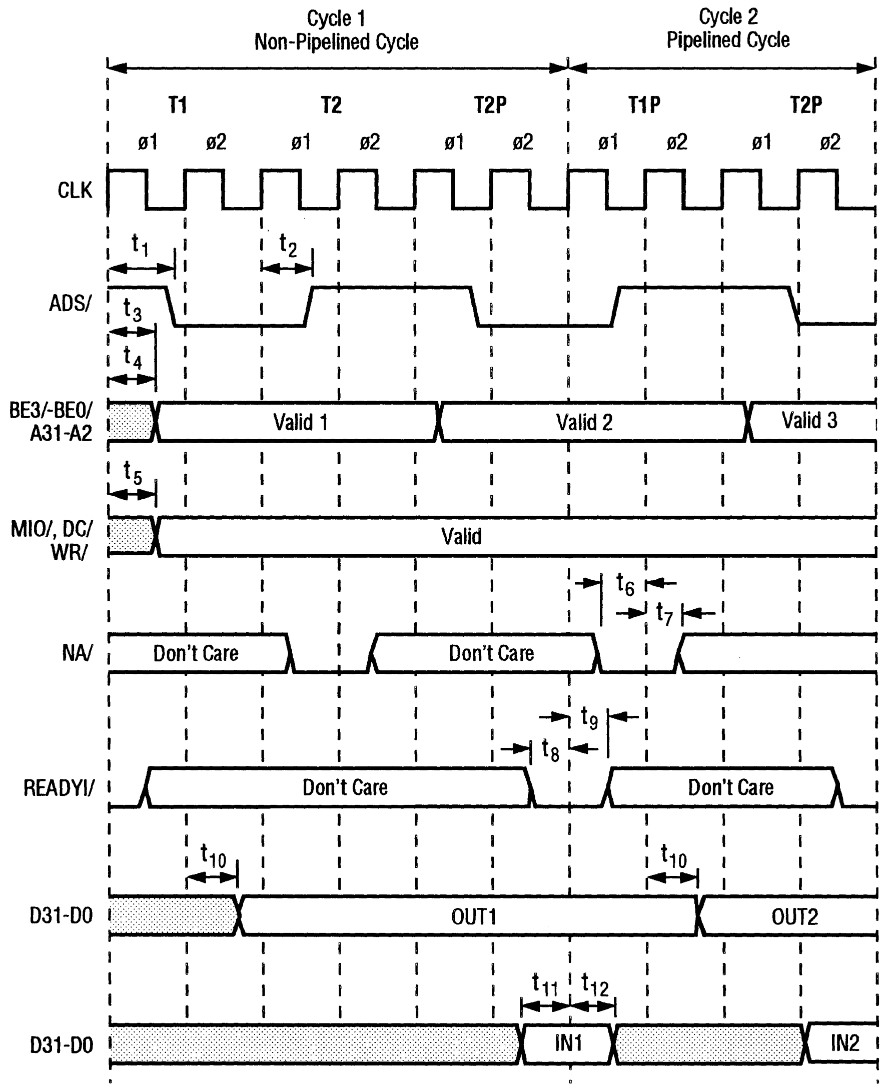
Symbol	Parameter	Min	Max	Units
t_1	HOLD hold from CLK rising edge	4	-	ns
t_2	ADS/ valid from CLK rising edge	4	21	ns
t_3	ADS/ hold from CLK rising edge	4	30	ns
t_4	Address valid from CLK rising edge	4	23	ns
t_5	BE3/-BE0/ valid from CLK rising edge	4	24	ns

Symbol	Parameter	Min	Max	Units
t_6	ADS/, Address & BE3/-BE0/ tristate from CLK rising edge	4	30	ns
t_7	MIO/, DC/, WR/ valid from CLK rising edge	-	4	ns
t_8	MIO/, DC/, WR/ tristate from CLK rising edge	4	30	ns
t_9	NA/ setup to CLK rising edge	7	-	ns
t_{10}	NA/ hold from CLK rising edge	3	-	ns
t_{11}	READYI/ setup to CLK rising edge	9	-	ns
t_{12}	READYI/ hold from CLK rising edge	6	-	ns
t_{13}	Write data setup to CLK rising edge	5	27	ns
t_{14}	Write data tristate from CLK rising edge	2	22	ns
t_{15}	Read data setup to CLK rising edge	7	-	ns
t_{16}	Read data hold from CLK rising edge	5	-	ns

53C700-66 Bus Master Memory Read and Write Cycle – Non-Pipelined Next Address

Symbol	Parameter	Min	Max	Units
t_1	HOLD hold from CLK rising edge	3	-	ns
t_2	ADS/ valid from CLK rising edge	4	16	ns
t_3	ADS/ hold from CLK rising edge	4	17	ns
t_4	Address valid from CLK rising edge	4	21	ns
t_5	BE3/-BE0/ valid from CLK rising edge	4	21	ns
t_6	ADS/, Address & BE3/-BE0/ tristate from CLK rising edge	4	26	ns
t_7	MIO/, DC/, WR/ valid from CLK rising edge	-	4	ns
t_8	MIO/, DC/, WR/ tristate from CLK rising edge	4	28	ns
t_9	NA/ setup to CLK rising edge	7	-	ns
t_{10}	NA/ hold from CLK rising edge	3	-	ns
t_{11}	READYI/ setup to CLK rising edge	7	-	ns
t_{12}	READYI/ hold from CLK rising edge	5	-	ns
t_{13}	Write data setup to CLK rising edge	5	24	ns
t_{14}	Write data tristate from CLK rising edge	2	20	ns
t_{15}	Read data setup to CLK rising edge	7	-	ns
t_{16}	Read data hold from CLK rising edge	5	-	ns

Figure 7-13. Bus Master Memory Read and Write Cycle - Pipelined Next Address



* The time to tristate for address, BE3/-BE0/, MIO/, DC/, R/W and data is the same as shown in the Non-Pipelined timings.

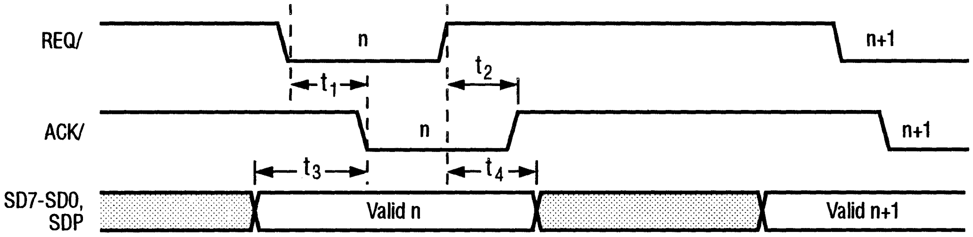
53C700 Bus Master Memory Read and Write Cycle – Pipelined Next Address

Symbol	Parameter	Min	Max	Units
t_1	ADS/ valid from CLK rising edge	4	21	ns
t_2	ADS/ hold from CLK rising edge	4	30	ns
t_3	Address valid from CLK rising edge	4	23	ns
t_4	BE3/-BE0/ valid from CLK rising edge	4	24	ns
t_5	MIO/, DC/, WR/ valid from CLK rising edge	-	4	ns
t_6	NA/ setup to CLK rising edge	7	-	ns
t_7	NA/ hold from CLK rising edge	3	-	ns
t_8	READYI/ setup to CLK rising edge	9	-	ns
t_9	READYI/ hold from CLK rising edge	6	-	ns
t_{10}	Data valid from CLK rising edge	5	27	ns
t_{11}	Data setup to CLK rising edge	7	-	ns
t_{12}	Data hold from CLK rising edge	5	-	ns

53C700-66 Bus Master Memory Read and Write Cycle – Pipelined Next Address

Symbol	Parameter	Min	Max	Units
t_1	ADS/ valid from CLK rising edge	4	16	ns
t_2	ADS/ hold from CLK rising edge	4	17	ns
t_3	Address valid from CLK rising edge	4	21	ns
t_4	BE3/-BE0/ valid from CLK rising edge	4	21	ns
t_5	MIO/, DC/, WR/ valid from CLK rising edge	-	4	ns
t_6	NA/ setup to CLK rising edge	7	-	ns
t_7	NA/ hold from CLK rising edge	3	-	ns
t_8	READYI/ setup to CLK rising edge	7	-	ns
t_9	READYI/ hold from CLK rising edge	5	-	ns
t_{10}	Data valid from CLK rising edge	5	24	ns
t_{11}	Data setup to CLK rising edge	7	-	ns
t_{12}	Data hold from CLK rising edge	5	-	ns

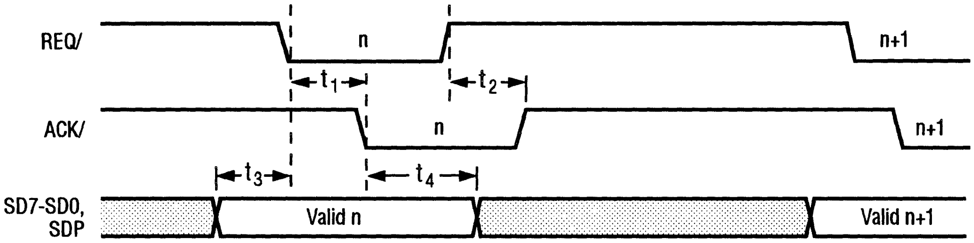
Figure 7-14. Initiator Asynchronous Send



53C700/53C700-66 Initiator Asynchronous Send

Symbol	Parameter	Min	Max	Units
t_1	ACK/ asserted from REQ/ asserted	10	-	ns
t_2	ACK/ deasserted from REQ/ deasserted	10	-	ns
t_3	Data setup to ACK/ asserted	55	-	ns
t_4	Data hold from REQ/ deasserted	20	-	ns

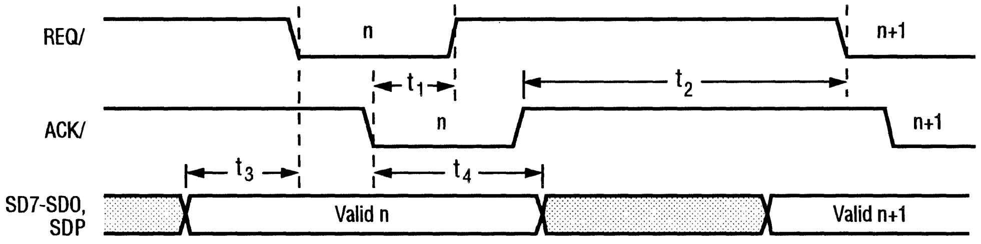
Figure 7-15. Initiator Asynchronous Receive



53C700/53C700-66 Initiator Asynchronous Receive

Symbol	Parameter	Min	Max	Units
t_1	ACK/ asserted from REQ/ asserted	10	-	ns
t_2	ACK/ deasserted from REQ/ deasserted	10	-	ns
t_3	Data setup to REQ/ asserted	0	-	ns
t_4	Data hold from ACK/ deasserted	0	-	ns

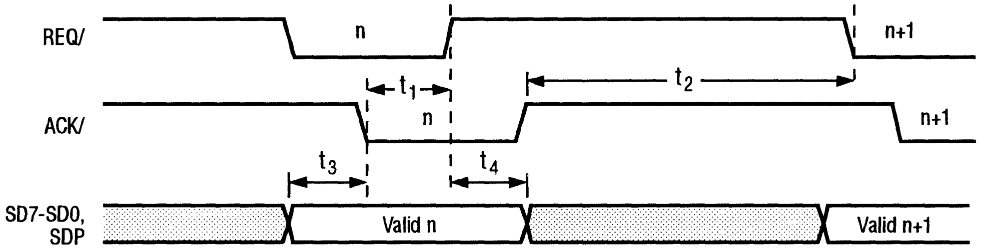
Figure 7-16. Target Asynchronous Send



53C700/53C700-66 Target Asynchronous Send

Symbol	Parameter	Min	Max	Units
t_1	REQ/ deasserted from ACK/ asserted	10	-	ns
t_2	REQ/ asserted from ACK/ deasserted	10	-	ns
t_3	Data setup to REQ/ asserted	55	-	ns
t_4	Data hold from ACK/ asserted	20	-	ns

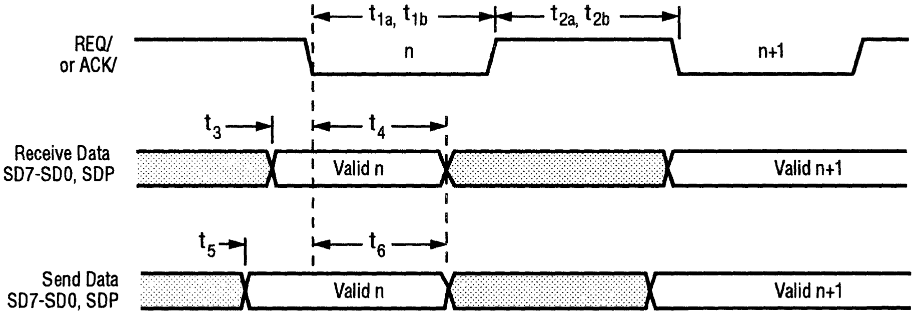
Figure 7-17. Target Asynchronous Receive



53C700/53C700-66 Target Asynchronous Receive

Symbol	Parameter	Min	Max	Units
t_1	REQ/ deasserted from ACK/ asserted	10	-	ns
t_2	REQ/ asserted from ACK/ deasserted	10	-	ns
t_3	Data setup to ACK/ asserted	0	-	ns
t_4	Data hold from REQ/ deasserted	0	-	ns

Figure 7-18. Initiator and Target Synchronous Transfers



SCSI-1 Transfers (Single Ended, 5.0 MB/sec, 50 MHz clock)*

Symbol	Parameter	Min	Max	Units
t_1	Send REQ/ or ACK/ assertion pulse width	90	-	ns
t_2	Send REQ/ or ACK/ deassertion pulse width	90	-	ns
t_1	Receive REQ/ or ACK/ assertion pulse width	90	-	ns
t_2	Receive REQ/ or ACK/ deassertion pulse width	90	-	ns
t_3	Send data setup to REQ/ or ACK/ asserted	55	-	ns
t_4	Send data hold from REQ/ or ACK/ asserted	100	-	ns
t_5	Receive data setup to REQ/ or ACK/ asserted	0	-	ns
t_6	Receive data hold from REQ/ or ACK/ asserted	45	-	ns

SCSI-1 Transfers (Differential, 4.17 MB/sec, 50 MHz clock)*

Symbol	Parameter	Min	Max	Units
t_1	Send REQ/ or ACK/ assertion pulse width	95	-	ns
t_2	Send REQ/ or ACK/ deassertion pulse width	95	-	ns
t_1	Receive REQ/ or ACK/ assertion pulse width	84	-	ns
t_2	Receive REQ/ or ACK/ deassertion pulse width	84	-	ns
t_3	Send data setup to REQ/ or ACK/ asserted	63	-	ns
t_4	Send data hold from REQ/ or ACK/ asserted	110	-	ns
t_5	Receive data setup to REQ/ or ACK/ asserted	0	-	ns
t_6	Receive data hold from REQ/ or ACK/ asserted	45	-	ns

*[(clock period)(2)(transfer period)]$^{-1}$ = MB/sec

SCSI-2 Fast Transfers (10.0 MB/sec, 40 MHz clock) 53C700-66 only

Symbol	Parameter	Min	Max	Units
t_1	Send REQ/ or ACK/ assertion pulse width	35	-	ns
t_2	Send REQ/ or ACK/ deassertion pulse width	35	-	ns
t_1	Receive REQ/ or ACK/ assertion pulse width	24	-	ns
t_2	Receive REQ/ or ACK/ deassertion pulse width	24	-	ns
t_3	Send data setup to REQ/ or ACK/ asserted	33	-	ns
t_4	Send data hold from REQ/ or ACK/ asserted	45	-	ns
t_5	Receive data setup to REQ/ or ACK/ asserted	0	-	ns
t_6	Receive data hold from REQ/ or ACK/ asserted	10	-	ns

Active Negation Enabled (bit 4 in CTEST8)
Use Alternate SCSI clock if desirable

SCSI-2 Fast Transfers (10.0 MB/sec, 50 MHz clock)* 53C700-66 only

Symbol	Parameter	Min	Max	Units
t_1	Send REQ/ or ACK/ assertion pulse width	35	-	ns
t_2	Send REQ/ or ACK/ deassertion pulse width	35	-	ns
t_1	Receive REQ/ or ACK/ assertion pulse width	24	-	ns
t_2	Receive REQ/ or ACK/ deassertion pulse width	24	-	ns
t_3	Send data setup to REQ/ or ACK/ asserted	33	-	ns
t_4	Send data hold from REQ/ or ACK/ asserted	40**	-	ns
t_5	Receive data setup to REQ/ or ACK/ asserted	0	-	ns
t_6	Receive data hold from REQ/ or ACK/ asserted	10	-	ns

Active Negation Enabled (bit 4 in CTEST8)
Use Alternate SCSI clock if desirable

* Transfer period bits (bits 6-4 in SXFER register) are set to zero and the Extra Clock Cycle of Data Setup bit (bit 7 in SCNTL1) set.

** Analysis of system configuration is recommended due to reduced driver skew margin in differential systems.

Appendix A
Register Summary

SCNTL0 Register R/W (00h)

0	Target Mode
1	Assert ATN/ on Parity Error
2	Enable Parity Generation
3	Enable Parity Checking
4	Select with ATN/
5	Start Sequence
6	Arbitration Mode bit 0
7	Arbitration Mode bit 1

SCNTL1 Register R/W (01h)

0	Start SCSI Receive
1	Start SCSI Send
2	Assert Even SCSI Parity
3	Assert SCSI RST/
4	Connected
5	Enable Selection & Reselection
6	Assert contents of SODL on SCSI bus
7	Extra Clock Cycle of Data Setup

SDID Register R/W (02h)

0	ID0
1	ID1
2	ID2
3	ID3
4	ID4
5	ID5
6	ID6
7	ID7

SIEN Register R/W (03h)

0	Parity Error
1	SCSI RST/ Received
2	Unexpected Disconnect
3	SCSI Gross Error
4	Selected or Reselected
5	Selection or Reselection Timeout
6	Function Complete
7	Phase Mismatch or ATN/ Active

SCID Register R/W (04h)

0	ID0
1	ID1
2	ID2
3	ID3
4	ID4
5	ID5
6	ID6
7	ID7

SXFER Register R/W (05h)

0	Synchronous Offset bit 0
1	Synchronous Offset bit 1
2	Synchronous Offset bit 2
3	Synchronous Offset bit 3
4	Synchronous Transfer Period bit 0
5	Synchronous Transfer Period bit 1
6	Synchronous Transfer Period bit 2
7	Disable Halt on a Parity Error or ATN/

SODL Register R/W (06h)

0	SDB0
1	SDB1
2	SDB2
3	SDB3
4	SDB4
5	SDB5
6	SDB6
7	SDB7

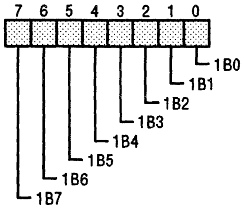
1 = Active
0 = Inactive

SOCL Register R/W (07h)

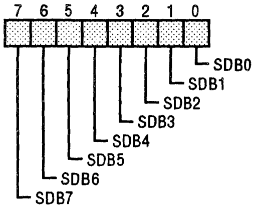
0	I/O
1	C/D
2	MSG/
3	ATN/
4	SEL/
5	BSY/
6	ACK/
7	REQ/

1 = Active
0 = Inactive

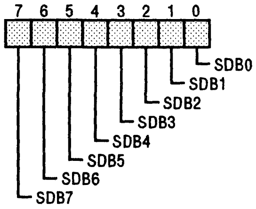
SFBR Register R (08h)



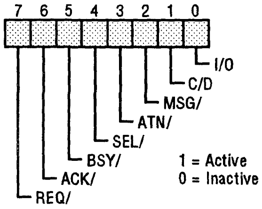
SIDL Register R (09h)



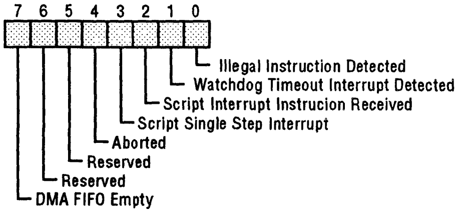
SBDL Register R (0Ah)



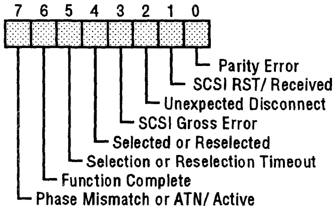
SBCL Register R (0Bh)



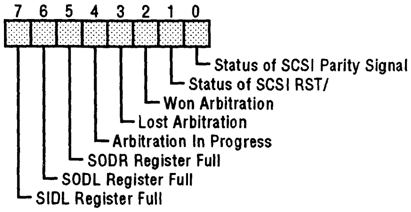
DSTAT Register R (0Ch)


7 6 5 4 3 2 1 0
Illegal Instruction Detected
Watchdog Timeout Interrupt Detected
Script Interrupt Instrucion Received
Script Single Step Interrupt
Aborted
Reserved
Reserved
DMA FIFO Empty

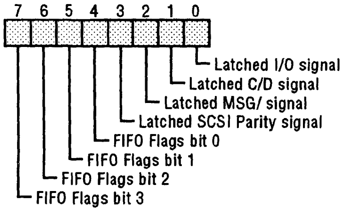
SSTAT0 Register R (0Dh)



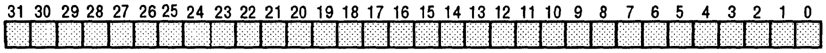
SSTAT1 Register R (0Eh)



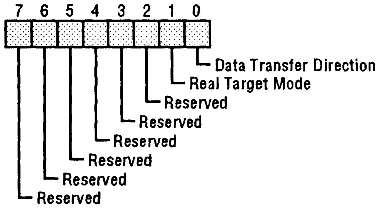
SSTAT2 Register R (0Fh)



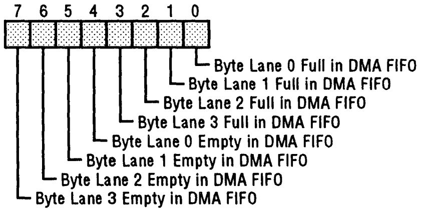
SCRATCHA Register R/W (10-13h) – 53C700-66



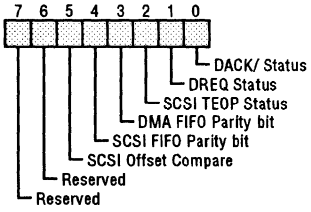
CTEST0 Register R (14h)



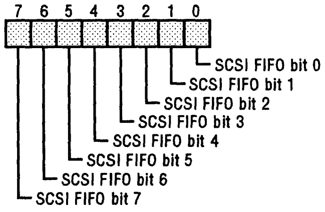
CTEST1 Register R (15h)


7 6 5 4 3 2 1 0
Byte Lane 0 Full in DMA FIFO
Byte Lane 1 Full in DMA FIFO
Byte Lane 2 Full in DMA FIFO
Byte Lane 3 Full in DMA FIFO
Byte Lane 0 Empty in DMA FIFO
Byte Lane 1 Empty in DMA FIFO
Byte Lane 2 Empty in DMA FIFO
Byte Lane 3 Empty in DMA FIFO

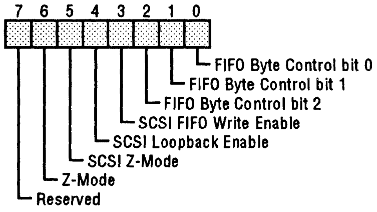
CTEST2 Register R (16h)



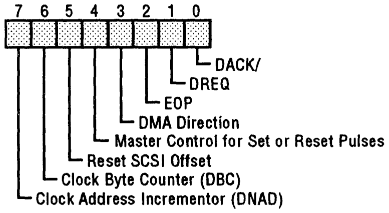
CTEST3 Register R (17h)



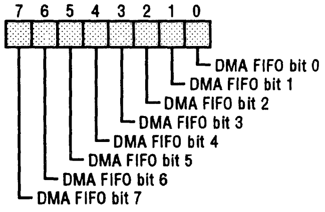
CTEST4 Register R/W (18h)



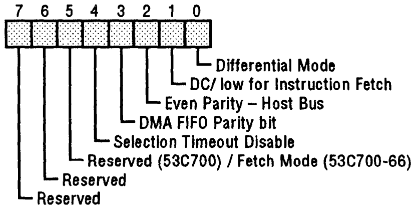
CTEST5 Register R/W (19h)



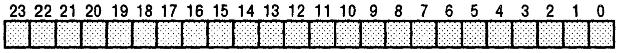
CTEST6 Register R/W (1Ah)



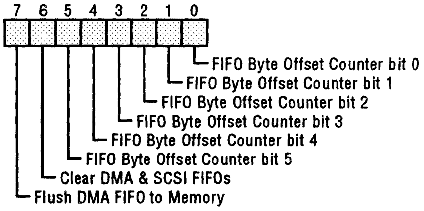
CTEST7 Register R/W (1Bh)


7 6 5 4 3 2 1 0
Differential Mode
DC/ low for Instruction Fetch
Even Parity – Host Bus
DMA FIFO Parity bit
Selection Timeout Disable
Reserved (53C700) / Fetch Mode (53C700-66)
Reserved
Reserved

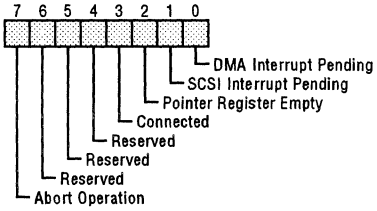
TEMP Register R/W (1C-1Fh)



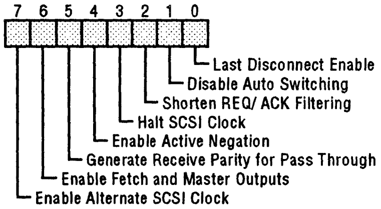
DFIFO Register R/W (20h)


7 6 5 4 3 2 1 0
FIFO Byte Offset Counter bit 0
FIFO Byte Offset Counter bit 1
FIFO Byte Offset Counter bit 2
FIFO Byte Offset Counter bit 3
FIFO Byte Offset Counter bit 4
FIFO Byte Offset Counter bit 5
Clear DMA & SCSI FIFOs
Flush DMA FIFO to Memory

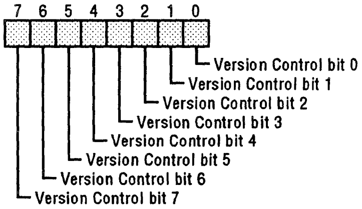
ISTAT Register R/W (21h)



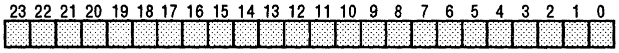
CTEST8 Register R/W (22h) – 53C700-66 only


7 6 5 4 3 2 1 0
Last Disconnect Enable
Disable Auto Switching
Shorten REQ/ ACK Filtering
Halt SCSI Clock
Enable Active Negation
Generate Receive Parity for Pass Through
Enable Fetch and Master Outputs
Enable Alternate SCSI Clock

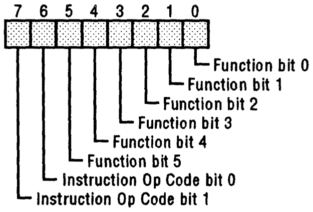
CTEST9 Register R/W (23h) – 53C700-66 only



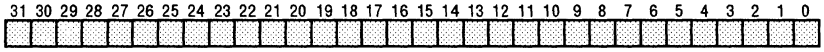
DBC Register R/W (24-26h)



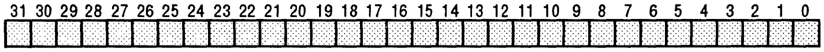
DCMD Register R/W (27h)



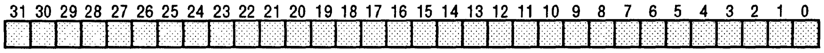
DNAD Register R/W (28-2Bh)



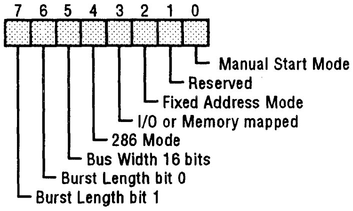
DSP Register R/W (2C-2Fh)



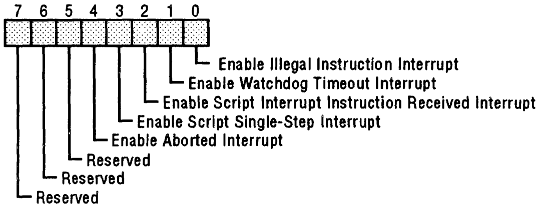
DSPS Register R/W (30-33h)



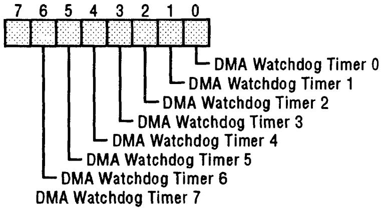
DMODE Register R/W (34h)



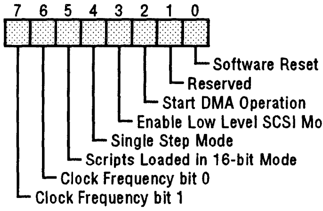
DIEN Register R/W (39h)


7 6 5 4 3 2 1 0
Enable Illegal Instruction Interrupt
Enable Watchdog Timeout Interrupt
Enable Script Interrupt Instruction Received Interrupt
Enable Script Single-Step Interrupt
Enable Aborted Interrupt
Reserved
Reserved
Reserved

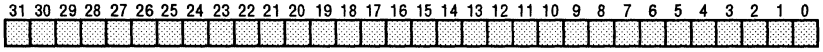
DWT Register R/W (3Ah)



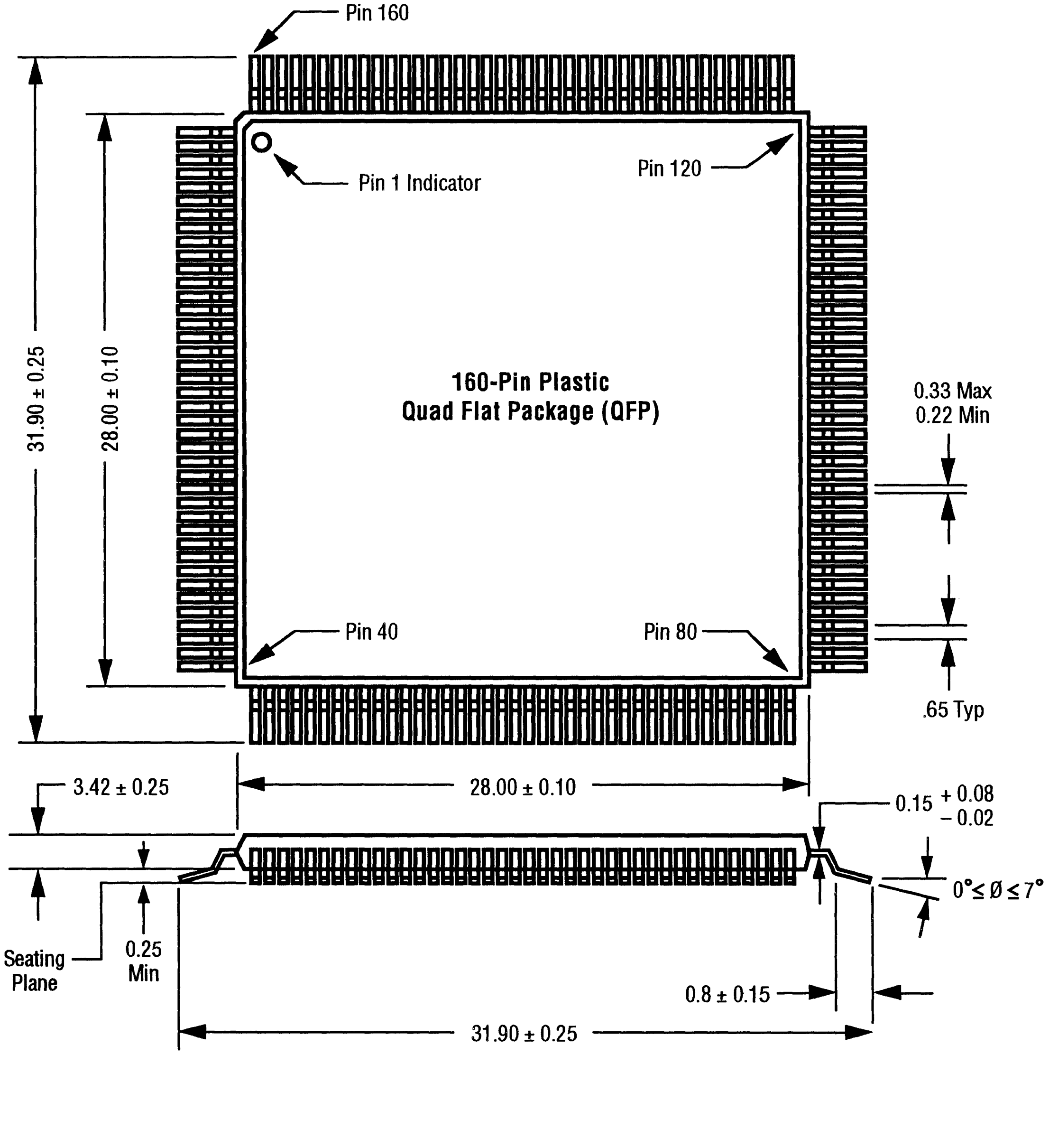
DCNTL Register R/W (3Bh)



SCRATCHB Register R/W (3C-3Fh) – 53C700-66 only



Appendix B
Mechanical Drawing



Note: All dimensions are in millimeters. This package is known as a Metric QFP, and should not be confused with the similar JEDEC QFP. Unless otherwise specified, all dimensions are nominal.

Appendix C
SCSI Engineering Notes

Introduction

This section of the manual contains SCSI Engineering Notes (SENs) which describe 53C700/ 53C700-66 hardware implementation issues.

NCR

SCSI
Engineering
Notes

No. 824
Interfacing the NCR 53C700/53C700-66 to the Intel 80386SX
Rev. 1.1, 1/93

This Engineering Note explains how the NCR 53C700/53C700-66 can interface to the Intel 80386SX. The Intel 80386SX has a 16-bit interface, and the NCR 53C700/53C700-66 has a 32-bit (386 mode) or 16-bit (286 mode) interface. Therefore, to directly connect the 80386SX to the NCR 53C700/53C700-66, the chip must be used in 286 mode.

286 Mode

The 53C700/53C700-66 is initialized for 286 mode in the DMODE register, bit 4. The 286 mode must be initialized before accessing any other registers in the chip.

The hardware implementation is a direct connect to the 80386SX with the exception of signal A0 on the 53C700/53C700-66. For example, data transfers are performed as described in the following tables.

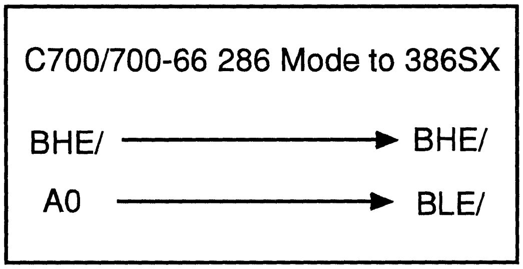
Table 1. Intel 80386SX

BHE/	BLE/	
0	0	word
0	1	byte XFER on D15-D8
1	0	byte XFER on D7-D0
1	1	never occurs

Table 2. NCR 53C700/53C700-66 286 Mode

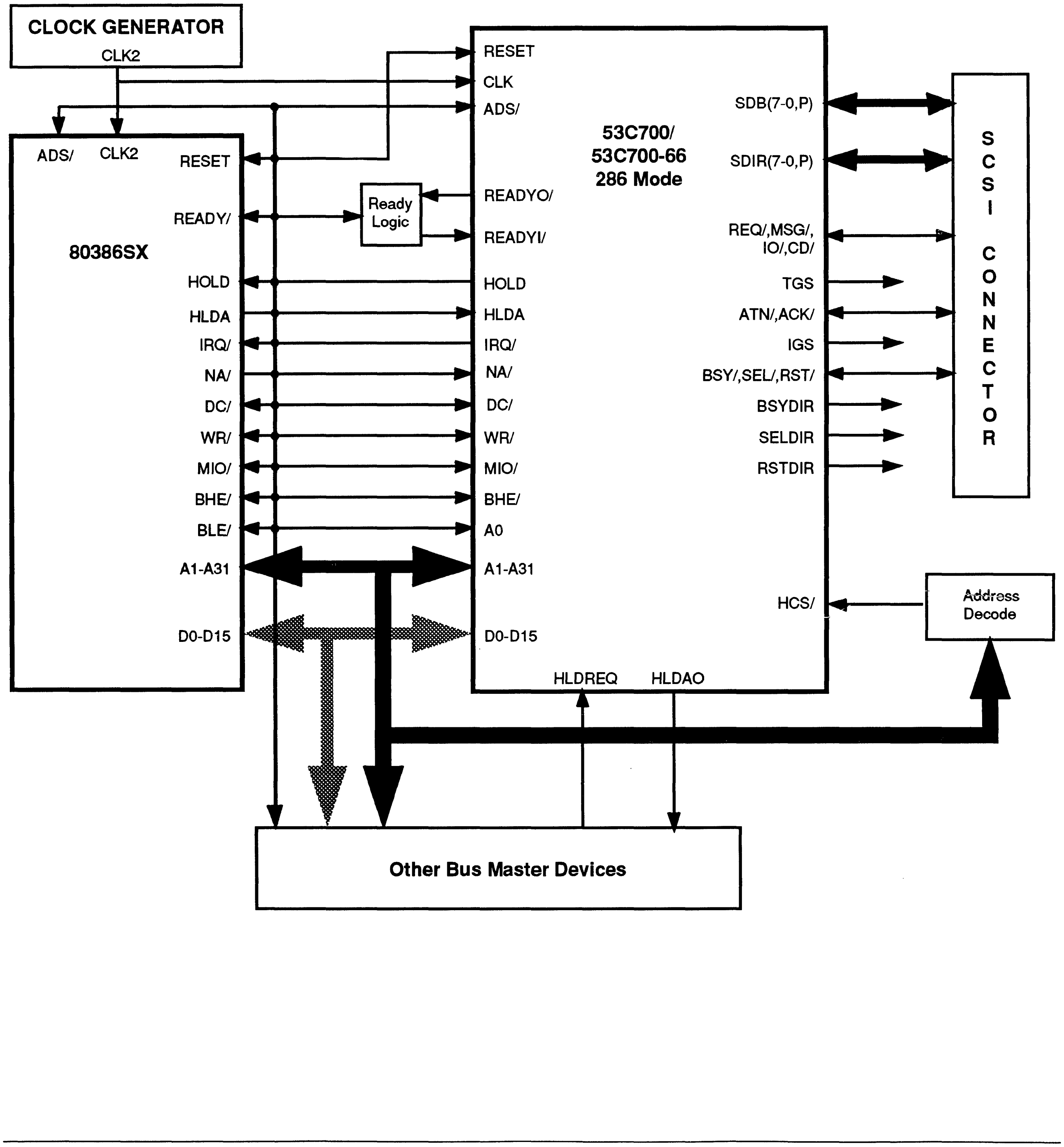
BHE/	A0	
0	0	word XFER
0	1	byte XFER on D15-D8
1	0	byte XFER on D7-D0
1	1	never occurs

Therefore, the 53C700/53C700-66 BHE/ connects to the 80386SX BHE/ and the 53C700/ 53C700-66 A0 connects to the 80386SX BLE/.



This connection is shown on the following page in Figure 1.

Figure 1. SCSI I/O Processor Block Diagram



NCR

SCSI
Engineering
Notes

No. 828
53C700/53C700-66 to 80286 Interface
Rev. 1.1, 1/93

The following is a suggested interface between the NCR 53C700/53C700-66 in 286 mode, and the Intel 80286. The purpose of this note is to guide in the design of an interface, not to provide a complete solution. Please note that the interface given in this diagram has not been tested, and the designer should perform a complete test procedure before implementing this interface.

286 Mode

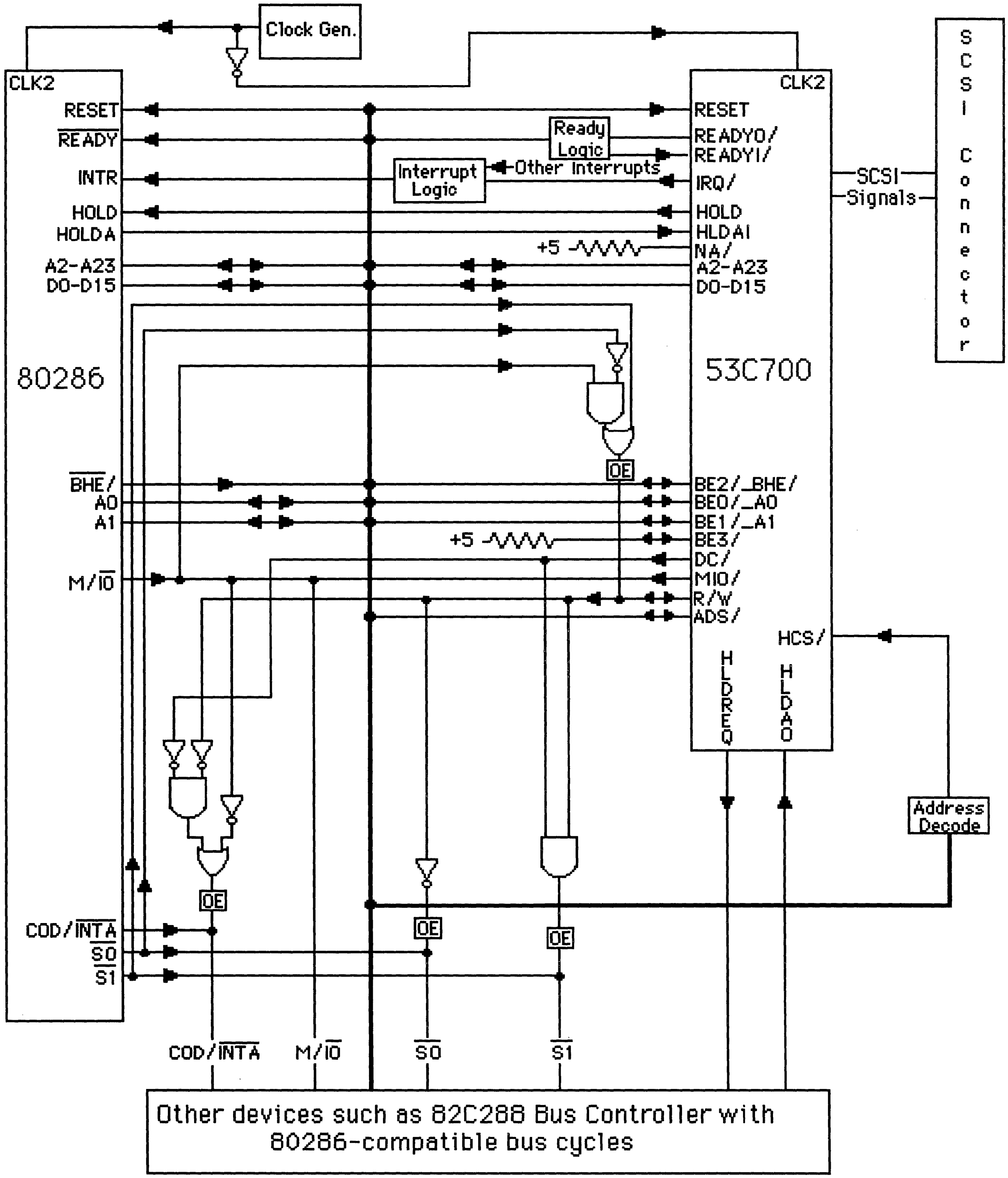
The 53C700/53C700-66 must be set for 286 mode by setting bit 4 of the DMODE register before accessing any other registers.

Interface

The 286 uses four signals (S0/, S1/, M_IO/, and COD_INTA/) to define the seven different bus conditions, whereas the 53C700/53C700-66 uses three signals (DC/, MIO/, and R/W) to define the same seven conditions. The 53C700/53C700-66 signals must be mapped to the equivalent 286 signals so it can interact with devices expecting 286 compatible bus signals. A K-map or other logic reduction tools can be used to do this mapping. The interface is given in Figure 1 on the following page.

The Interrupt Logic box is an interrupt controller such as the Intel 8259 Interrupt Controller.

When the 286 is bus master, its four signals are connected directly to the external devices, bypassing the 53C700/53C700-66 to 286 conversion logic. In this case, if the 286 accesses the 53C700/53C700-66 internal registers, it needs to produce a 53C700/53C700-66 compatible R/W signal. The logic connected to the R/W signal produces this signal. When the 53C700/53C700-66 is bus master, its three signals are transformed to the four 286 compatible signals which drive the external devices. Care must be taken that both the devices are not driving the bus at the same time. Unused data and address lines should be tied high.



OE = Output enable logic such as a Bus Transceiver.

Figure 1. 80286 to 53C700 Interface

SCSI
Engineering
Notes

No. 830
53C700/53C700-66 to 80286 Interface
Rev.1.1 , 1/93

Following is a suggested interface between the NCR 53C700/53C700-66 and the Motorola 68030. The purpose of this application note is to guide people in their own designs. The main functions of the interface have been tested on a Macintosh SE board. However, the interface tests have by no means been exhaustive so unanticipated problems may exist.

Interface

Two components in the circuit warrant an explanation.

- MC88915
- 74LS148

The MC88915 is a low skew CMOS PPL clock driver and it is used to multiply the input clock frequency (68030 clk) by two. The 2 x clock output is used to drive the NCR 53C700/53C700-66, the asynchronous bus control Pal and the arbitration Pal.

The 74LS148 is a 8-line to 3-line priority encoder and it is used to enable up to 7 more interrupts in addition to the one already coming from the 53C700/53C700-66.

The interface employs several Pals and their logical function is described below.

Address Decode Logic: A Pal 16L8 is used to decode the 53C700/53C700-66 addresses and select the chip by asserting the HCS/ signal. The address selecting the chip is 5800 XXXX Hex. This address is defined as the 68030 Direct Slot expansion on the Macintosh SE board.

Byte Enable Logic: A Pal 16V8 is used to convert between the Motorola size signals (SIZØ, SIZ1) and address signals (AØ, A1) to the NCR 53C700/53C700-66 byte enable signals (BEØ/, BE1/, BE2/, BE3/) when the 53C700/53C700-66 is in slave mode. When the 53C700/53C700-66 is in master mode the reverse conversion is done. The 53C700/53C700-66 HLDAI signal is used to select the conversion direction. It should be noted that the Pal also takes care of converting from big endian (Motorola) to little endian (Intel) mode and visa versa.

Asynchronous Bus Control Logic: A Pal 22V1Ø is used to generate the Motorola bus control signals (AS/, DS/, DBEN/, ECS/, RW/) from the NCR 53C700/53C700-66 signals (ADS/, HOLD, WR/) when the 53C700/53C700-66 is bus master. It should be noted that the interface only accommodates asynchronous transfers using the Motorola signals DSACK0/ and DSACK1/.

Arbitration Logic: A Pal 16R4 is used to transform the Motorola arbitration control signals (BR/ , BG/, BGACK/) to the NCR 53C700/53C700-66 signals (HOLD, HLDAI) and vice \versa depending on which chip is bus master.

In general, it is recommended to tie tri-state outputs high through 10 KΩ pull-up resistors if there is a possibility that these signals may temporarily leave an input in the high impedance state. Unused input signals must be held at a valid logic level. In this circuit, all unused inputs are tied high to minimize current draw in bi-polar devices.

The reserved pins on the NCR 53C700/53C700-66 should be tied high through a 10 KΩ resistor. Pins 93 through 102 should be pulled up independently of pins 89 through 92. If the reserved pins are not tied high the NCR 53C700/53C700-66 registers may read back incorrect values. Typically the chip will read back FF's during slave mode register reads. (Refer to the 53C700/53C700-66 data manual). In addition, use 0.1 µF capacitors between the following power and ground pins: 2 (Vss) and 6 (Vdd), 15 (Vss) and 19 (Vdd), 31 (Vss) and 39 (Vdd), 54 (Vss) and 49 (Vdd), 61 (Vss) and 64 (Vdd), 75 (Vss) and 79 (Vdd), 96 (Vss) and 99 (Vdd), 105 (Vss) and 109 (Vdd), 136 (Vss) and 139 (Vdd), 156 (Vss) and 153 (Vdd). The above decoupling is recommended for optimum noise isolation. This configuration takes advantage of the separate ground planes within the internal logic and the output drivers in the 53C700/ 53C700-66.

SCSI

The SCSI signals are connected to a SCSI connector and terminated through 220/330 Ω resistor packs. The reset signal on the SCSI bus is connected to a switch so that the bus can be reset manually. There are 4 LED's connected to the BSY/, IRQ/, HCS/ and HLDAI. These signals will tell you if the SCSI bus is driven, if the 53C700/53C700-66 has sent an interrupt, if the 53C700/53C700-66 has been selected, and if the 53C700/53C700-66 is a bus master, respectively.

Note: Schematics showing the implementation of this design are available for downloading from the NCR Microelectronic Products Division Electronic Bulletin Board (719) 596-1649. The filename for this data is SEN830.Zip. Under this filename will be found two files: the design, in a VIEWlogic database, and a set of PAL equations used in the design.

SCSI
Engineering
Notes

No. 835
53C700/53C700-66 Implementation Issues
Rev.1.1 , 1/93

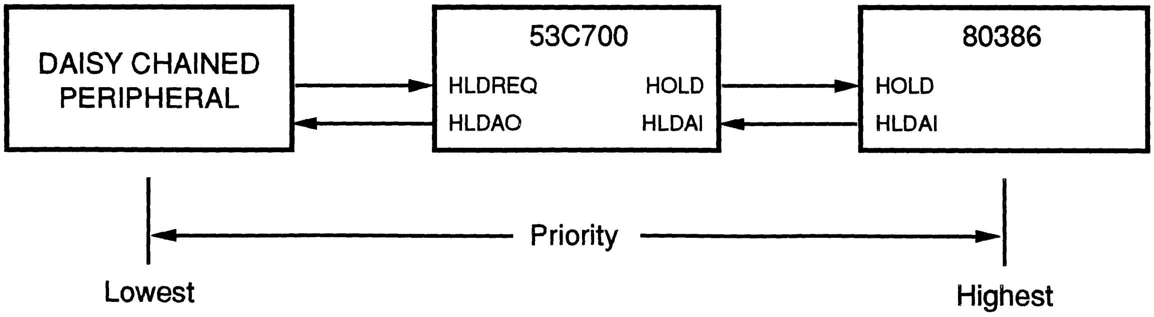
This Engineering Note applies to the NCR 53C700 and NCR 53C700-66 chips. For simplicity, the text refers to both chips as 53C700.

The 53C700 is both a bus master and a bus slave device. This causes some confusion with the READY signals, and the HOLD / HOLD-ACKNOWLEDGE signals. This applications note shows how to use the two groups of signals as well as the correct way to synchronize to the 53C700 clock. While the 53C700 is shown in its native 80386 environment, these concepts are most helpful to designers who are not using the 80386.

Bus Master and Bus Slave

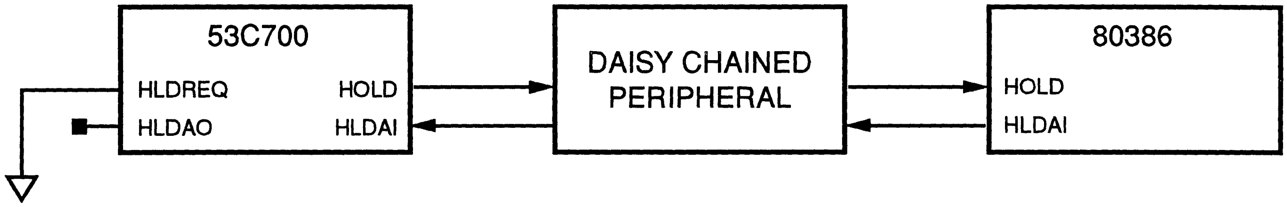
A bus master is simply a device that controls the system bus. A microprocessor is an example of a bus master. It drives the address and control signals. A bus slave is a device that responds to a bus master. RAM memory is an example of a bus slave. Some devices, such as DMA controllers and the 53C700 are both masters and slaves. The 53C700 is a bus slave when some other device accesses its registers. It becomes a bus master by asking for and receiving permission from the system microprocessor to own the bus. To ask permission to become a bus master in a 80386 system, a device asserts the HOLD signal. The microprocessor will finish its current instruction then grant the request by asserting the hold acknowledge signal, or HLDA. After HLDA is driven true, the new bus master may drive the address and bus control signals.

Figure 1. 53C700 has high priority in Daisy Chain



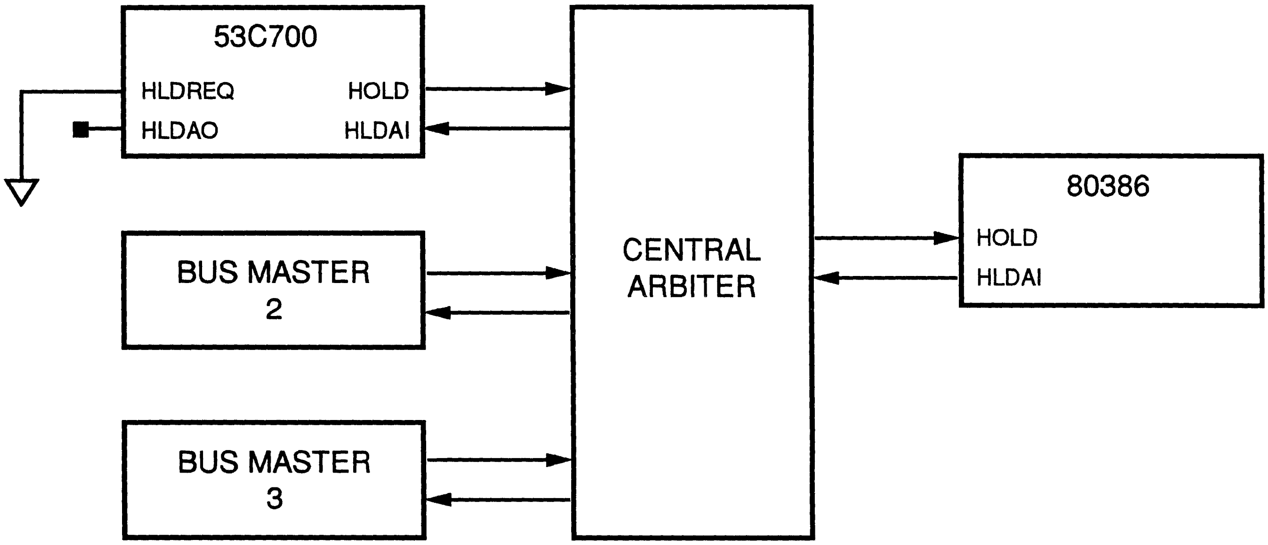
When there are more than two bus masters, some method of arbitration must be employed to allow competing devices to decide who wins. There are two arbitration methods: Daisy Chain, and Central Arbiter. The previous illustration shows how to connect the 53C700 as the highest priority peripheral, second only to the 80386. The next illustration, shows how to connect the 53C700 as the lowest priority device. It is important to ground the HLDREQ input.

Figure 2. 53C700 has lowest priority in Daisy Chain



In real-time systems, or in systems with many arbitrating devices, the central arbitration method gives better results. The priority of each device is set inside the arbiter. System designers may either design their own arbiter, or choose one from the microprocessor family in use. The 80386 family includes the 82380, which may be adapted to become a central arbiter by using the Cascade Mode.

Figure 3. Central Arbitration



Wait States

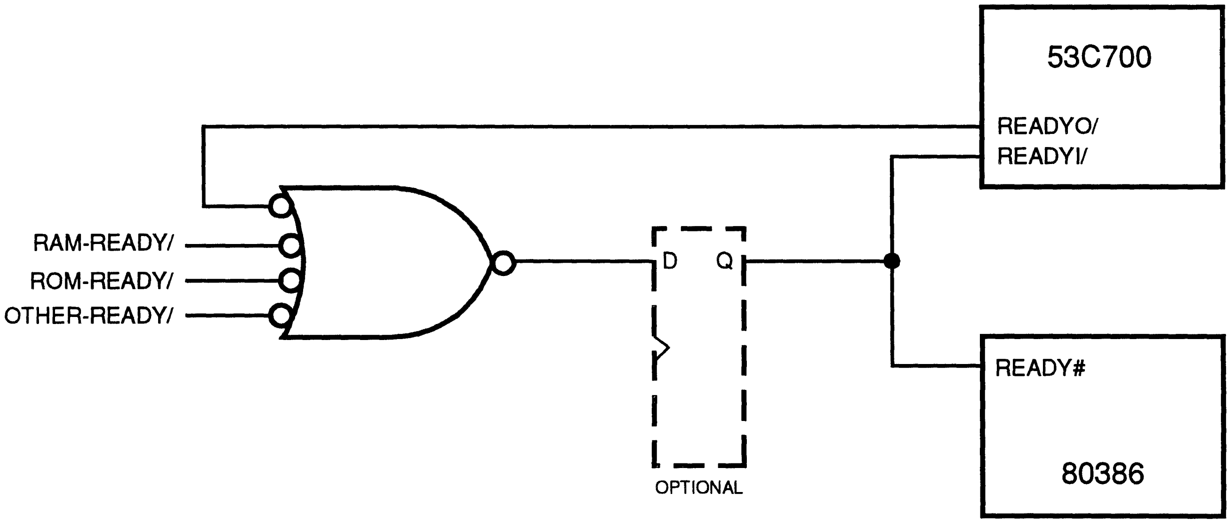
.'he 80386 is a very fast bus master that can 'talk' to slower devices through a mechanism popularly known as wait states. The 386 is a state machine with a basic bus cycle of two (internal) clocks, or states. The first state is T1 and the second is T2. A fast slave will assert READY so that the entire cycle consists of T1 state followed by T2 state. If the slave needs more time to receive or send data, it drives the READY signal to not-ready. The bus master then idles in T2. The bus master outputs do not change in T2, rather they change on the transition to T1 for the next cycle, after the slave drives ready true. For historical reasons, this idling in T2 is called "adding wait states". A fast slave adds no wait states; it is said to be a zero wait state device.

The 53C700 requires wait states when operating as a slave, but not as a master. In all cases, the timing diagrams shown in the 53C700 data manual show the exact number of wait states required. These are not optional, but are required by the device. The case that causes the most confusion is Slave Mode Write: it will latch the incoming data on the same rising clock edge (phase 1 of T2) that it asserts READYO on. In the timing diagram, this is the clock edge that t10 is refrained to. Manipulating the READYI signal will not change when the 53C700 latches the incoming data.

READY

The 53C700 has two ready signals, READYI (input) and READYO (output). When the ;3C700 becomes a bus master, it samples READYI from the slave device in order to know when to end the cycle. The 53C700 will idle in T2, adding wait states indefinitely, until READYI is driven true.

Figure 4. 53C700 must see the same READY as the 80386



When the 53C700 operates as a bus slave it must return a ready to the processor to end the cycle. READYO is provided for this purpose. The 53C700 knows how many wait states it needs, and drives READYO to signal that it is able to end the cycle. These wait states are fixed inside the 53C700; they are not programmable. The 53C700 must also know exactly which clock edge the 80386 sees the ready signal go true. So in slave mode, the 53C700 drives READYO and samples READYI. This allows the system designer the option of synchronizing the ready signal. The 53C700 synchronizes READYO to the 80386 clock; it does not require external synchronization. However other devices in the system may require synchronization. In all cases, the 53C700 must 'see' the same ready signal that the 80386 sees.

Synchronizing to the Clock

Both the 80386 and the 53C700 divide the input clock by two and use their internal, half-speed clock to sequence the internal state machines. External devices must be correctly synchronized to this internal clock. Unfortunately, this half-speed clock is not available outside the chip. The only way to synchronize to the external clock is with the RESET signal.

Figure 5. Internal Clocks

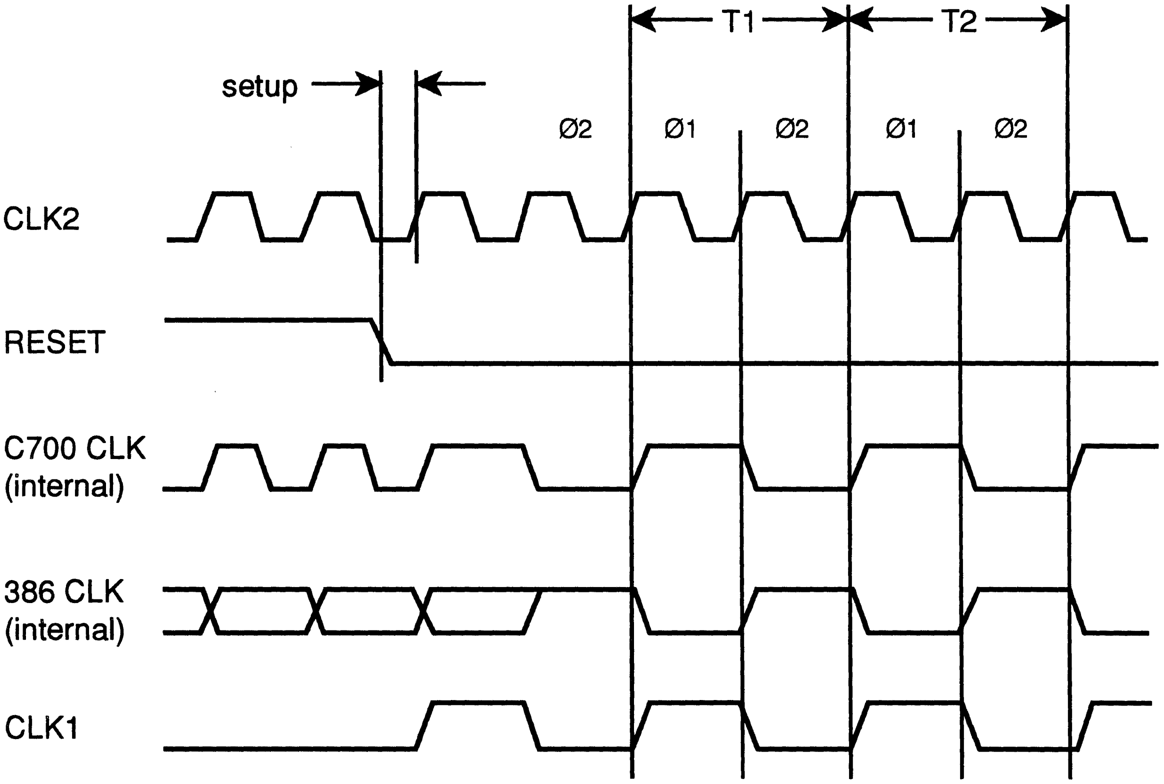
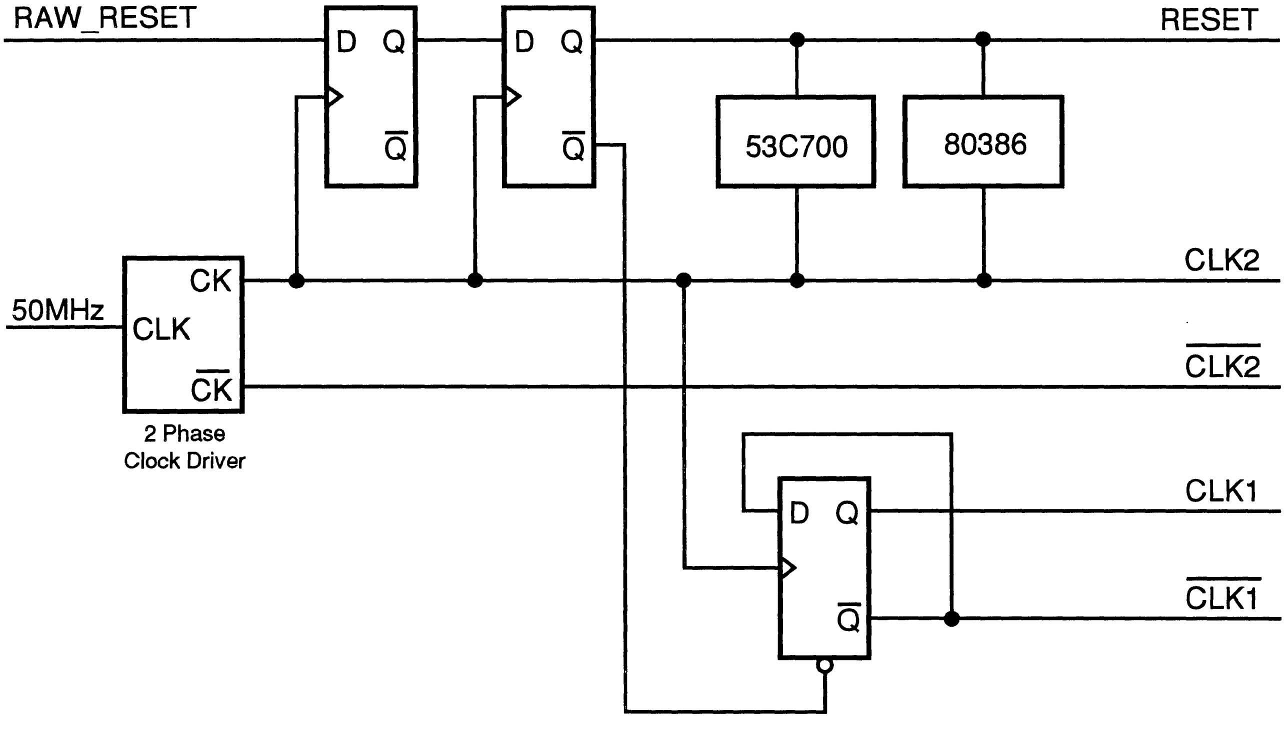


Figure 5 shows how both the 53C700 and 80386 come out of the reset state. The the high speed input clock is labeled CLK2. The active high RESET pulse width is required to be five CLK2's minimum for the 53C700 and fifteen CLK2's minimum for the 80386. Notice that both chips generate internal clocks with phase one of T1 starting on the third rising edge of CLK2 after RESET goes false. It is unimportant that the two internal clocks are complemented with respect to each other. What is important is that each circuit can always distinguish between phase one and phase two of the clock.

These clocks are not available outside either chip. But certain inputs to both devices must be synchronized to the phase of these internal clocks. If these input signals come from 80386 family devices, they are probably synchronized correctly. But if they come from non-386 devices then it will be necessary to re-generate this clock externally in order to synchronize to the phase of the internal clocks. CLK 1 is an externally generated clock to help synchronize inputs to the 53C700 internal clock.

Figure 6 shows one method of generating this external CLK1. When CLK1 is high, the 53C700 is in phase one; when CLK1 is low, the 53C700 is in phase two. The circuit shown in figure 6 creates both the true and complementary version of each clock. When working with high speed clocks, it is good practice to buffer the clock once, then run everything off the buffered clock.

Figure 6. Sample CLK1 generator



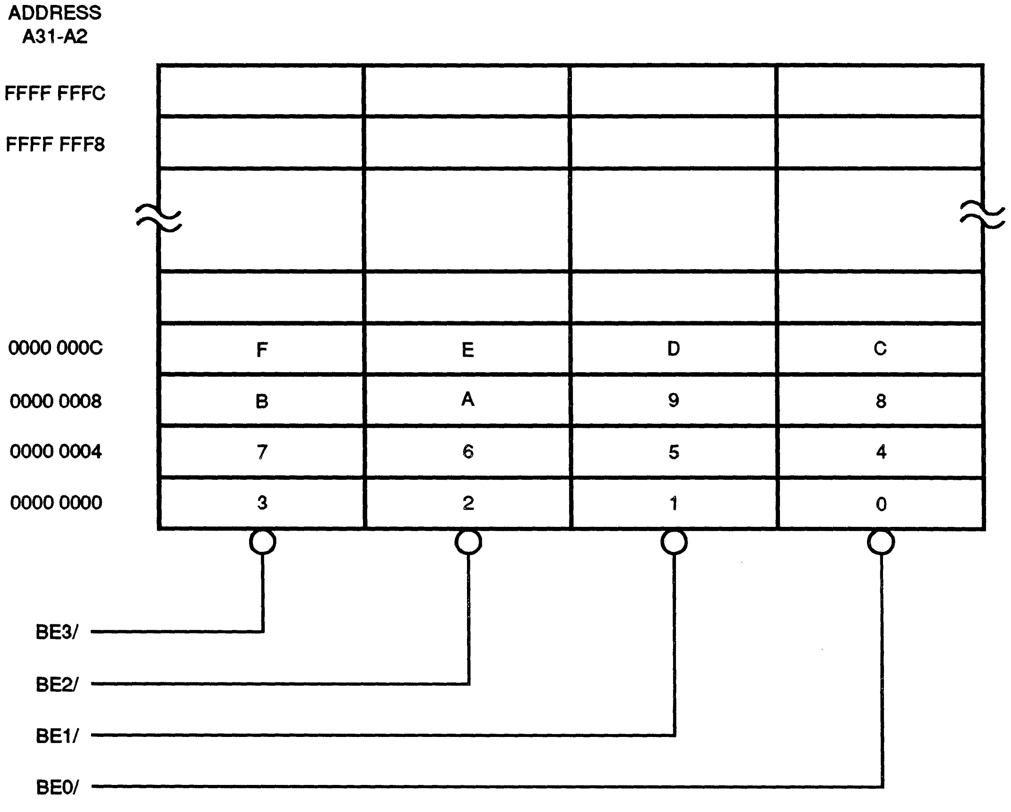
Since several devices are connected to RESET in figure 6, it is good practice to synchronize this signal to the clock. Each of the devices requires that RESET stabilize before the clock rising edge. If RESET is not synchronized, then there is a narrow timing window where some of the devices will trigger on RESET, but others will not. The two flip-flop's in series will eliminate metastability if (and only if) the recovery-time of the first flip-flop is smaller than the clock period. Metastability is a complex subject and will not be covered here, except to say that there is a tradeoff between recovery-time and what is called MTBF (Mean Time Between Failure). MTBF is somewhat misleading. In metastability studies, "failure" means that the metastable device took longer than the specified recovery-time to recover--not that the device went metastable. So, if we continuously operate the flip-flop under conditions that cause metastability (violate the setup and hold times), then once every MTBF, on the average, the device will take longer than the specified recovery-time to recover. If the device fails this way on two successive clock edges, then the double synchronizer fails. In general, CMOS devices recover from the metastable state faster than bi-polar devices.

32-bit Memory Structure

Figure 7 shows how 32-bit memory is organized in a 80386 system. The bus master generates long word address only. It also generates four byte enable signals (BE3/ - BE0/) to access any of the four bytes within that long word address. For example, to access the word starting at address 0000 000D (hex), the system would place 0000 000C on the address bus and drive BE1/ and BE2/ true.

Each column, enabled by a byte enable signal, is called a byte lane. Whether the 53C700 is operating as a bus master or slave, it will ignore any data on byte lanes whose corresponding byte enable signal is false.

Figure 7. 32-bit Memory Organization



ADDRESS A31-A2				
FFFF FFFC				
FFFF FFF8				
0000 000C	F	E	D	C
0000 0008	B	A	9	8
0000 0004	7	6	5	4
0000 0000	3	2	1	0

The 53C700 recognizes two different data types: script instruction and SCSI data. Script instructions must begin on long word boundaries. SCSI Data need not be aligned in any particular way. The 53C700 can access data either: 4 bytes at a time; 3 bytes at a time; 2 bytes at a time; or 1 byte at a time. The 53C700 will correctly move any misaligned bytes at the beginning, or any odd bytes at the end of a transfer. If the data is misaligned at the beginning of a transfer, the 53C700 will align itself to a long word boundary by transferring the first 3, 2, or 1 bytes. The 53C700 will then perform 32-bit transfers until the end of a transfer. On the very last transfer, the 53C700 will move either 4, 3, 2 or 1 byte without processor intervention.

SCSI
Engineering
Notes

No. 850
Setting the NCR 53C700-66 Transfer Rate
1.0. , 1/93

This note explains how to get the proper data transfer speed on the SCSI bus when using the NCR 53C700-66 SCSI I/O Processor chip.

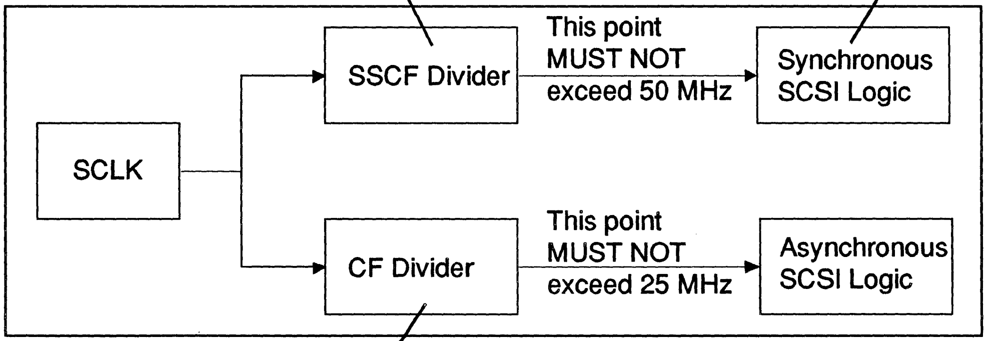
Data transfer rates are controlled by the configuraton of the following bits.

- *Register 0B SCSI Bus Control Line (SBCL) – bits 1 (SSCF1) and 0 (SSCF0)*
  SSCF1-0 bits selects the factor by which the frequency of SCLK is divided before being presented to the synchronous SCSI control logic. The output from this divider must not exceed 50 MHz.
- *Register 3B DMA Control (DCNTL) – bits 7 (CF1) and 6 (CF0)*
  CF1-0 bits selects the factor by which the frequency of SCLK is divided before being presented to the asynchronous SCSI core logic. This divider must be set according to the input clock frequency in the table. The output from this divider must not exceed 25 MHz.
- *Register 05 SCSI Transfer (SXFER) – bits 6 (TP2), 5 (TP1) and 4 (TP0)*
  TP2-0 determines the SCSI synchronous transfer period when sending synchronous SCSI data in either initiator or target mode.

These bits control the programmable dividers in the chip. On the following page are tables of these bits and the division factors they produce. The illustration shows how the chip uses the divisors.

SSCF1	SSCF0	SSCF Divisor
0	0	Set by DCNTL
0	1	1.0
1	0	1.5
1	1	2.0

TP2	TP1	TP0	XFERP
0	0	0	4
0	0	1	5
0	1	0	6
0	1	1	7
1	0	0	8
1	0	1	9
1	1	0	10
1	1	1	11



CF1	CF0	CF Divisor	SCLK (SCSI Clock) Frequency given in Mega Hertz
0	0	2.0	37.51 - 50.00 MHz
0	1	1.5	25.01 - 37.50 MHz
1	0	1.0	16.67 - 25.00 MHz
1	1	3.0	50.01 - 66.67 MHz

Example:
SCLK = 40 MHz, SCSI Transfer Rate = 10 MB/s
SSCF = 1, CF = 0, XFERP = 4 (TP = 0)
(40 MHz ÷ 1 = 40 MHz sync. logic speed)
(40 MHz ÷ 2 = 20 MHz async. logic speed)
SCSI synch. logic speed ÷ XFERP = SCSI synch. rate
(40 MHz ÷ 4 = 10 MHz = 10 MB/sec)

Appendix D
53C700 vs 53C700-66 Differences

This appendix presents the differences between the NCR 53C700 and the NCR 53C700-66.

Features	NCR 53C700	NCR 53C700-66
Pin 93	Reserved	MASTER/ – Master Status This feature is enabled by bit 6 of CTEST8 (For more information see page 3-4)
Pin 94	Reserved	FETCH/ – Fetch Opcode This feature is enabled by bit 6 of CTEST8 (For more information see page 3-4)
Pin 95	Reserved	SCLK – SCSI Clock This feature is enabled by bit 7 of CTEST8 (For more information see page 3-4)
CTEST8 Register	Reserved	Chip Test 8 Register (R/W) Hex 23 (For more information see page 4-27)
CTEST9 Register	Reserved	Chip Test 9 Register (R/W) Hex 22 (For more information see page 4-28)
SCRATCHA	Reserved	Scratch A Register (Read/Write) Hex 10-13 (For more information see page 4-18)
SCRATCHB	Reserved	Scratch B Registers (read/Write) Hex 3C-3F (For more information see page 4-35)
Clock Speed	Max. 50 MHz	Max. 66 MHz
TolerANT	Not available	This feature is enabled by setting the EAN bit (bit 4) of CTEST8 (For more information see page 1-1)

NCR

READER'S COMMENT FORM

F-8763 0687

BOOK TITLE	BOOK NO.	PRINT DATE

To help us plan future editions of this document, please take a few minutes to answer the following questions. Explain in detail using the space provided. Include page numbers where applicable.

Are there any technical errors or misrepresentations in the document?

Is the material presented in a logical and consistent order?

Is it easy to locate specific information in the document?

Is there any information you would like to have added to the document?

Are the examples relevant to the task being described?

Could parts of the document be deleted without affecting the document's usefulness?

Did the document help you to perform your job?

Any general comments?

NAME ______________________

TITLE ______________________

COMPANY ______________________

ADDRESS ______________________

____

Thank you for your evaluation of this document.
Fold the form as indicated and mail to NCR. No postage is necessary in the U.S.A.

CUT ALONG THIS LI

fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

NCR Corporation
ATTENTION: Publication Services
WHQ-4
1700 S. Patterson Blvd.
Dayton, Ohio 45409

fold

cut

NCR Microelectronic Products Division Sales Locations

For literature on any NCR product or service,
call the NCR hotline toll-free
1-800-334-5454

Division Plant Locations

NCR Microelectronic Products Division
2001 Danfield Court
Fort Collins, CO 80525
(303) 226-9500

Disk and Tape ASIC Products
General ASIC Products
PC and Workstation ASIC Products
Customer Owned Tooling Products
Technology Products
Communications Products

NCR Microelectronic Products Division
1635 Aeroplaza Drive
Colorado Springs, CO 80916
(719) 596-5795
FAX (719) 527-8225

I/O Products
SCSI Software Products
Graphics Products

Multichip Module Technology Center

North American Sales Locations

Northwest Sales
1731 Technology Drive, Suite 600
San Jose, CA 95110
(408) 441-1080

Southwest Sales
3300 Irvine Avenue, Suite 255
Newport Beach, CA 92660
(714) 474-7095

North Central Sales
8000 Townline Avenue, Suite 209
Bloomington, MN 55438
(612) 941-7075

South Central Sales
17304 Preston Road, Suite 635
Dallas, TX 75252
(214) 733-3594

Northeast Sales
92 Montvale Ave., Suite 3500
Stoneham, MA 02180-3602
(617) 438-0381

Southeast Sales
1051 Cambridge Square, Suite C
Alpharetta, GA 30201
(404) 740-9151

International Sales Locations

European Sales Headquarters
Westendstrasse 193
80686 Muenchen

Postfach 210370
80673 Muenchen

Germany
49 89 57931199

Asia/Pacific Sales Headquarters
35th Floor, Shun Tak Centre
200 Connaught Road
Central
Hong Kong
852 859 6044

T43934I

1293-15MD

Printed in U.S.A.