NCR

Microelectronic Products Division
Colorado Springs

# *NCR 53C700*

## *SCSI I/O Processor*

*Data Manual Rev. 2.5*

## PRINTING HISTORY

While the information herein presented has been checked for both accuracy and reliability, NCR Corporation assumes no responsibliity for either its use or any inaccuracies.

*Prepared by NCR Microelectronics Products Division*

| Revision No. | Print Date | NCR internal Part No(s). covered by each manual |
|---|---|---|
| 2.3 Preliminary | 10/89 | 609-3400616 only |
| 2.4 Preliminary | 12/89 | 609-3400625/631 (up to and including) |
| 2.5 Preliminary | 2/90 | 609-3400634 (includes all previous revisions) |

## Additional Information

53C700 Programmer's Guide
SEN 822, 53C700 SCSI I/O Processor

## Trademarks

SCRIPTS is a registered trademark of NCR Corporation
IBM, Micro Channel are registered trademarks of International Business Machines Corporation

# TABLE OF CONTENTS

# List of Tables

# List of Figures

**Figure Description** **Page**

# Chapter 1
# Introduction

## NCR 53C700

- Supports up to 25 MHz 80386 Memory Bus Speeds
- Supports 32-bit Word Data Bursts with Variable Burst Lengths
- Unique Interrupt Status Reporting
- High-Speed Asynch/Sync SCSI Bus Transfers
- High Performance
- CMOS
- Supports Variable Block Size & Scatter/Gather Data Transfers
- Minimizes SCSI I/O Start Latency
- Performs Complex Bus Sequences without Interrupts
- Memory Transfers in excess of 50 MBytes/sec
- Single +5V supply

### I/O Performance

The demands on today's I/O interfaces are being pushed by the increasing performance of personal computers and workstations. High powered CPU's, both CISC and RISC, only provide marginal system performance if their I/O interfaces are not properly designed. Interrupt service routines which often take an excess of several hundred microseconds to execute can be a large source of performance delays. Interrupts may be generated for exception conditions, I/O completion, I/O status save/restore, or accommodation of the myriad of options currently available in I/O definitions. Interrupts can be reduced by using programmed I/O; however, this can be time consuming and does not provide a complete solution for multi-tasking operations.

### The Need For I/O Flexibility

Options in bus protocol allow increased flexibility. Need for I/O flexibility is partially responsible for the popularity of the SCSI standard. I//O flexibility allows configuration of systems for a wide range of peripherals from high performance disk drives to hand held scanners.

Additionally, I/O flexibility supports command queueing, asynchronous/synchronous data transfers, caching controllers, and peer level communication. However, flexibility can increase interrupts and severely impact system performance.

### A Better Solution

*First generation (NCR5380)* SCSI devices are register oriented and require processor intervention to make the most fundamental protocol decisions. Users like the flexibility of these devices because the low-level firmware interface provides specific information about the SCSI bus and improved testability of the SCSI device. This generation of SCSI devices typically requires in excess of 4,000 lines of code to specify a SCSI-1 initiator role implementation.

*Second generation (NCR53C90)* SCSI devices provide on-chip state machines allowing some complex bus sequences to be performed automatically which reduces protocol overhead. However, these devices have little decision making capability because the internal sequences are fixed in hardware or in some cases software. This generation of SCSI devices typically requires in excess of 2,500 lines of driver support software.

The flexibility of the SCSI bus has created a dilemma for system integrators and OEM's alike The dilemma is determining whether first and second generation SCSI devices should be used as non-intelligent, stand-alone devices or should they be integrated into intelligent host adaptor boards. Non-intelligent SCSI host ports or host bus adapters require a fair amount of processor intervention, but are inexpensive to implement. Intelligent host adapters are more expensive than non-intelligent adapters. They provide slower decision making capabilities (less powerful CPU's), experience interpretation delays (2-8 msec) required to start any I/O, and suffer from interprocessor communication delays. Consequently, non-intelligent host adapters consistently outperform their intelligent counterparts in many systems.

With MIPS increasing in the system CPU, the delays caused by intelligent host adapter cards increase in severity. The simplest solution is to build complex versatile sequences inside the SCSI components or to add additional CPU power in the intelligent host adapters. Both solutions are costly and do not adequately address the problem.

## Third Generation Requirements

To accommodate the flexibility requirements of the SCSI bus, reducing interrupts and controlling host adapter cost, an additional level of intelligence and integration is required for next generation devices. Third generation SCSI devices must make decisions based on phase changes on the SCSI bus, compare specific data values, and allow for interrupt minimization.

This requires a programmable SCSI device that executes SCSI oriented commands. These new devices must reduce interrupt service routine complexities by providing unique status reporting for any interrupts that occur. A fully integrated DMA channel would also allow full use of available host bus bandwidth.

This is the key to overall I/O performance, given current use of virtual memory schemes which require the ability to support scatter/gather memory operations without processor intervention. Third generation SCSI devices require only a few hundred lines of driver code. This code supports exception conditions and passes addresses of the user data buffer to the device.

# Chapter 2
# SCSI I/O Processor Description

The NCR 53C700 SIOP is the first intelligent SCSI host adapter on a chip. A high-performance re-usable SCSI core and an intelligent 32-bit bus master DMA have been integrated with a 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.

## SCSI Component

The SCSI core is designed to allow simple migratation to SCSI-2 wide bus and enhanced synchronous transfer rate requirements. It offers synchronous transfers up to 6.25 MBytes/sec with asynchronous transfers greater than 5 MBytes/sec. 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 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 integrated DMA core 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. The SCRIPTS PROCESSOR™ is a special 2MIPS processor on the SCSI chip.

## DMA Component

The DMA component is a bus master DMA device that attaches easily to the 80386, 80286, 80386SX, and 80376 processors. It supports 25 MHz 80386 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, 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 in excess of 40 MBytes/sec.

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.

## 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 nsec! This compares to 2 - 8 msec 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.

## 2.1 NCR 53C700 SIOP Benefit Summary

*Benefits* *Features*

### PERFORMANCE

- Supports up to 25 MHz 80386 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 nsec to Begin Compared to 2 - 8 msec
- Performs Complex Bus Sequences without Interrupts Including Restore Data Pointers
- Unique Interrupt Status Reporting - Reduces ISR Overhead
- High-Speed Async/Sync SCSI Bus Transfers
  5.0 MBytes/sec asynchronous
  6.25 MBytes/sec synchronous(with future migration to 10 MBytes/sec)
- Memory transfers in excess of 50 MBytes/sec

### INTEGRATION

- Full 32-Bit DMA Bus Master
- High Performance SCSI Core
- Integrated SCRIPTS PROCESSOR
- Allows Intelligent Host Adapter Performance on a Mother Board

### 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 or 80386 Support
- Externally Adaptable To EISA, MCA, and other System buses
- Supports Changes From Initiator to Target Roles Dynamically

## 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

## RELIABILITY

- 10K volts ESD protection SCSI Signals
- Greater than 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-Thru 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

## SCSI Specifications

This Data Manual is not a SCSI specification. It assumes some prior knowledge of the SCSI proposed standard. To obtain a copy of the proposed standard write to:

ANSI
1430 Broadway
New York, NY 10018

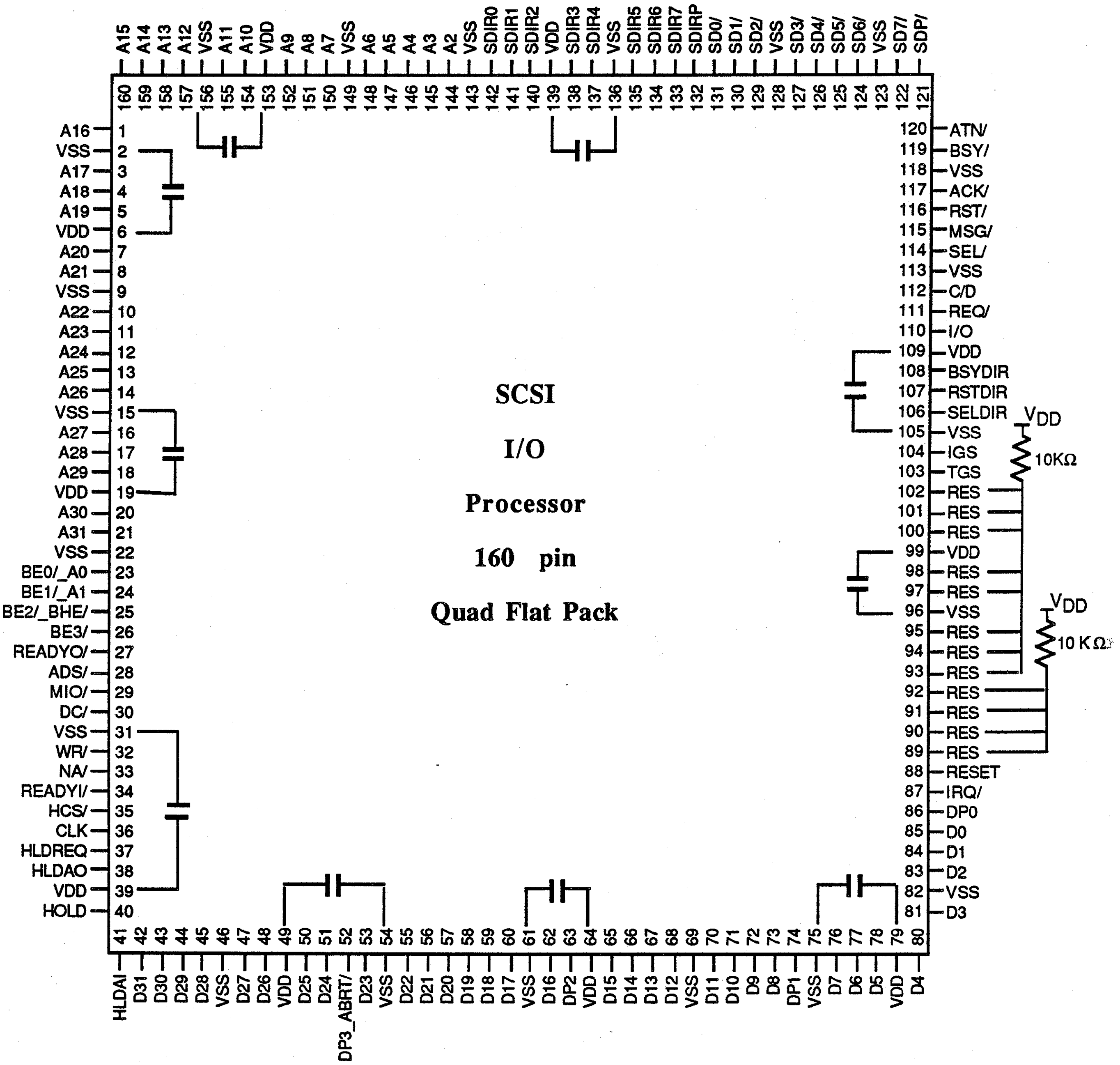
(212) 642-4900
Approximate cost: $25.00

Ask for document number:
X3.131 - 1986 (SCSI-1)

Global Engineering Documents
2805 McGaw
Irving, CA 92714

(800) 854-7179 or (714) 261-1455
Approximate cost: $50.00 ($60.00, non USA)

Ask for document number:
X3.131-198x(SCSI-2) Revision 6

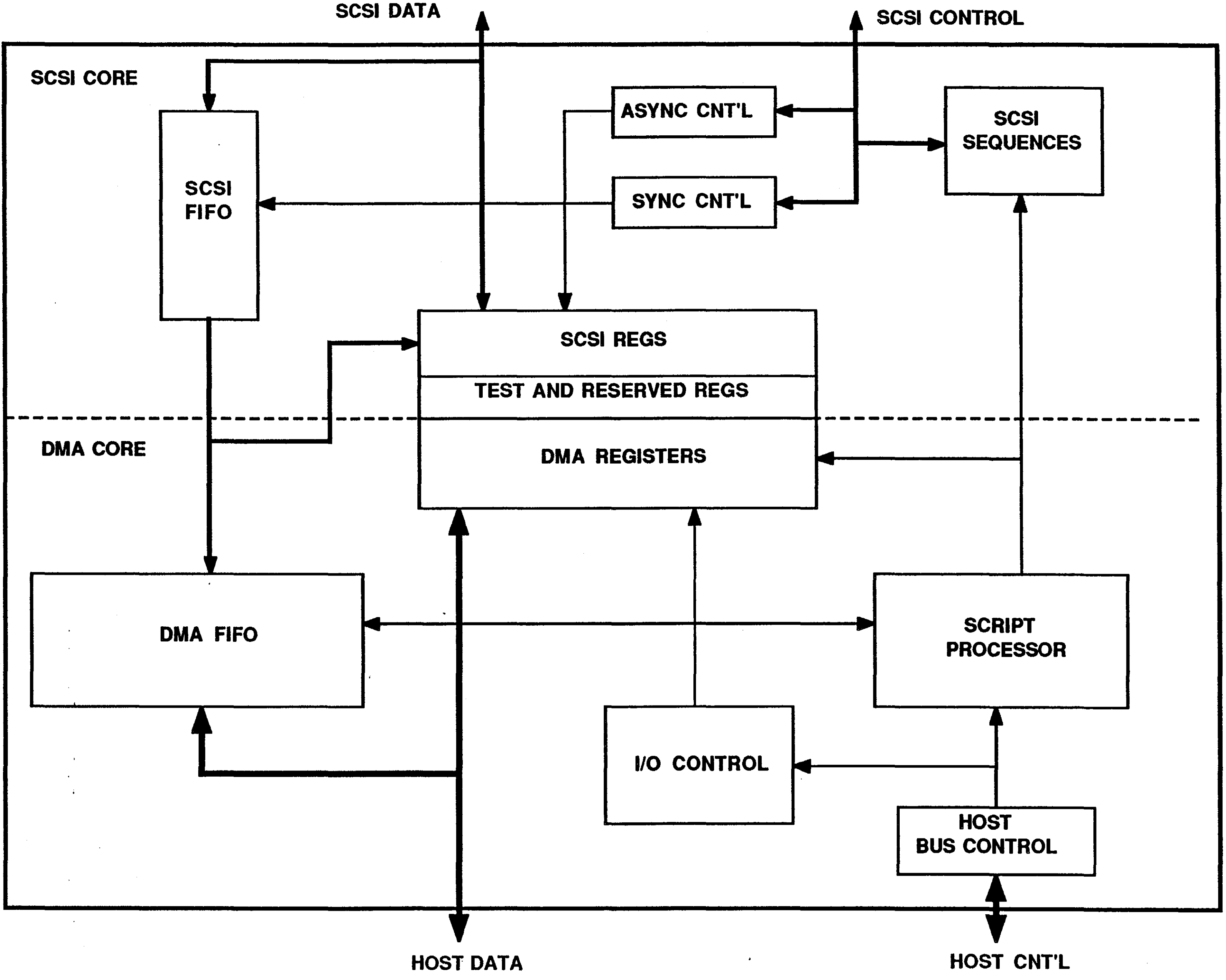


* NOTE: The above decoupling is recommended for optimum noise isolation. This configuration takes advantage of the separate ground planes contained in the SIOP for address and data lines, control lines, and SCSI pads. Use capacitor values of .01 uF or .1 uF.

**NOTE: Pins 102-100, 98, 97, and 95-89 are reserved for wide SCSI.

*Figure 1. NCR 53C700 Pinout*

Figure 2. 53C700 SCSI I/O PROCESSOR BLOCK DIAGRAM



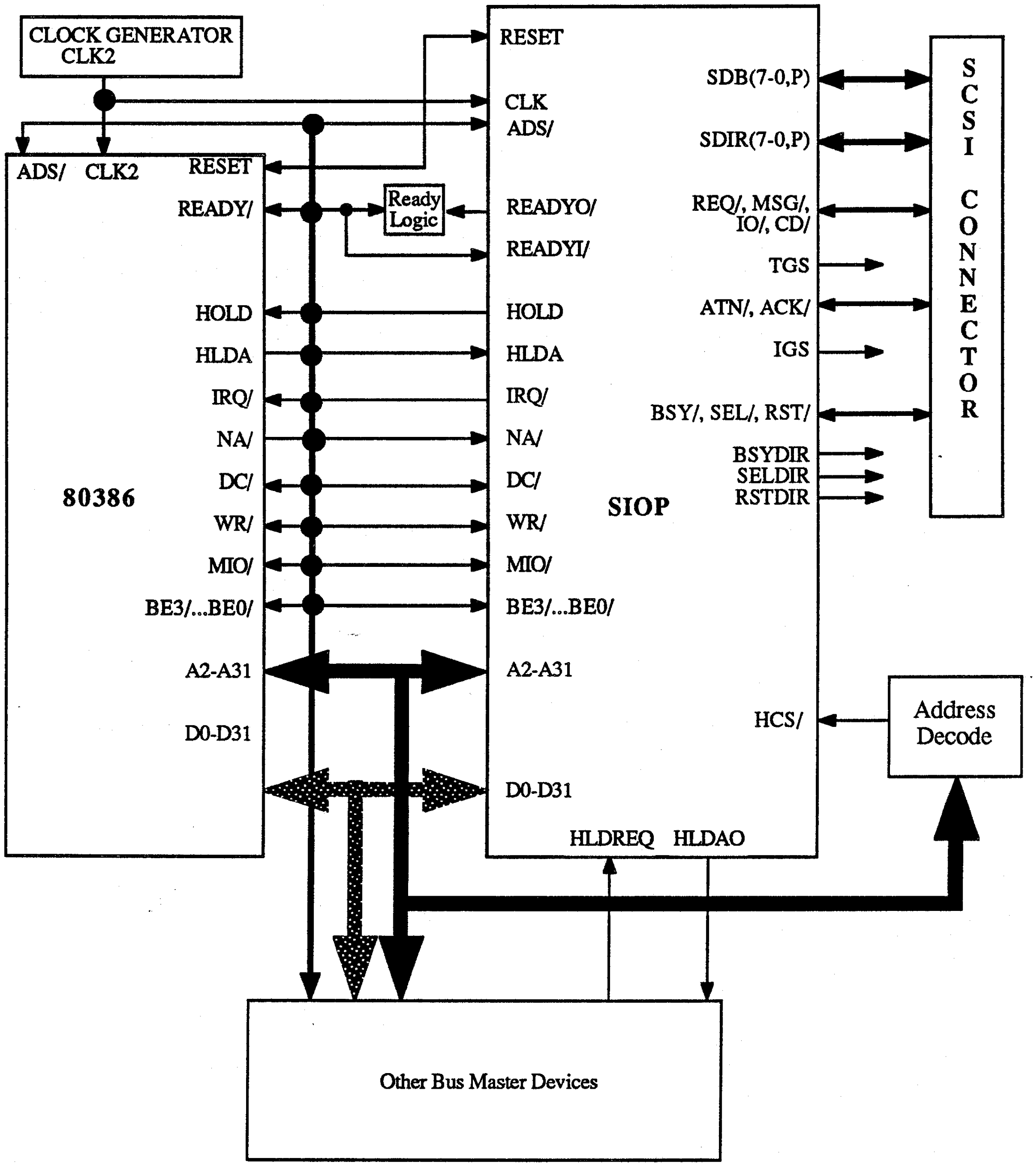


*Figure 3. SCSI I/O Processor Block Diagram*

The required pins for the 80386 version of the SIOP are summarized below.

*Table 1. SCSI I/O Processor Pinouts*

| Pin Name | Pin Description | Pin Number(s) |
|---|---|---|
| **80386 Signals** | | |
| DP3_ABRT/ | Host Data Parity / Abort | 1 |
| DP2:0 | Host Data Parity | 3 |
| D31:00 | Host Data Bus | 32 |
| A31:2 | Host Address Bus | 30 |
| BE3/:0/ | Byte Enables | 4 |
| WR/, DC/, MIO/ | Bus Cycle Control lines | 3 |
| ADS/, NA/ | Address strobe, & Next Address | 2 |
| READYI/, READYO/ | Ready In & Ready Out | 2 |
| HOLD, HLDREQ | Peripheral request, IOP Hold Request | 2 |
| HLDAI,HLDAO | Hold Ack In & Out | 2 |
| RESET | Reset | 1 |
| IRQ/ | Interrupt Request | 1 |
| HCS | Chip Select | 1 |
| CLK | Clock input | 1 |
| Total | | 85 |
| **SCSI Signals** | | |
| SDB7:0,SDBP | SCSI Data bus | 9 |
| SDIR7:0, SDIRP | SCSI Data Direction signals | 9 |
| REQ, MSG, I/O, C/D | Target signals | 4 |
| ATN, ACK | Initiator signals | 2 |
| TGS, IGS | Target & Initiator group select | 2 |
| BSY/, SEL/, RST/ | Control input signals | 3 |
| BSYDIR, SELDIR, RSTDIR | Control output signals | 3 |
| Total | | 32 |
| **Reserved Signals** | | |
| Reserved for Wide SCSI | | 12 |
| Chip Total | | 129 |

# Chapter 3
# Pinout Description

There are two groups of signals - SCSI signals and 80386 signals. The NCR 53C700 SIOP386 will be referred to as the SIOP throughout this document. A slash ("/") indicates an active-low signal. An underscore ("_") indicates a dual-function pin.

*Table 2. 80386/80286 Interface Signals*

| Pin # | Type | Signal | Description |
|---|---|---|---|
| 21-20<br>18-16<br>14-10, 8-7<br>5-3<br>160-157<br>155-154<br>152-150<br>148-144 | I/O | A31-A2 | Tri-state, active-high, Host address bus. These signals provide physical memory addresses or I/O port addresses dependent on the MIO/signal state. The 80386 drives the address lines in slave mode and the SIOP drives the address lines in master mode. In 80286 mode, the A31-24 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 A5-A2 in 80386 mode, and A5-A0 in 80286 mode. |
| 42-45<br>47-48<br>50-51,53<br>55-60,62<br>65-68, 70-73<br>76-78,80-81<br>83-85 | I/O | D31-D0 | Tri-state, active-high, Host data bus. These signals provide the data path between the 80386/80286 and the SIOP. Data can be transferred on 32-bit and 16-bit buses. This is controlled by bits 4 and 5 in the **DMODE** register. The **DMODE** register should not be changed while the DMA is active. Data outputs are synchronized with the CLK signal. |
| 52 | I/O | DP3_ABRT | Tri-state, active-low, Host data parity for byte lane 3 or abort. 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 (bit 2, the **SCNTL0** register) for this signal as an ABRT/signal. When ABRT/ is asserted, the DMA transfer is aborted. The ABRT input is asynchronous and requires a 40 nsec pulse width. When this signal is a parity signal, its output is synchronized with the CLK signal. |
| 63,74,86 | I/O | DP2-DP0 | Tri-state, active-high, Host data parity bits. These signals are the parity bits for the Host data bus (D23-D0). DP2 is the parity signal for data bits D23-D16, DP1 is the parity signal for data bits D15-D8, and DP0 is the parity signal for data bits D7-D0. 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 **SCNTLO** register), these signals are placed in a high-impedance state. Parity outputs are synchronized with the CLK signal. |
| 26 | I/O | BE3/ | Tri-state, active-low. This signal enables data transfer in the D31-D24 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 mode. The output is synchronized with the CLK signal. |

*Table 2. 80386/80286 Interface Signals (continued)*

| Pin # | Type | Signal | Description |
|---|---|---|---|
| 25 | I/O | BE2/_BHE/ | Tri-state, active-low. In 386 mode, this signal enables data transfer in the D23-D16 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. |
| 24 | I/O | BE1/_A1 | Tri-state, active-low, byte enable or address. In 386 mode, this signal enables data transfer in the D15-D8 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 mode. The output is synchronized with the CLK signal. |
| 23 | I/O | BE0/_A0 | Tri-state, active-low, byte enable or address. In 386 mode, this signal enables data transfer in the D7-D0 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. |
| 32 | I/O | R/W | Tri-state, Read/Write. 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. |
| 30 | O | DC/ | Tri-state, data control output. 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. |
| 29 | O | MIO/ | Tri-state, memory input/output. 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. |
| 28 | I/O | ADS/ | Tri-state, active-low, address status. ADS/ indicates that address and control signals are valid and stable. In Slave mode, this signal is driven by the 386/286. In Master mode, it is driven by the SIOP. The output is synchronized with the CLK signal. |

*Table 2. 80386/80286 Interface Signals (continued)*

| Pin # | Type | Signal | Description |
|---|---|---|---|
| 33 | I | NA/ | Active-low, next address request. 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. |
| 34 | I | READYI/ | Active low, ready transfer acknowledge. 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 to be inserted to extend the bus cycle if needed. The 53C700 always needs a READYI/ to terminate a read or write signal. |
| 27 | O | READYO/ | Active low, ready output signal. 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. |
| 40 | O | HOLD | Active high, hold request output. 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. |
| 37 | I | HLDREQ | Active high, bus hold request input. 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. |
| 41 | I | HLDAI | Active-high, bus hold acknowledge in. 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. |
| 38 | O | HLDAO | Active-high, bus hold acknowledge out. 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 Interface in Section 6 for more information on HOLD/HLDA schemes.

*Table 2. 80386/80286 Interface Signals (continued)*

| Pin # | Type | Signal | Description |
|---|---|---|---|
| 88 | I | RESET | Active-high, hardware reset. 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. The signal is also defines the 01 and 02 clock phases. |
| 87 | O | IRQ/ | Active-low, open drain, interrupt request. This signal is asserted in response to an interrupt condition or when a SCSI SCRIPTS TMinterrupt instruction is issued. This signal has an internal pull-up resistor. The output is synchronized with the CLK signal. |
| 35 | I | HCS/ | Active-low, Host chip select. This signal is generated by external address decoding to allow the SIOP's registers to be memory or I/O mapped. |
| 36 | I | CLK | Square wave clock which 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 with a 35% to 65% duty cycle. |
| 122<br>124-127<br>129-131, 121 | I/O | SDB7-<br>SDB0<br>SDBP | 48 mA, open drain, active-low, SCSI data/parity bus. |
| 133-135<br>137-138<br>140-142 | O | SDIR7-<br>SDIR0 | Active-high, SCSI data direction. In differential mode, these signals control the direction of external differential pair transceivers for the SD7/ - SD0/ 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. |
| 132 | O | SDIRP | Active-high, SCSI parity direction. 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. |
| 120 | O | ATN/ | 48 mA open drain, SCSI attention. 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. |
| 115 | I/O | MSG/ | 48 mA open drain, SCSI message phase signal. 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. |

*Table 2. 80386/80286 Interface Signals (continued)*

| Pin # | Type | Signal | Description |
|---|---|---|---|
| 110 | I/O | I/O | 48 mA, open drain, SCSI input-output phase. 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. |
| 112 | I/O | C/D | 48 mA, open drain, SCSI control-data phase. 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 of 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 | | | |

| Pin # | Type | Signal | Description |
|---|---|---|---|
| 111 | I/O | REQ/ | 48 mA, open drain, active-low, SCSI data transfer request. 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. |
| 117 | I/O | ACK/ | 48 mA open drain, active-low, SCSI acknowledge. 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. |
| 119 | I/O | BSY/ | 48 mA open drain, active-low, SCSI busy. 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. |

*Table 2. 80386/80286 Interface Signals (continued)*

| Pin # | Type | Signal | Description |
|---|---|---|---|
| 116 | I/O | RST/ | 48 mA open drain, active-low,SCSI reset. 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. |
| 114 | I/O | SEL/ | 48 mA open drain, active-low, SCSI select. This signal is asserted to select or reselect another SCSI device. This signal can be directly connected to the single-ended SELDIR line. In differential mode, it is an input only and the SELDIR signal asserts SEL/ on the SCSI bus. |
| 108 | O | BSYDIR | SCSI busy/direction. This signal controls the assertion of the SCSI BYS/ 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. |
| 107 | O | RSTDIR | 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. |
| 106 | O | SELDIR | 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. |
| 103 | O | TGS | 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. |
| 104 | O | IGS | 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 4
# Registers

This chapter contains descriptions of all 53C700 registers.

*Table 3. Register Addresses and Descriptions*

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

|  |  |  |  | Address (Hex) |
|---|---|---|---|---|
| **SIEN** (R/W) | **SDID** (R/W) | **SCNTL1** (R/W) | **SCNTL0** (R/W) | 00 |
| **SOCL** (R/W) | **SODL** (R/W) | **SXFER** (R/W) | **SCID** (R/W) | 04 |
| **SBCL** (R) | **SBDL** (R) | **SIDL** (R) | **SFBR** (R) | 08 |
| **SSTAT2** (R) | **SSTAT1** (R) | **SSTAT0** (R) | **DSTAT** (R) | 0C |
| RESERVED |  |  |  | 10 |
| **CTEST3** (R) | **CTEST2** (R) | **CTEST1** (R) | **CTEST0** (R) | 14 |
| **CTEST7** (R/W) | **CTEST6** (R/W) | **CTEST5** (R/W) | **CTEST4** (R/W) | 18 |
| **TEMP** |  |  |  | 1C |
| RESERVED |  | **ISTAT** (R/W) | **DFIFO** (R/W) | 20 |
| **DCMD** (R/W) | **DBC** (R/W) |  |  | 24 |
| **DNAD** (R/W) |  |  |  | 28 |
| **DSP** (R/W) |  |  |  | 2C |
| **DSPS** (R/W) |  |  |  | 30 |
| RESERVED |  |  | **DMODE** (R/W) | 34 |
| **DCNTL** (R/W) | **DWT** (R/W) | **DIEN** (R/W) | RESERVED | 38 |
| RESERVED |  |  |  | 3C |

*Figure 4. NCR 53C700 Register Address Map*

## 4.1 53C700 Register Descriptions

*Notes:*

*NCR reserved bits are designated as "RES" in each register map. These bits should always be written to 0, discard all information read from them.*

*Unless otherwise indicated, all bits in the registers are active high. The "Default" notation refers to the SIOP's register values after the chip is powered-up or reset. The 8-bit registers can be addressed one, two, or four at a time with the exception of the DFIFO register. This register must be accessed by an 8-bit read or write ONLY.*

## 4.2 SCSI Registers

### Register 00 SCSI Control<br>Register 0 (SCNTL0)

*Read/Write*

| | ARB1 | ARB0 | START | WATN/ | EPC | EPG | AAP | TR |
|---|---|---|---|---|---|---|---|---|
| | 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,<br>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 threshold of the connected bit in the **SSTAT1** register to verify that SIOP is connected to the SCSI bus before starting any arbitration sequences in Low-Level mode. 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, then it asserts BSY/ and asserts its SCSI ID contained in the **SODL** register onto the SCSI bus. 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.

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 53C700 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 timeout occurs, the Selection Timeout bit is set to 1 in the **SSTAT0** register.

### Bit 5 START Start Sequence

When this bit is written to 1, one of the arbitration or selection sequences will start. The SIOP will start the sequence 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. This bit is cleared automatically when the selection sequence is complete.

### Bit 5 (cont)

The arbitration sequence can be aborted by resetting this bit to 0. If the sequence is aborted, check the connected bit in the **SSTAT0** register to verify that the SIOP is not connected on the SCSI bus.

### Bit 4 WATN Select with ATN/ on a Start Sequence

When this bit is set to 1, the SCSI ATN/ signal is asserted during the selection phase. ATN/ is asserted when BSY/ is deasserted while selecting a target device. When executing SCSI SCRIPTS™, this bit is controlled by the SCRIPTS PROCESSOR, use it in Low-Level mode. While attempting to select a target device and a selection timeout 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 this bit is set to 1, the SCSI data bus is checked for odd parity when data is received across the SCSI bus in either initiator or target mode. The host data bus is checked for odd parity if bit 2, the Enable Parity Generation/Parity Through bit, is written to 0. Host data bus parity is checked as data is loaded into the **SODL** register when sending SCSI data in either initiator or target mode. If a parity error is detected, bit 0 of the **SSTAT0** register is set to 1 and optionally an interrupt can be generated.

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. A phase change to Message In phase must be generated to indicate this condition (**SXFER** register, bit 7 = 0)

If the SIOP is operating in initiator mode and a parity error is detected, ATN/ can optionally be asserted (bit 1), but the transfer continues until the target changes phase to Message Out. If this bit is written to 0, then parity errors are not reported.

### Bit 2 EPG Enable Parity Generation/Parity Through

When this bit is 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. If this bit is written to 0, then the parity present on the host data parity lines, DP3 - DP0 will flow through the SIOP's internal FIFOs and be enabled onto the SCSI bus when sending data. To enable the DP3_ABRT/ signal so that it functions as an Abort signal (ABRT/), write this bit to 1.

### Bit 1 AAP Assert ATN/ on Parity Error

When this bit is 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 following parity errors can occur: a parity error detected on data received from the SCSI bus or a parity error detected on data transferred to the SIOP from the host data bus. 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/ in this manner.

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 automatically asserted on the SCSI bus when a parity error is received.

### Bit 0 TRG Target Mode

When this bit is 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 written to 0, the SIOP is an initiator device.

## Register 01 SCSI Control
## Register 1 (SCNTL1)

*Read/Write*

|  | EXC | ADB | ESR | CON | RST | PAR | 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 this bit is set to 1, an extra clock period of data setup is added to each SCSI data transfer. The extra clock cycle can provide additional system design margin. However, the extra setup time may affect the SCSI data transfer rates. Resetting this bit to 0 disables the extra clock cycle of data setup when sending SCSI data.

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

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

As an initiator, to assert **SODL**, the phase lines (MSG/, C/D, I/O) in the **SOCL** register must be written to match the phase asserted by the target. 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 diagnostics testing or operation in Low-Level mode.

### Bit 5 ESR Enable the SIOP to respond to Selection & Reselection

When this bit is set to 1, the SIOP is enabled to respond to bus-initiated selections and reselections. The SIOP can respond to selections and reselections in both initiator and 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 anytime the SIOP becomes connected as an initiator or a target. It is set to 1 after successfully completing arbitration or when the SIOP has responded to a bus-initiated selection or reselection attempt. It should be written to 1 after successfully completing simple arbitration when operating in Low-Level mode. When this bit is 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 PAR Assert even SCSI Parity (force bad parity)

When this bit is set to 1 and the Enable Parity Generation bit is set in the **SCNTL0** register, the SIOP asserts even parity. It forces a SCSI parity error on each byte sent to the SCSI bus from the SIOP. If parity checking is enabled, then the SIOP checks data received for even parity. Use this bit for diagnostic testing, set 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 are 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 Low-Level operation.

### 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 Low-Level operation.

## Register 02 SCSI Destination ID Register (SDID)

*Read/Write*

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

### Bits 7 - 0 ID7 - ID0 SCSI ID 7 - 0

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 .

## Register 03 SCSI Interrupt Enable Register (SIEN)

*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 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.

In target mode, this bit is the ATN/ Active bit. When the ATN/ active bit is 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 this bit is 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 Timeout

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

### Bit 4 SEL Selected or Reselected

When this bit is 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 this bit is 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 or the synchronous offset caused the SCSI FIFO to be overwritten.

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 this bit is set to 1, the IRQ/ signal is asserted when a target device unexpectedly disconnects from the SCSI bus. This bit is valid only 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). 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 this bit is 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 this bit is set to 1, the IRQ/ signal is asserted if the SIOP detects a parity error while sending or receiving SCSI data.

Parity checking must be enabled (**SCNTL0** register, bit 3). In initiator mode, an interrupt is not generated until the transfer is complete or until the target changes phases.

In 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 SCSI or DMA FIFOs 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.

This interrupt is masked by writing this bit to 0.

## Register 04 SCSI Chip ID Register (SCID)

*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 sets up 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.

## Register 05 SCSI Transfer Register (SXFER)

*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 SCSI or DMA FIFOs 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 this bit is set to 1, the SIOP does not halt the SCSI transfer when ATN/ or a parity error is received.

**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 and their relationship to the synchronous data transfer period used by the SIOP.

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

The minimum Synchronous Transfer Period:

when sending SCSI data

= TCP * (4 + XFERP + 1)

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

= TCP * (4 + XFERP)

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

when receiving SCSI data

= TCP * (4 + XFERP)

where

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)

*Table 4. 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 |

*Table 4a. Synchronous Transfer Period Examples*

| CLK (MHz) | SCSI CLK ÷ DCNTL bit 7 | XFERP | Min Synch Transfer Period (nsec) | Synch Transfer Period (MBytes/ sec) |
|---|---|---|---|---|
| 50 | ÷2 | 0 | 160 | 6.25 |
| 40 | ÷2 | 0 | 200 | 5 |
| 37.50 | ÷1.5 | 0 | 160 | 6.25 |
| 33 | ÷1.5 | 0 | 181.82 | 5.5 |
| 25 | ÷1 | 0 | 160 | 6.25 |
| 20 | ÷1 | 0 | 200 | 5 |
| 16.67 | ÷1 | 0 | 239.95 | 4.17 |

### 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 next 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.

*Table 5. SCSI Synchronous Offsets Used by the SIOP*

| MO3 | MO2 | MO1 | MO0 | Synchronous Offset |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 - Asynchronous Operation |
| 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 |
| 1 | 0 | 0 | 1 | Reserved - Not Used |
| 1 | 0 | 1 | 0 | Reserved - Not Used |
| 1 | 0 | 1 | 1 | Reserved - Not Used |
| 1 | 0 | 0 | 0 | Reserved - Not Used |
| 1 | 0 | 0 | 1 | Reserved - Not Used |
| 1 | 0 | 1 | 0 | Reserved - Not Used |
| 1 | 0 | 1 | 1 | Reserved - Not Used |
| 1 | 1 | 0 | 0 | Reserved - Not Used |
| 1 | 1 | 1 | 1 | Reserved - Not Used |

## Register 06 SCSI Output Data Latch Register (SODL)

*Read/Write*

| | SD7<br>7 | SD6<br>6 | SD5<br>5 | SD4<br>4 | SD3<br>3 | SD2<br>2 | SD1<br>1 | SD0<br>0 |
|---|---|---|---|---|---|---|---|---|
| Default >>> | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### Bits 7 - 0 SD7 - SD0 Bit 7-SCSI Output Data Latch Bit 0-SCSI Output Data Latch

This register is used primarily for diagnostics testing or programmed I/O operation. 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 to assert the contents on this register. Use this register to send data via programmed I/O. Data flows through this register when sending data in any mode. It is also used to write to the synchronous data FIFO when testing the chip.

## Register 07 SCSI Output Control Latch Register (SOCL)

*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 programmed I/O operation. 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 the 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.

## Register 08 SCSI First Byte Received Register (SFBR)

*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 |
|---|---|---|---|---|---|---|---|

**Bits 7 - 0 1B7 - 1B0**
**Bit 7 - First Byte Received**
**Bit 0 - First Byte Received 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

This register contains the SCSI ID of a selecting or reselecting device after a selection or reselection. The register contents will change after a Block Move Instruction is executed for receiving data.

## Register 09 SCSI Input Data Latch Register (SIDL)

*Read Only*

| | 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 |

<u>**Bits 7 - 0**</u> **SD7 - SD0**

**Bit 7 - SCSI Input Data Latch Register**

**Bit 0 - SCSI Input Data Latch Register**

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 out to the host FIFO. This register differs from the **SBDL** register, this register contains latched data and the **SBDL** always contains exactly what is currently on the SCSI data bus.

## Register 0A SCSI Bus Data Lines Register (SBDL)

*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 |

<u>**Bits 7 - 0**</u> **SD7 - SD0**
**SCSI Data Bit 7 - SCSI Data Bit 0**

This register contains the SCSI data bus status. Even though the SCSI data bus is active-low, these bits are active-high. 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.

### Register 0B SCSI Bus Control Lines Register (SBCL)

*Read Only*

| | 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 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**

This register contains the SCSI control bus status. Even though the SCSI data bus is active-low, these bits are active-high. The signal status is not latched and is a true representation of exactly 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.

## 4.3 Status Registers

### Register 0C DMA Status Register (DSTAT)

*Read Only*

| | DFE | RES | RES | ABRT | SSI | SIR | WTD | OPC |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 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**

These bits are 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 by another device (Parity Generation mode) or a software abort command is issued by setting Bit 7 of the **ISTAT** register.

**Bit 3 SSI SCRIPT 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 SCRIPT Single Step Interrupt:

1. In the Single Step Mode bit (DCNTL register, bit 4) equals 1, there will be a SCRIPT Single Step interrupt after successfully executing each SCRIPT instruction.
2. The SIOP encounters a branch condition while executing pipelined instructions (pipeline mode, DCNTL register, bit 3).

### Bit 2 SIR SCRIPT Interrupt Instruction Received

This status bit is set whenever an INT instruction is executed.

### Bit 1 WTD Watchdog Timeout Detected

This status bit is set when the Watchdog Timer Counter has decremented to zero. This only applies when the 53C700 is in Master mode. If this counter decrements to zero, it indicates that the memory device did not assert its READYI/ signal within the specified timeout period.

### Bit 0 OPC Illegal Instruction Detected

This status bit is set anytime an illegal instruction is fetched.

*NOTE:*

If executing 8-bit reads of the **DSTAT** and **SSTAT0** registers to clear interrupts, insert 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. NOP
3. NOP
4. Read **SSTAT0** to clear the SCSI interrupt.

## Register 0D SCSI Status Register 0 (SSTAT0)

*Read Only*

| M/A | CMP | STO | SEL | SGE | UDC | RST/ | PAR |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 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 Phase mismatch - Initiator mode or ATN/ active - Target mode

In 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 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 Timeout

This bit is set to 1 when a selection or reselection timeout occurs. A timeout occurs when the device trying to be selected or reselected did not respond within the specified 250 msec timeout 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 and Reselection bit must be set to 1 in the **SCNTL1** register for the SIOP to respond to selection and reselection attempts.

**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 or the synchronous offset caused the SCSI FIFO to be overwritten.
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 he SIOP is operating as an initiator.

**Bit 2 UDC Unexpected Disconnect**

This bit is set to 1 when the SIOP is operating in initiator mode and the target device unexpectedly disconnects from the SCSI bus. This bit is only valid when the SIOP operates in the initiator mode. When the SIOP is executing SCSI SCRIPTS™, an unexpected disconnect is defined to be a disconnect that does not occur after receiving either a Disconnect Message (04h) or a Command Complete Message (00h). When the SIOP operates in Low-Level Mode, any disconnect can cause an interrupt, even a valid SCSI disconnect.

**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 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. NOP
3. NOP
4. Read **SSTAT0** to clear the SCSI interrupt.

## Register 0E SCSI Status Register 1 (SSTAT1)

*Read Only*

| ILF | OLF | ORF | 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.

### Bit 6 ORF SODR Register Full

This bit is set to 1 when the SCSI Output Data Register (**SODR**, a hidden buffer register which is not accessible) 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.

### 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 SCSI Output Data Register (**SODR**, a hidden buffer register which is not accessible), and then to the **SODL** register 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.

### 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.

## Register 0F SCSI Status Register 2 (SSTAT2)

*Read Only*

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

Default >>>

### 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.

*Table 6. 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. They are useful when operating in Low-Level mode.

## Registers 10-13

### Reserved

## 4.4 Chip Test Registers

### Register 14 Chip Test Register 0 (CTEST0)

*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.

### Register 15 Chip Test Register 1 (CTEST1)

*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.

## Register 16 Chip Test Register 2 (CTEST2)

*Read Only*

| RES<br>7 | RES<br>6 | SOFF<br>5 | SFP<br>4 | DFP<br>3 | TEOP<br>2 | DREQ<br>1 | DACK<br>0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |

Default >>>

### 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 when the **CTEST6** register reads data out of the 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 DACK Data Acknowledge Status

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

## Register 17 Chip Test Register 3 (CTEST3)

*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 |

<u>Bits 7 - 0</u> **SF7 - SF0**
**Bit 7 - SCSI FIFO**
**Bit 0 - SCSI FIFO**

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 in the **CTEST2** register. The FIFO Full Bits in the **SSTAT2** register can be read to determine how many bytes currently reside in the SCSI Synchronous FIFO. Reading this register when the SCSI FIFO is empty causes a SCSI Gross Error (FIFO underflow).

## Register 18 Chip Test Register 4 (CTEST4)

*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 |

<u>Bit 7</u> **RES** **Reserved**

<u>Bit 6</u> **ZMOD** **Z Mode - High-Impedance Mode**

Writing this bit to 1 makes the SIOP place <u>all</u> 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. Do not use this bit in a system environment, all outputs are tri-stated and this may cause problems in a systems application.

<u>Bit 5</u> **SZM** **SCSI Z Mode - SCSI High-Impedance Mode**

Setting this bit to 1 causes the SIOP to place certain 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 <u>not</u> be in a high-impedance state. In order to transfer data on the SCSI bus, this bit must be written to 0.

<u>Bit 4</u> **SLBE** **SCSI Loopback Enable**

Setting this bit to 1 enables "Loopback" Mode. Loopback allows the user to assert any SCSI signal. 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

*Table 7. Byte Lane Selection for 32-bit DMA FIFO*

| FBL2 | FBL1 | FBL0 | DMA FIFO Byte Lane |
|---|---|---|---|
| 0 | X | X | None - FIFO access disabled |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 2 |
| 1 | 1 | 1 | 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.

## Register 19 Chip Test Register 5 (CTEST5)

*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. In 386 mode, 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, if the current address is on a 16-bit word or a 32-bit longword boundary (**DNAD** contents, the bus width) and the host interface mode used (286 or 386 mode). 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. 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 control bit controls the operation of bits 3 - 0. When this bit is 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.

## Register 1A Chip Test Register 6 (CTEST6)

*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 |

**Bits 7 - 0** **DF7 - DF0**
**Bit 7-DMA FIFO**
**Bit 0-DMA FIFO**

Writing 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 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 SCRIPT$^{TM}$.

## Register 1B Chip Test Register 7 (CTEST7)

*Read Only*

| RES | RES | RES | STD | DFP | EVP | DC | DIFF |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Default >>> | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7** **RES** **Reserved**

**Bit 6** **RES** **Reserved**

**Bit 5** **RES** **Reserved**

**Bit 4** **STD** **Selection Timeout Disable Bit Setting**

This bit disables the selection timeout timer.

**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. 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 via programmed I/O, write this bit to the FIFO as the parity bit for each byte loaded. When writing data to the DMA FIFO, set this bit with the status of the parity bit to be written to the FIFO *before* writing the byte to the FIFO. For the details of performing a diagnostic test of the DMA FIFO, please refer to section 6.4, Diagnostics DMA FIFO Test.

### Bit 2 EVP Even Parity

Setting this bit to 1 causes the SIOP to generate even parity when sending data to the host bus. Resetting this bit to 0 causes the SIOP to maintain odd parity throughout the chip.

### 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 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. When this bit is reset to 0 and the SIOP fetches instructions from memory, 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. Resetting this bit enables single-ended mode. This bit should be set to 1 in the initialization routine if the differential pair interface is to be used.

## Registers 1C-1F Temporary Stack Register (TEMP)

*Read/Write*

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

## Register 20 DMA FIFO Register (DFIFO)

*Read/Write*

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

The DFIFO register can only be accessed by 8-bit reads or writes, because 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 this bit is set to 1, data residing in the DMA FIFO is transferred to or from memory (according to the internal DMAWR signal) starting at the address in the **DNAD** register. The internal DMAWR signal controlled by the **CTEST5** register, bit 3 determines the direction of the transfer. Once the SIOP has successfully transferred the data, this bit should be written to 0.

### Bit 6 CLF Clear DMA and SCSI FIFOs

When this bit is set to 1, all data pointers for the SCSI and DMA FIFOs are cleared. In addition to the SCSI and DMA FIFO pointers, 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 - BO0
### Bit 5-FIFO Byte Offset Counter
### Bit 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. To preserve the Byte Offset Counter bits, read, then write the value read back to these bits.

To determine how many bytes reside in the DMA FIFO when an error occurs, 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 or 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** registêr

6) Mask any carry bits by ANDing the result with 3F hex

7) The final result will be between 0 and 32 bytes

## Register 21 Interrupt Status Register (ISTAT)

*Read/Write*

| ABRT<br>7 | RES<br>6 | RES<br>5 | RES<br>4 | CON<br>3 | PRE<br>2 | SIP<br>1 | DIP<br>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 in the SCRIPT operation of the SIOP. Use it as a polling register if interrupts are not enabled. There may be stacked interrupts pending. When polling, read this register after clearing an interrupt to check for stacked interrupts.

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 Reserve registers (23h and 22h), **ISTAT** register (21h) and the **DFIFO** register (20h). The **ISTAT** register will read valid data but, the **DFIFO** register will show FFs. Therefore, the **DFIFO** register can only be accessed by an 8-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 an interrupt is received, reset this bit 0 before 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 this **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 aborted interrupt and to see if any other interrupting conditions have occurred.

**Bits 6-4 RES Reserved**

**Bit 3 CON Connect/disconnect**

This status bit is set to 1 when the 53C700 has won arbitration on the SCSI bus. It is reset to 0 when the 53C700 is disconnected from the SCSI bus.

This bit is automatically set anytime the SIOP becomes connected as an initiator or a target. It is set to 1 after successfully completing arbitration or when the SIOP has responded to a bus-initiated selection or reselection attempt.

**Bit 2 PRE Pointer Register Empty**

This status bit is set 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.

**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 Timeout 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) SCRIPT single step interrupt received

3) SCSI SCRIPT Interrupt instruction

4) Watchdog timer counter decremented to zero, indicating that a host memory timeout occurred

5) Illegal SCRIPT instruction detected

*NOTE:*

If executing 8-bit reads of the **DSTAT** and **SSTAT0** registers to clear interrupts, insert 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. NOP
3. NOP
4. Read **SSTAT0** to clear the SCSI interrupt.

## 4.5 DMA Registers

### Registers 24-26 DMA Byte Counter Register (DBC)

*Read/Write*

Default >>> all zeros

This 24-bit register determines 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.

### Register 27 DMA Command Register (DCMD)

*Read/Write*

Default >>> all zeros

This 8-bit register determines the instruction for the SIOP to execute. It contains the opcode of a SCRIPT instruction that has been fetched. This register has a different function for each instruction. For a complete description, please refer to the instruction set of the SIOP in Section 5.

## Registers 28-2B DMA Next Address For Data (DNAD)

*Read/Write*

Default >>> all zeros

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

*Note*
*In future generations of the 53C700 family, this register will only be used for Block Move Instructions.*

## Registers 2C-2F DMA SCRIPTS Pointer Register (DSP)

*Read/Write*

Default >>> all zeros

If the SIOP is executing SCSI SCRIPTS™, the address of the first SCSI SCRIPT™ should be written to this register. In normal SCRIPT operation, once the start address of the SCSI SCRIPT™ is written to this register, the SCRIPT instructions are automatically fetched and executed until an interrupt condition occurs.

In single step mode, there is a SCRIPT single step interrupt after each 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 SCSI SCRIPT.

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

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

## Register 30-33 DMA SCRIPTS Pointer Save Register (DSPS)

*Read/Write*

Default >>> all zeros

This 32-bit register contains the second longword of a Select, Reselect, Jump, Call, Return, or Interrupt SCRIPT 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.

## Register 34 DMA Mode (DMODE)

*Read/Write*

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

### Bit 7 BL1 Burst Length Bit 1

### Bit 6 BL0 Burst Length Bit 1

*Table 8. 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 determined by the host bus width and whether the bytes are transferred to or from an odd-byte boundary.

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 this bit is set to 1, the SIOP executes Block Move instructions by transferring data 16-bits at a time. Writing this bit to 1 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 this bit is set to 1, the SIOP operates in 80286 mode. In 80286 mode, 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 SCRIPT instructions are fetched 16-bits at a time. Initialize this bit first if the SIOP is to operate in an 80286 system.

### 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. This bit does not have an effect on instruction fetch operations, it only applies to data being transferred to/from memory. Writing this bit to 1 causes the SIOP to transfer data to an I/O-mapped device. Writing this bit to 0 causes the SIOP to transfer data to a memory-mapped device. 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 address pointer so that it will not increment after each data transfer. The address pointer is located in the **DNAD** register. If this bit is 0, this pointer increments after each data transfer. If this bit is 1, this pointer will not increment after each data transfer. Use this pointer to transfer data to/from one port address, i.e. a serial port.

### 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.

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

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

### 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.

## Register 39 DMA Interrupt Enable Register (DIEN)

*Read/Write*

| | RES | RES | RES | ABRT | SPI | SSI | 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 any abort condition. Abort conditions can occur in two ways: the DP3_ABRT/ input signal is asserted by another device (Parity Generation mode) 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 SCRIPT Single Step Interrupt**

Writing this bit to 1 asserts the IRQ/ signal when the SCRIPT Single Step Interrupt occurs. Resetting this bit to 0 disables the assertion of IRQ/ when a SCRIPT 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 SCRIPT Single Step Interrupt after successfully completing each instruction.

2) If the SIOP encounters a branch condition while executing pipelined instructions (Pipeline mode, **DMODE** register, bit 1).

**Bit 2 SIR Enable SCRIPT Interrupt Instruction Received Interrupt**

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

**Bit 1 WTD Enable Watchdog Timeout Interrupt**

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

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

## Register 3A DMA Watchdog Timer Register (DWT)

*Read/Write*

Default >>> all zeros

The DMA Watchdog Timer Register provides a timeout 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 timeout value to this register during initialization. Every time that the SIOP transfers data to/from memory, the value stored in this register is loaded into the counter. Disable the timeout 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 timeout of 50 µsec was desired, then at 50 MHz this register should be loaded with a value of 9D hex.

## Register 3B DMA Control Register (DCNTL)

*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.

*Table 9. SIOP Input Clock Frequencies for Bit 6, Register 3B*

| CF1 | CF0 | Clock Frequency | Internal divide by fo r SCSI core clock |
|---|---|---|---|
| 0 | 0 | 37.51 - 50 MHz | 2 |
| 0 | 1 | 25.01 - 37.50 MHz | 1.5 |
| 1 | 0 | 16.67 - 25.00 MHz | 1 |
| 1 | 1 | Reserved | Reserved |

These two bits determine the clock period used by the SCSI portion of the SIOP to comply with the ANSI timings. If CF1=0 and CF0=0, the clock period used by the SCSI core is the CLK input divided by 2. If CF1=0 and CF0=1, the clock period used by the SCSI core is the CLK input divided by 1.5. If CF1=1 and CF0=0, the clock period used by the SCSI core is the CLK input divided by 1.

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

SCSI SCRIPTS™ instructions are fetched 16-bits at a time when this bit is written to 1. 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. SCSI SCRIPTS™ instructions are fetched 32-bits at a time when this bit is written to 0.

### Bit 4 SSM Single Step Mode

Writing this bit to 1 stops the SIOP after completing each instruction. The SCRIPT Single Step Interrupt bit in the **DSTAT** register becomes 1 after each instruction is executed. If the SCRIPT 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 SCRIPT Single Step Interrupt and then set the START DMA (bit 2)in this register.

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

Writing this bit to 1 places the SIOP in the Low-Level Mode. 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. Writing this bit to 0 disables Low-Level Mode.

### Bit 2 STD Start DMA Operation

The SIOP fetches a SCSI SCRIPT™ 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 respective default values 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.

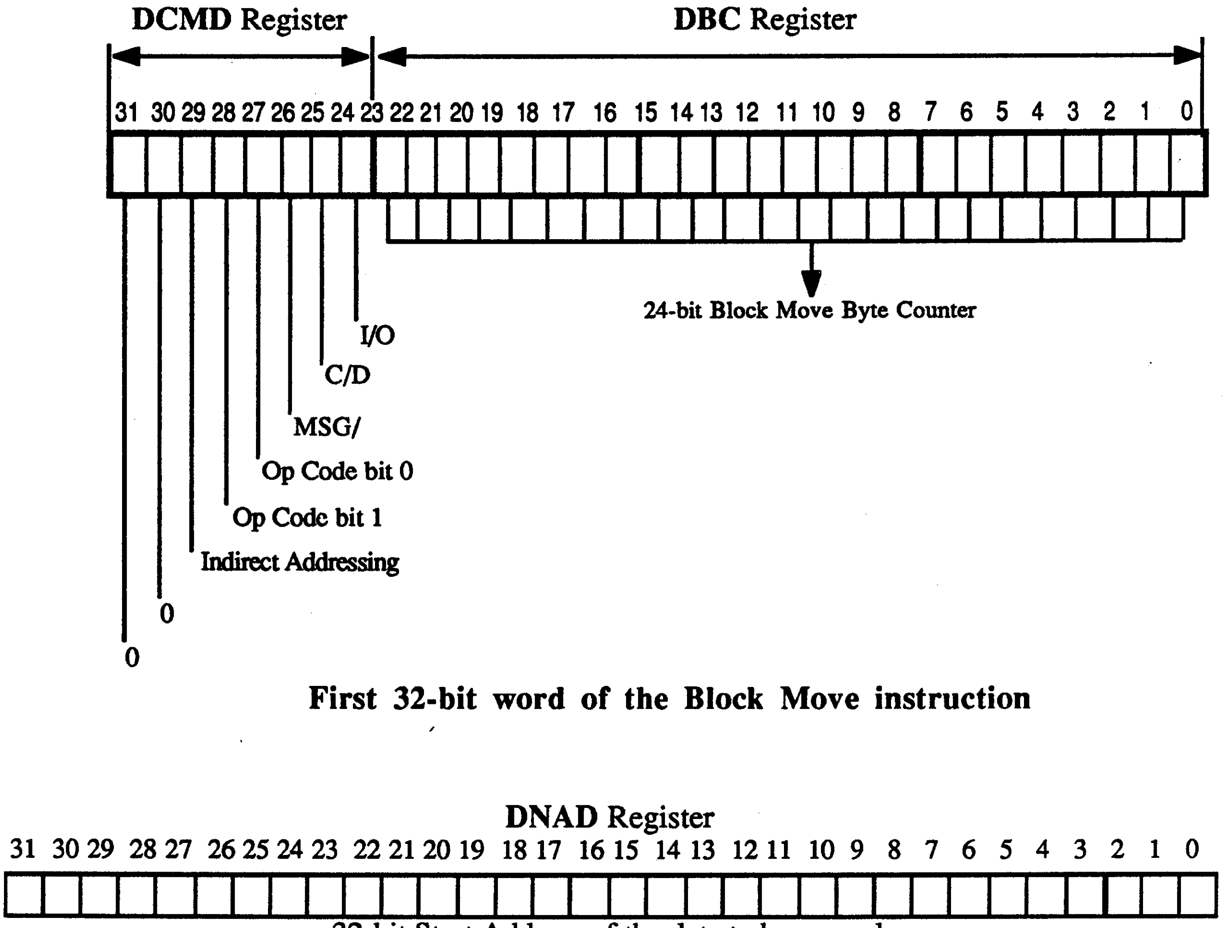
# Chapter 5
# Instruction Set of the SCSI I/O Processor

The SCSI I/O Processor fetches and executes its own instructions by becoming a bus master and loading two 32-bit words into its registers. This is referred to as executing SCSI SCRIPTS™. The SCSI SCRIPTS™ mode of executing instructions allows the SIOP to make decisions based on the status of the SCSI bus.

It also off-loads the microprocessor from servicing numerous interrupts. Instructions can also be executed by pipelining them in one at a time using pipeline mode.

There are three types of instructions implemented: Block Move Instructions, I/O

## 5.1 Block Move Instructions



**First 32-bit word of the Block Move instruction**

**Second 32-bit word of the Block Move instruction**

*Figure 5. Block Move Instruction Register*

### Indirect Addressing Field (bit 29)

When set to 0, SCSI or user data is moved to(from) the 32-bit data start address for the block move.

The value is loaded into the chip's address register and incremented as data is transferred.

When set to 1, the 32-bit SCSI or user data start address for the Block Move 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 long word (four byte transfer across the Host computer bus).

This option implies three DMA long word 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 specification of a table of data buffer addresses. 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. Also this feature makes it possible to locate SCSI SCRIPTS in a PROM.

### Op Code Field (bits 28,27)

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

### Target Mode

| OPC1 | OPC0 | Instruction defined |
|---|---|---|
| 0 | 0 | **MOVE** - Block Move Instruction |
| 0 | 1 | Reserved - An Illegal Instruction Interrupt will occur |
| 1 | 0 | Reserved - An Illegal Instruction Interrupt will occur |
| 1 | 1 | 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 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 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 | **MOVE**-Reserved, DO NOT use |
| 0 | 1 | **WMOV** - Wait Block Move Instruction |
| 1 | 0 | Reserved - An Illegal Instruction Interrupt will occur |
| 1 | 1 | Reserved - An Illegal Instruction Interrupt will occur |

### WMOV 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.

### Phase Field (MSG, C/D, & I/O) (bits 26, 25, 24)

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.

*Table 10. Phase Field Definitions for SCSI Information Transfer Phase*

| 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 standardization |
| 1 | 0 | 1 | Reserved for future standardization |
| 1 | 1 | 0 | Message Out |
| 1 | 1 | 1 | Message In |

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

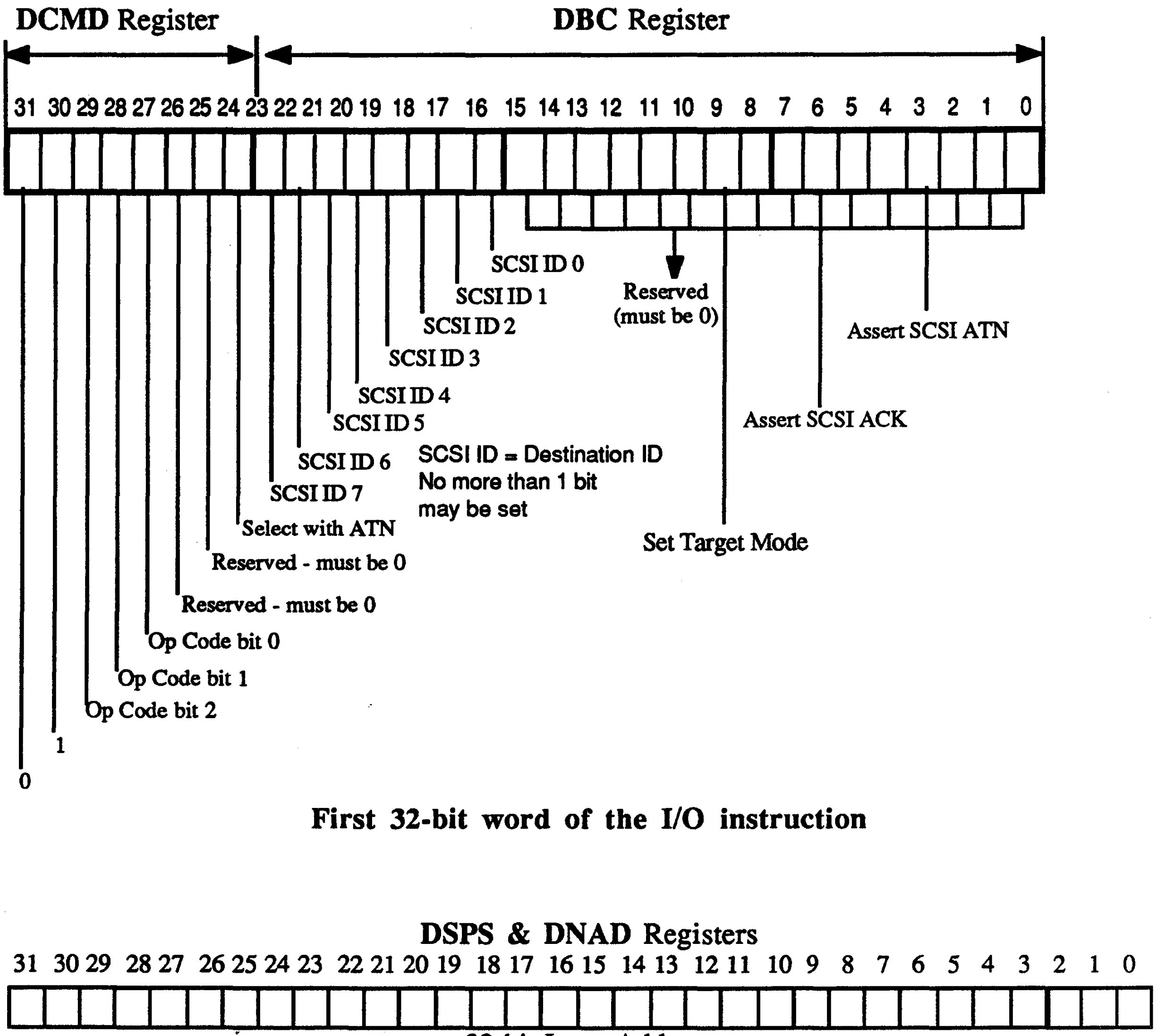
## Transfer Counter Field (bits 23-0, DBC register)

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 .

## 5.2 I/O Instructions



**First 32-bit word of the I/O instruction**

**Second 32-bit word of the I/O instruction**

*Figure 6. I/O Instruction Register*

*Note*
*In future generations of the 53C700 family, the second 32-bit word of the I/O instruction will be loaded into the DSPS only.*

### Op Code Field (bits 29, 28, 27)

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

### Target Mode

*Table 11. Target Mode Instruction Descriptions for I/O Instructions*

| OPC2 | OPC1 | OPC0 | 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 |
| 1 | 0 | 1 | Reserved - An Illegal Instruction Interrupt will occur |
| 1 | 1 | 0 | Reserved - An Illegal Instruction Interrupt will occur |
| 1 | 1 | 1 | Reserved - An Illegal Instruction Interrupt will occur |

### RESELECT Instruction

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

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

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

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

### CLEAR Instruction

1) When the Assert ACK/ and/or Assert ATN/ are 1, the corresponding bits are reset to 0 in the **SOCL** register. Do not use this instruction in target mode.

## Initiator Mode

*Figure 12. Initiator Mode Instruction Description for I/O Instructions*

| OPC2 | OPC1 | OPC0 | 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 |
| 1 | 0 | 1 | Reserved - An Illegal Instruction Interrupt will occur |
| 1 | 1 | 0 | Reserved - An Illegal Instruction Interrupt will occur |
| 1 | 1 | 1 | Reserved - An Illegal Instruction Interrupt will occur |

### SELECT Instruction

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

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 nsec), after the SIOP has received a Disconnect Message or a Command Complete Message.

### WAIT RESELECT Instruction

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

1) 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

1) 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 10)

To enable the 53C700 as a target device set bit 10 to 1. This sets bit 0 of the **SCNTL0** register to 1. The 53C700 remains in target device mode until this bit or bit 0 in the **SCNTL0** register is reset to 0.

### Assert ACK/ (bit 6) and 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 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.

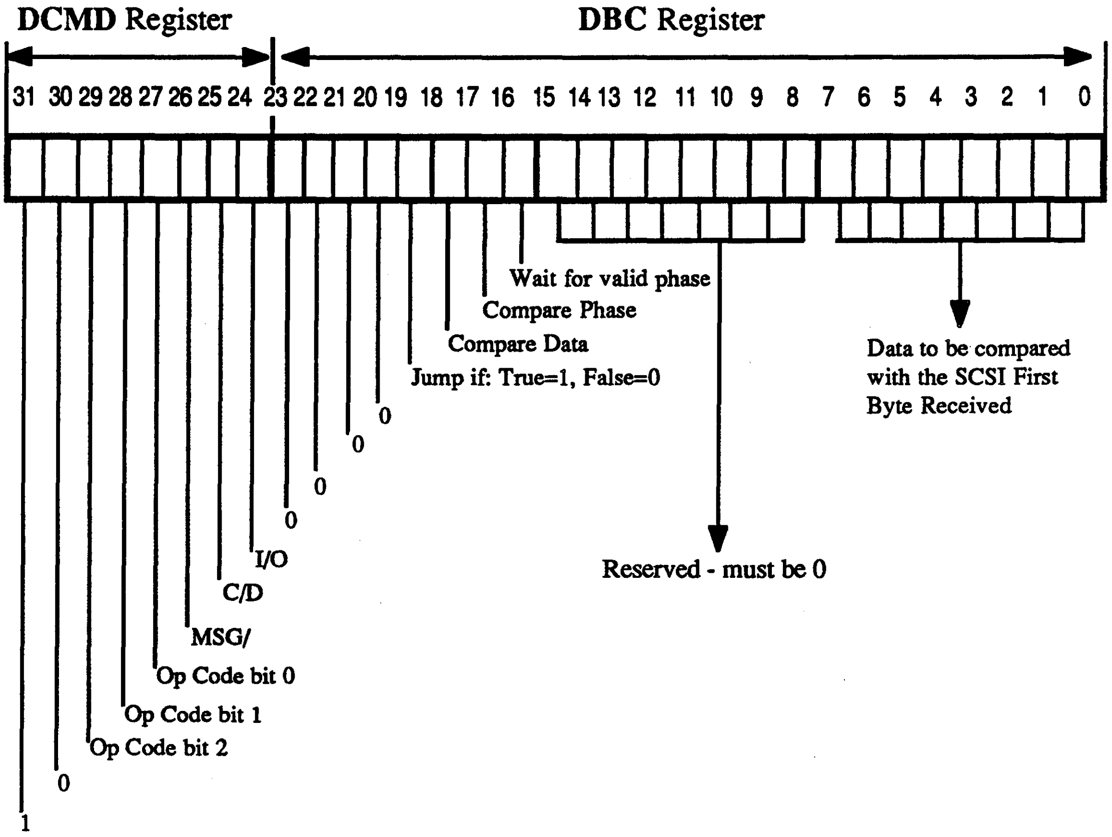
ACK/ and ATN/ are not asserted on the SCSI bus unless the SIOP is operating as an initiator or the SCSI Loopback Enable bit is 1 in the **CTEST4** register.

### Jump Address Field

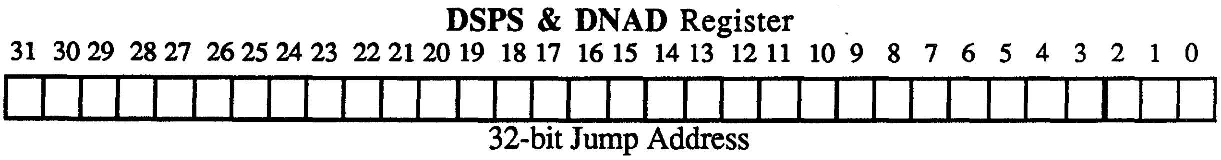
This thirty-two bit field specifies the address of the instruction to fetch when the SIOP encounters a jump condition. The SIOP fetches instructions from the address pointed to by this field whenever the SIOP encounters a SCSI condition that is different from the condition specified in the instruction.

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.

## 5.3 Transfer Control Instructions



**First 32-bit word of the Transfer Control instructions**



**Second 32-bit word of the Transfer Control instructions**

*Figure 7. Transfer Control Instruction Register*

*Note*

*In future generations of the 53C700 family, the second 32-bit word of the Transfer Control Instruction will be loaded into the DSPS only.*

## Op Code 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 initiator or target mode.

*Table 13. Transfer Control Instruction Op Code Field Definitions*

| OPC2 | OPC1 | OPC0 | 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 |
| 1 | 0 | 0 | Reserved - An Illegal Instruction Interrupt will occur |
| 1 | 0 | 1 | Reserved - An Illegal Instruction Interrupt will occur |
| 1 | 1 | 0 | Reserved - An Illegal Instruction Interrupt will occur |
| 1 | 1 | 1 | Reserved - An Illegal Instruction Interrupt will occur |

## JUMP Instruction (bits 29, 28, 27)

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

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

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 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

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 long word 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.

*Table 14. Phase Field Definitions for SCSI during Transfer Control Instructions*

| 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 standardization |
| 1 | 0 | 1 | Reserved for future standardization |
| 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 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 6
# Functional Description

## 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. It 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.

## Normal SCRIPT execution

To start a SCRIPT, write the SCRIPT address to the **DSP** Register. Then wait for an interrupt or poll the ISTAT register.

## Single-step SCRIPT execution

To execute single-step mode (one instruction at a time) set bit 4 to 1 in the **DCNTL** Register. To start a SCRIPT, write a SCRIPT 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 SCRIPT.

## 6.1 Basic Programming Steps to start SCRIPT Execution

The following list gives the programming steps for initializing the 53C700 to start fetch and execution of SCRIPT instructions.

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

2. Program SCNTLO (00h)

| Bits | Description |
|---|---|
| 0 | Target Mode |
| 1 | Assert ATN/ on Parity Error |
| 2 | Enable Parity Generation |
| 3 | Enable Parity Checking |

3. Program SCNTL1 (01h)

| Bits | Description |
|---|---|
| 5 | Enable selection and reselection |

4. Program SIEN (03h)

| Bits | Description |
|---|---|
| 0 | Enable Parity Error interrupt |
| 1 | Enable SCSI RST/ Received interrupt |
| 2 | Enable Unexpected Disconnect interrupt |
| 3 | Enable SCSI Gross Error interrupt |
| 4 | Enable Selected or reselected interrupt |
| 5 | Enable selection or reselection timeout interrupt |
| 6 | Enable function complete interrupt |
| 7 | Enable phase mismatch or ATN/ Active interrupt |

5. Program SCID (04h)

| Bits | Description |
|---|---|
| 7-0 | Chip's ID |

6. Program SXFER (05h)

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

7. Program DMODE (34h)

| Bits | Description |
|---|---|
| 2 | Fixed address mode |
| 3 | I/O or memory mapped |
| 4 | 286 mode |
| 5 | Bus width 16 |
| 7-6 | Host burst length |

8. Program DIEN (39h)

| Bits | Description |
|---|---|
| 0 | Enable Illegal Instruction interrupt |
| 1 | Enable Watchdog Timeout interrupt |
| 2 | Enable Script Interrupt Instruction Received interrupt |
| 3 | Enable Script Pipeline/Step interrupt |
| 4 | Enable Aborted interrupt |

9. Program DCNTL (3Bh)

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

10. Program DSP (2Fh - 2Ch)

| Bits | Description |
|---|---|
| 31-0 | Start address of Script |

11. Program DCNTL (3Bh), if the Single-Step Mode bit was set previously.

| Bits | Description |
|---|---|
| 2 | Start DMA operation |
| 4 | Single-step mode |

## 6.2 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) Write the Loopback Enable bit in the **CTEST4** register to 1.

2) Set-up the desired arbitration mode as defined in the **SCNTL0** register.

3) Write the Start Sequence bit to 1 in the **SCNTL0** register.

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, write the BSY/ bit, bit 5 of 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, write 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 write the REQ/ bit to 1 in the **SOCL** register. To accommodate the 400 nsec Bus Settle Delay, assert REQ/ after asserting the phase signals,

10) The initiator role can be implemented by single stepping SCSI SCRIPTS™ and the 53C700 can loopback as a target or vice versa.

## 6.3 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 - 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 - 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 - 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 - 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* - Bit in **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 - 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 - 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 - 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 - 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 - Bit 3 in the **CTEST2** register

   This status bit is 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 - 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 - 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.

## Parity Control

*Table 15. Parity Control Signals and Descriptions*

| EPG (parity generation) | EPC (parity checking) | ASEP (even SCSI parity) | EVP (even host parity) | Description |
|---|---|---|---|---|
| 0 | 0 | x | 0 | Parity pass through (DP3-DP0).<br>No parity checking.<br>Odd parity passed through the chip (DP3-DP0). |
| 0 | 0 | x | 1 | Parity pass through (DP3-DP0).<br>No parity checking.<br>Even parity asserted to host bus when receiving from SCSI (done by inverting SCSI parity).<br>Odd parity asserted on SCSI bus when sending to SCSI (inverts host bus parity). |
| 0 | 1 | 0 | 0 | Parity pass through (DP3-DP0).<br>Parity checking (always checks odd SCSI parity).<br>Odd parity throughout chip. |
| 0 | 1 | 0 | 1 | Parity pass through (DP3-DP0).<br>Parity checking (always checks odd SCSI parity).<br>Odd SCSI parity asserted on SCSI bus when sending data (inverts host bus parity).<br>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).<br>Parity checking (always checks odd SCSI parity).<br>Asserts even parity on SCSI bus when sending data to SCSI bus (inverts parity from host bus). |

*Table 15. Parity Control Signals and Descriptions (cont)*

| EPG (parity generation) | EPC (parity checking) | ASEP (even SCSI parity) | EVP (even host parity) | Description |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | Parity pass through (DP3-DP0).<br>Parity checking (assuming odd parity received from host bus, this configuration will always generate a parity error when sending data to the SCSI bus).<br>Assert even parity on SCSI bus when sending data to SCSI bus (inverts parity from host bus).<br>Assert even parity on host bus when receivng data from SCSI bus (inverts parity from SCSI bus). |
| 1 | 0 | 0 | x | Parity generation (DP3-DP0 ignored).<br>No parity checking.<br>Odd parity generated on SCSI bus. |
| 1 | 0 | 1 | x | Parity generation (DP3-DP0 ignored).<br>No parity checking.<br>Even parity generated on SCSI bus. |
| 1 | 1 | 0 | x | Parity generation (DP3-DP0 ignored).<br>Parity checking.<br>Odd parity generated and checked on SCSI bus. |
| 1 | 1 | 1 | x | Parity generation (DP3-DP0 ignored).<br>Parity checking.<br>Even parity generated onto SCSI bus.<br>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* |
| *ASEP = Assert SCSI Even Parity* | *(**SCNTL1** register, bit 2)* | *x = don't care* |
| *EVP = Assert Even Host Parity* | *(CTEST7 register, bit 2)* | |

### Parity Errors and Interrupts

This table describes the options available when an parity error occurs. This table ONLY applies to the case where the Enable Parity Checking bit is 1 (**SCNTL0** register, bit 2).

*Table 16. Parity Errors & 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)*

## 6.4 Diagnostics

### DMA FIFO Test

The DMA FIFO is more complex than the SCSI FIFO. The DMA FIFO is a 36 X 8 bit 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.

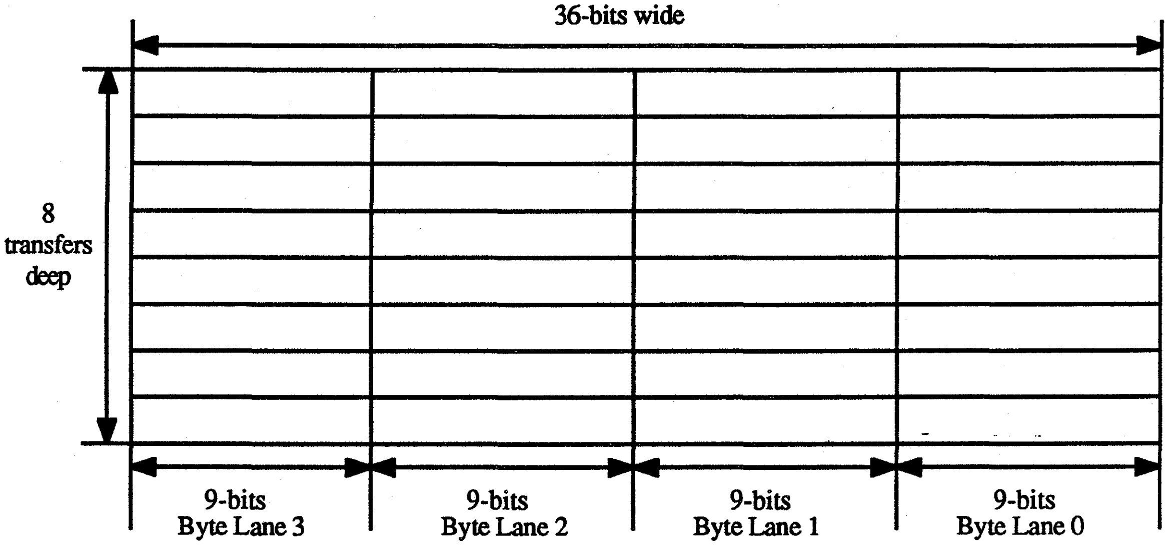


*Figure 8. DMA FIFO Sections*

To write data into the DMA FIFO, load the data 9 bits per instruction. Data is written to the top of the FIFO. DATA 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 below.

*Table 17. Byte Lane Descriptions*

| 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*

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);
```

## SCSI FIFO Test

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);
```

## 6.5 Abort Operation

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

If DP3_ABRT/ is used to perform a hardware abort, parity generation must be disabled through the **SCNTLO** Register, bit 2.

Read the **DSTAT** Register to clear an abort interrupt. If the **ISTAT** Register, bit 7 is used for a software abort, 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.

<u>SCSI asynchronous or synchronous receive</u>

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

# of bytes = DBC

<u>SCSI asynchronous send</u>

# 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.

### 53C700 Software Abort Example

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 53C700 is connected.

1. If connected, exit and wait for the current I/O to complete or schedule it for the next I/O.

If **ISTAT** Register, bit 3 shows that the 53C700 is not connected:

1. Issue an abort by asserting the **ISTAT** Register, bit 7.
2. Wait for the Abort interrupt.
3. Check for connection again (**ISTAT** Register, bit 3). As an initiator, the 53C700 may have been reselected during an abort. As a target, the 53C700 may have been selected during an abort.

If connected, restart the Wait Reselect or Wait Select instruction to continue the reselect or select. Exit and wait for the I/O to complete or schedule an I/O.

If it is not connected start a SCRIPT for a new I/O.

## 6.6 53C700 Disconnect

If there is an unexpected disconnect during a block move, use the following steps to save the state of 53C700 at disconnect.

1. Read the address of the current SCRIPT 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 or synchronous receive

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

# of bytes = DBC

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 the SCSI and DMA FIFOs (**DFIFO** register, bit 6).
4. Update the SCRIPT to handle reselect by the disconnecting device.

### Being Selected or Reselected when trying to perform Selection

#### Initiator Mode

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 initiator mode tries to select a target. If the 53C700 is executing

```
SELECT id, alt_addr
```

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

If the 53C700 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 53C700 has been reselected, the SCRIPT can then compare for id and jump to the corresponding device I/O SCRIPT.

If the 53C700 has been selected, the jump to alt_addr1 is then taken where a target SCRIPT can be executed.

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

### Being Selected or Reselected when trying to perform Reselection

#### Target Mode

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 53C700 is executing

```
RESELECT id, alt_addr
```

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

If the 53C700 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 53C700 has been selected, the IDs can be compared and then jump to the corresponding device I/O SCRIPT.

If the 53C700 is reselected, the jump to alt_addr1 is taken where an initiator SCRIPT can be executed.

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

## 6.7 80386 Interface

### 53C700 SIOP Data Paths

The data path through the SIOP are dependent on two things. First is data being moved in or out of the chip, second is SCSI data being sent asynchronously or synchronously.

The diagrams below show how data is moved to/from the SCSI bus in each of the different modes.

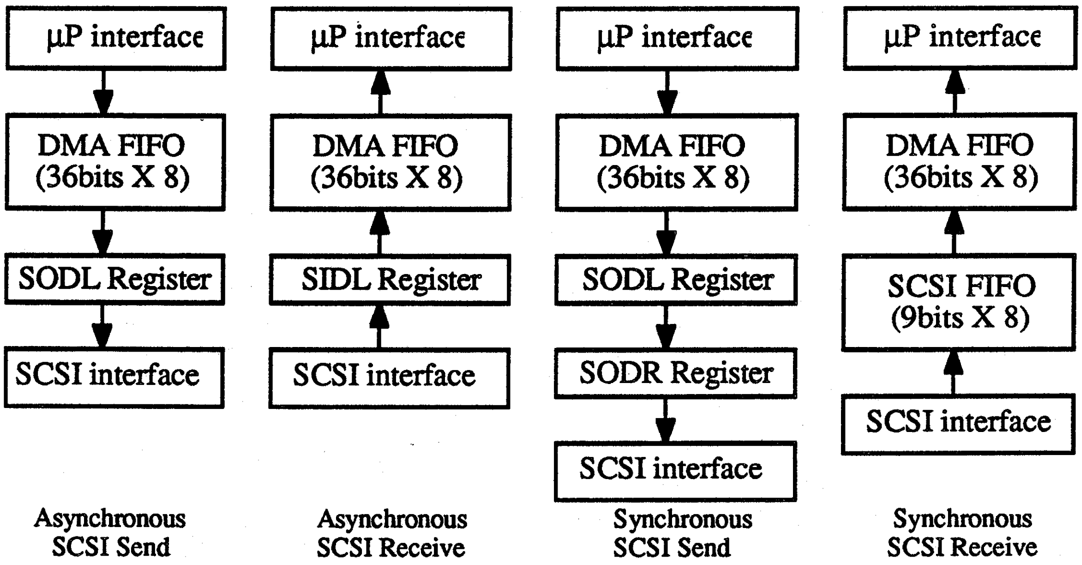


*Figure 9. 53C700 SIOP Data Paths*

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

### 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 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.

3) Read the SSTAT1 register and examine bit 6 to determine if any bytes are remaining in the SODR register. If bit 6 equals 1, then there is a byte in the **SODR** register and 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 examining bit 7 to determine if any bytes are left in the **SIDL** register. If bit 7 equals 1, then there is a byte in the **SIDL** register and 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 examine 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.

### Transfers 16- or 32-bit bus

The SIOP can transfer data 16-bits or 32-bit 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 in 16-bits per instruction fetch or 32-bits per instruction fetch.

*Table 18. Block Move Transfer Bit Descriptions*

| BW16 | 286M | Description |
|---|---|---|
| 0 | 0 | 32-Bit Data Transfers, SIOP asserts and expects 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 a 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 19. 80286 Mode 16-Bit Data Transfers (DMODE register, bit 4 = 1)*

| Address<br>(DNAD register) | | BHE1 | 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 |

| yy<br>15 8 | xx<br>7 0 | address (word aligned) |
|---|---|---|
| BHE/ | A0 | |

*Figure 10. Address, Data, and Byte Enables for 16-Bit Bus*

## 80386 Mode 16-Bit Data Transfers

*Table 20. 80386 Mode 16-Bit Data Transfers (DMODE register bit 4=0, bit 5=1)*

| | First Cycle | | | | | | Second Cycle | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| (DNAD register)<br>Address | 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 |

| zz<br>31 24<br>aligned) | yy<br>23 16 | xx<br>15 8 | ww<br>6 0 | address<br>(long word |
|---|---|---|---|---|
| BE3/ | BE2/ | BE1/ | BE0/ | |

*Figure 11. Address, Data Bus and Byte Enables for 32-Bit Bus*

*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.

### 80386 Mode

*Table 21. 32-Bit Data Transfers in 80386 Mode*

| | BE | BE | BE | BE |
|---|---|---|---|---|
| | 3/ | 2/ | 1/ | 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 address*

### Instruction Fetch Operation

Write the address containing the first SCSI SCRIPT™ to the **DSP** register to start the SIOP instruction fetch process. Even after the first SCSI SCRIPT™ address is written to the **DSP** register, the SIOP continues to fetch and execute its instructions by reading them from system memory.

These SCSI SCRIPTS™ are not required to reside in system memory and by decoding a certain address space, they could reside in a PROM. Store SCSI SCRIPTS™ only in a memory-mapped address, 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.

Write the SCRIPTS loaded in 16-bit Mode bit in the **DCNTL** register to 1, if SCSI SCRIPTS™ are to be loaded 16-bits per transfer. The DC/ control signal can be driven high or low depending on the status of the DC/ 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) will reside in the cache and data information (DC/ high) does not.

### 53C700 SIOP Bus Master Data Transfers

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 22. 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/O address 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* |

## HOLD - HLDA schemes

The SIOP gains control of the system bus when it needs to perform one of two operations: an instruction fetch, or a data transfer specified by a Block Move instruction. The SIOP requests control of the 80386 bus by asserting the HOLD output signal. When the HLDAI signal is driven active by the 80386 (or by some arbitration logic), then the SIOP assumes bus mastership and starts driving the system address and control signals.

The SIOP also allows another bus master device to request the bus through the use of a daisy-chaining technique. The SIOP has a HLDREQ input which can be used by another device to request the system bus.

The SIOP also has a HLDAO output signal which indicates when control of the bus is granted to the device that asserted the HLDREQ signal. The diagram below is an example of the daisy-chaining technique.

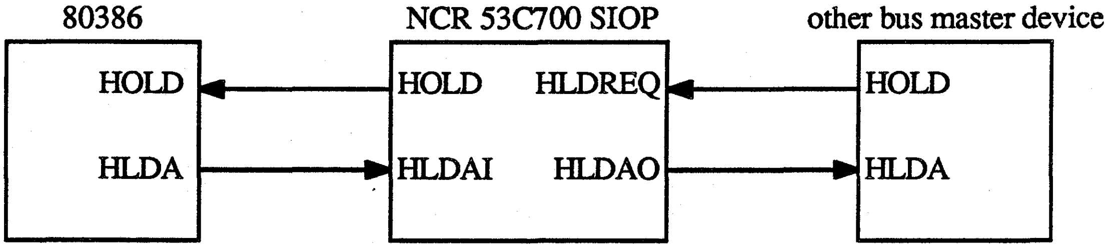


*Figure 12. HOLD - HLDA Daisy-chaining Scheme*

If the SIOP and the other bus master device both request the bus simultaneously, then the highest priority would be given to the SIOP. If the other bus master device requested the bus and the SIOP did not need access to the bus, then the HLDA signal received from the 80386 would then be passed through to the other bus master device.

A central arbiter is another method to allow bus master devices to gain access to the system bus. The arbiter receives bus requests from various bus master devices and prioritizes access according to some priority scheme. In this scheme, the HLDREQ and HLDAO signals are not used and only the SIOP uses HOLD and HLDAI. If not used, tie HLDREQ directly to VSS or through a pull-down resistor to VSS. Figure 13 is an example of the central arbiter technique.

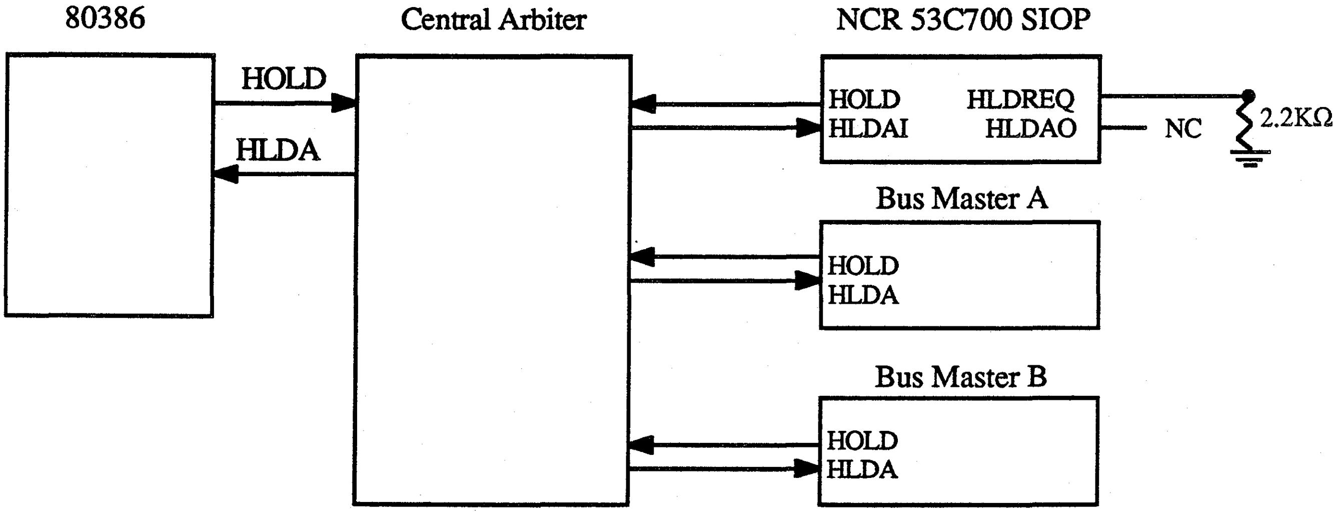


*Figure 13. Central Arbiter scheme*

## 6.8 SCSI Interface

### Single-Ended Mode

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.

### Differential Mode

In differential mode, the SDIR7-0, SDIRP, IGS, TGS, RSTDIR, BSYDIR, and SELDIR signals control the direction of external differential-pair transceivers. The recommended differential-pair transceivers are the 75176 or the DS3695.

### Terminator Network

The terminators provide the biasing needed to "pull" the inactive signal to an appropriate inactive voltage level ~ 3.0V. 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 active, then the impedance and inactive SCSI voltage levels may not be correct.

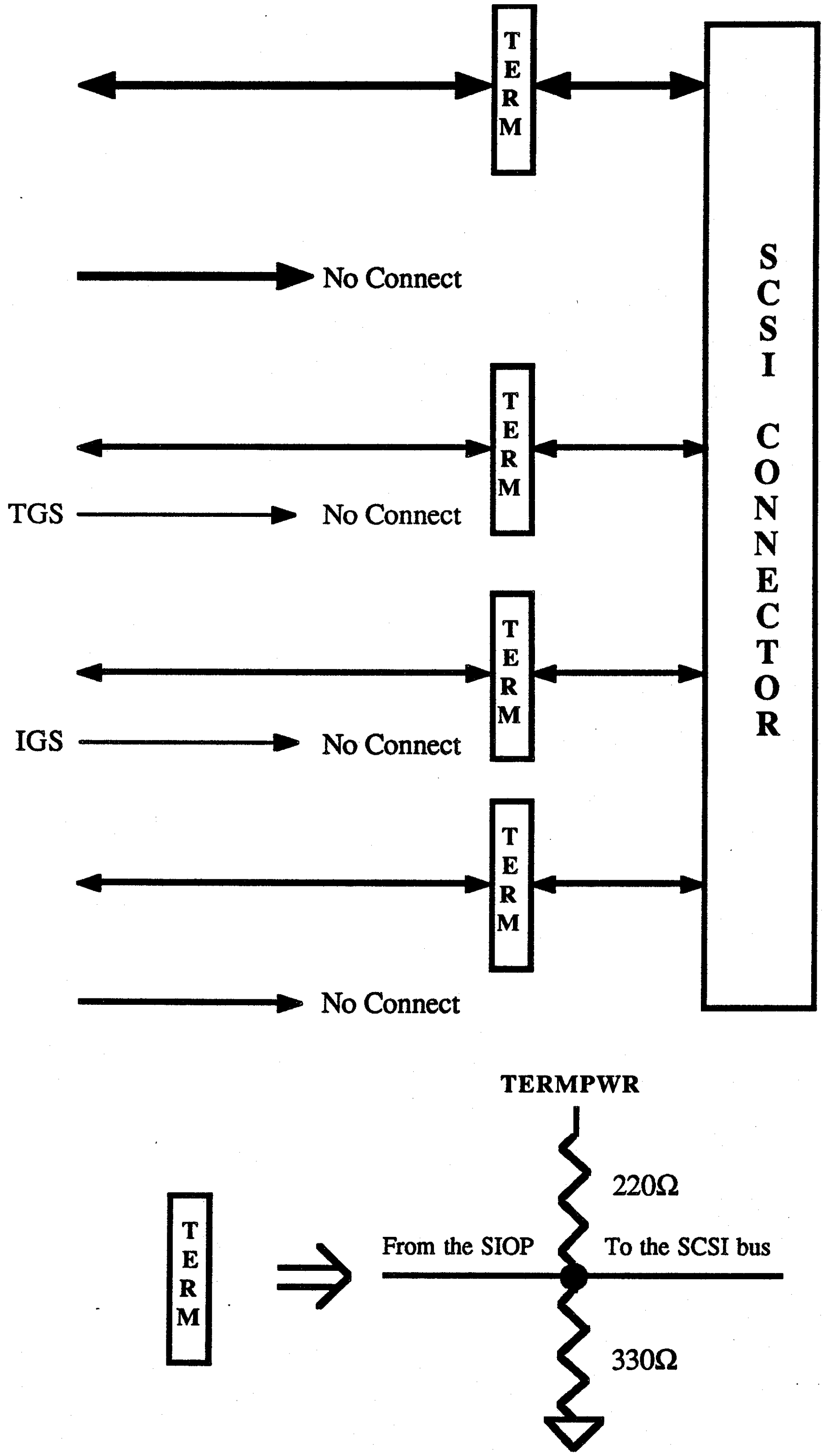


*Figure 14. SCSI Connector*

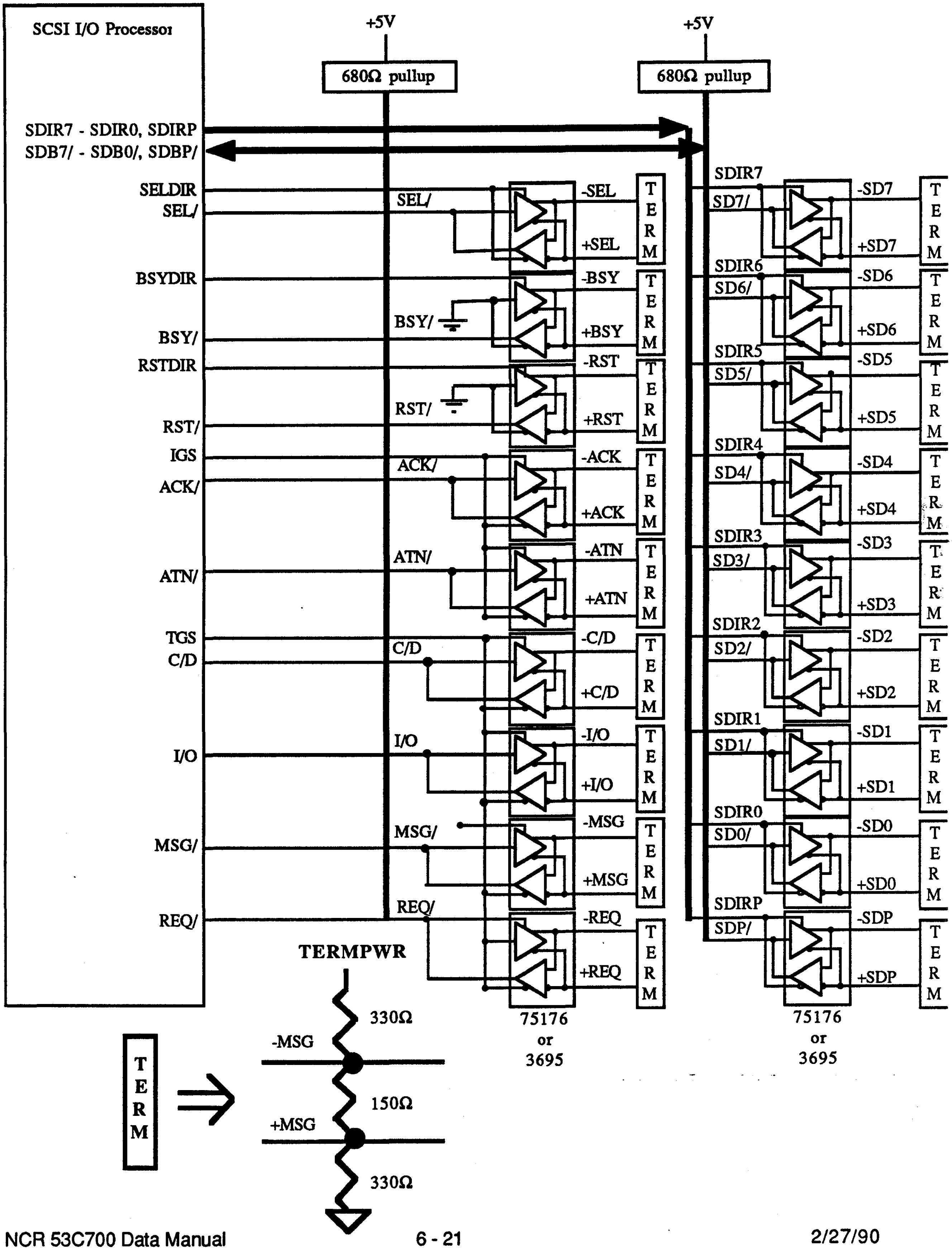

SCSI I/O Processor
+5V
680Ω pullup
+5V
680Ω pullup
SDIR7 - SDIR0, SDIRP
SDB7/ - SDB0/, SDBP/
SELDIR
SEL/
BSYDIR
BSY/
RSTDIR
RST/
IGS
ACK/
ATN/
TGS
C/D
I/O
MSG/
REQ/
SEL/
BSY/
RST/
ACK/
ATN/
C/D
I/O
MSG/
REQ/
-SEL
+SEL
-BSY
+BSY
-RST
+RST
-ACK
+ACK
-ATN
+ATN
-C/D
+C/D
-I/O
+I/O
-MSG
+MSG
-REQ
+REQ
TERM
SDIR7
SD7/
SDIR6
SD6/
SDIR5
SD5/
SDIR4
SD4/
SDIR3
SD3/
SDIR2
SD2/
SDIR1
SD1/
SDIR0
SD0/
SDIRP
SDP/
-SD7
+SD7
-SD6
+SD6
-SD5
+SD5
-SD4
+SD4
-SD3
+SD3
-SD2
+SD2
-SD1
+SD1
-SD0
+SD0
-SDP
+SDP
75176
or
3695
75176
or
3695
TERMPWR
330Ω
-MSG
150Ω
+MSG
330Ω
TERM

*Table 23. D.C. Characteristics*

**Absolute Maximum Stress Ratings**

| SYMBOL | PARAMETER | MIN | MAX | UNIT |
|---|---|---|---|---|
| Tstg | 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 | | 10K | V |

* Test using the human body model--100 pF at 1.5 KΩ

**Operating Conditions**

| SYMBOL | PARAMETER | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|
| $V_{DD}$ | Supply Voltage | 4.75 | | 5.25 | V |
| $I_{DD}$ | Dynamic Supply Current | | 35 | 50 | mA |
| $I_{DD}$ | Static Supply Current | | 1 | 5 | mA |
| Ta | Operating Free-Air | 0 | | 70 | °C |

**SCSI Signals - SD7/-SD0/, SDP/, REQ/, MSG/, I/O, C/D, ATN/, ACK/, BSY/, SEL/, RST/**

| SYMBOL | PARAMETER | MIN | MAX | UNIT | CONDITIONS |
|---|---|---|---|---|---|
| $V_{IH}$ | Input High Voltage | 2.0 | VDD + 0.5 | V | |
| $V_{IL}$ | Input Low Voltage | VSS - 0.5 | 0.8 | V | |
| $V_{OL}$ | Output Low Voltage | VSS | 0.4 | V | IOL = 48 mA |
| $V_{HYS}$ | Hysteresis | 200 | | mV | |
| $I_{IN}$ | Input Leakage Current | -10 | 10 | µA | |
| $I_{OZ}$ | Output Leakage Current | -10 | 10 | µA | |
| $C_{IN}$ | Input Capacitance | | 10.6 | pF | |

**SCSI Direction Control Signals - SDIR7-SDIR0, SDIRP, BSYDIR, SELDIR, RSTDIR, TGS, IGS**

| SYMBOL | PARAMETER | MIN | MAX | UNIT | CONDITIONS |
|---|---|---|---|---|---|
| $V_{OH}$ | Output High Voltage | 2.4 | VDD | V | IOH = -4 mA |
| $V_{OL}$ | Output Low Voltage | VSS | 0.4 | V | IOL = 4 mA |
| $I_{OH}$ | Output High Current | -2.0 | | mA | VOH = VDD - 0.5 V |
| $I_{OL}$ | Output Low Current | 4.0 | | mA | VOL = 0.4 V |
| $I_{OZ}$ | Output Leakage Current | -10 | 10 | µA | |
| $C_{IN}$ | Input Capacitance | | 1.059 | pF | |

**Input Signals - HCS/, HLDAI, NA/, RESET, HLDREQ, READYI/**

| SYMBOL | PARAMETER | MIN | MAX | UNIT | 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 | | 6.9 | pF | |

**CLK Input Signal**

| SYMBOL | PARAMETER | MIN | MAX | UNIT | 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 | | 6.9 | pF | |

**Output Signals - HOLD, HLDAO, MIO/, DC/, READYO/**

| SYMBOL | PARAMETER | MIN | MAX | UNIT | CONDITIONS |
|---|---|---|---|---|---|
| $V_{OH}$ | Output High Voltage | 2.4 | VDD | V | IOH = -8 mA |
| $V_{OL}$ | Output Low Voltage | VSS | 0.4 | V | IOL = 8 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 | |
| $C_{IN}$ | Input Capacitance | | 6.9 | pF | |

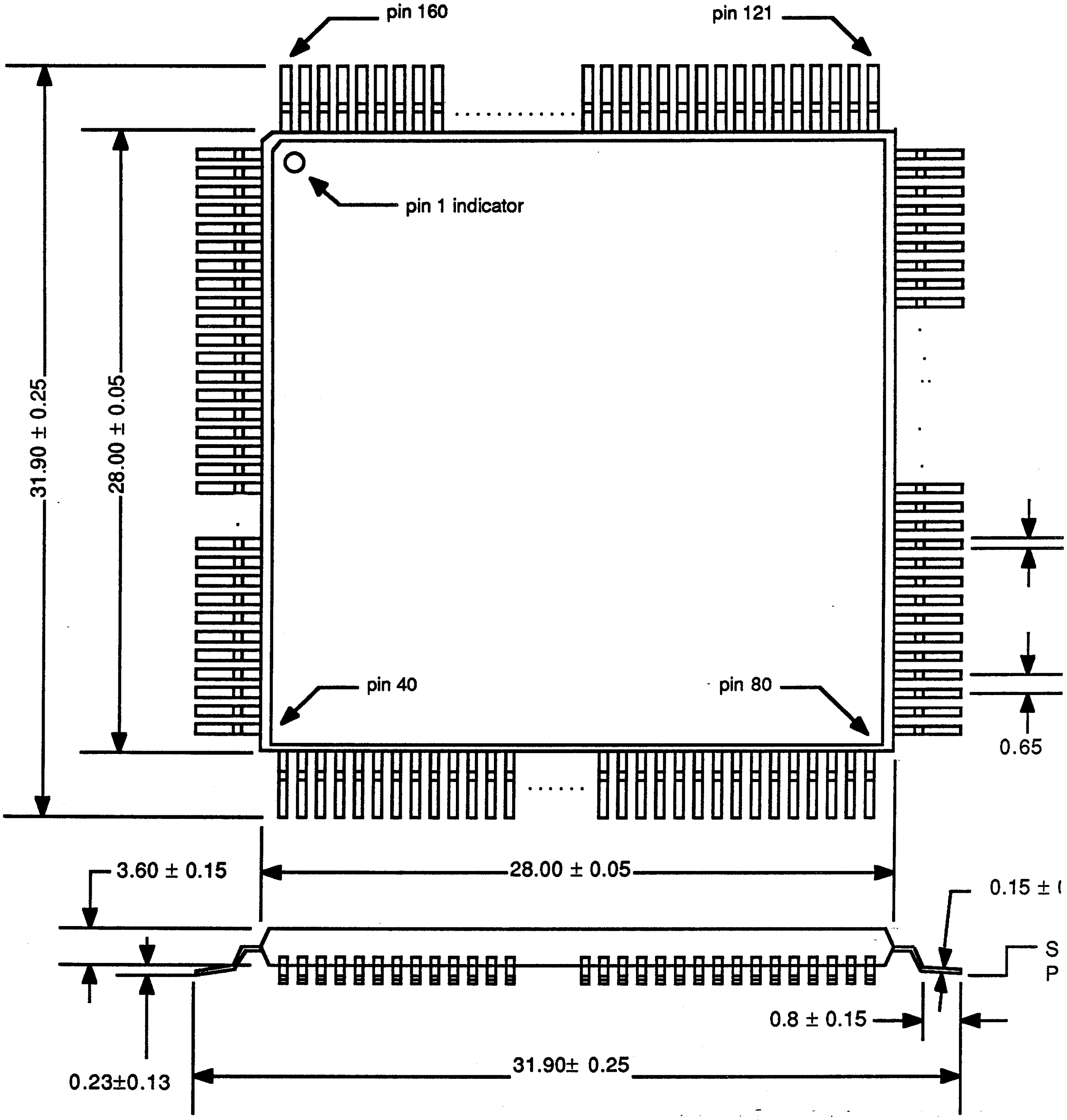
**Output Signal - IRQ/**

| SYMBOL | PARAMETER | MIN | MAX | UNIT | CONDITIONS |
|---|---|---|---|---|---|
| $V_{OL}$ | Output Low Voltage | VSS | 0.4 | V | IOL = 8 mA |
| $I_{OL}$ | Output Low Current | 8.0 | | mA | VOL = 0.4 V |
| $I_{OZ}$ | Output Leakage Current | -400 | 10 | µA | |
| $C_{IN}$ | Input Capacitance | | 52 | pF | |

**Bi-directional Signals - R/W/, A05, A31-A2, D31-D0, DP3_ABRT/, DP2-DP0, BE0/_A0, BE1/_A1, BE2/_BHE/, BE3/**

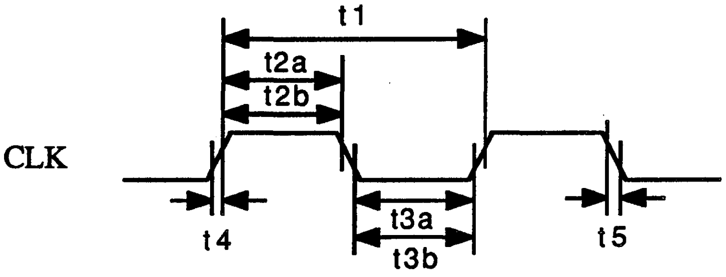
| SYMBOL | PARAMETER | MIN | MAX | UNIT | CONDITIONS |
|---|---|---|---|---|---|
| $V_{IH}$ | Input High Voltage | 2.0 | VDD + 0.5 | V | |
| $V_{IL}$ | Input Low Voltage | VSS - 0.5 | 0.8 | V | |
| $V_{OH}$ | Output High Voltage | 2.4 | VDD | V | IOH = -8 mA |
| $V_{OL}$ | Output Low Voltage | VSS | 0.4 | V | IOL = 8 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 | 10 | µA | |
| $I_{OZ}$ | Output Leakage Current | -10 | 10 | µA | |
| $C_{IN}$ | Input Capacitance | | 6.9 | pF | |

*Figure 15. NCR 53C700 Pinout - Mechanical Drawing*



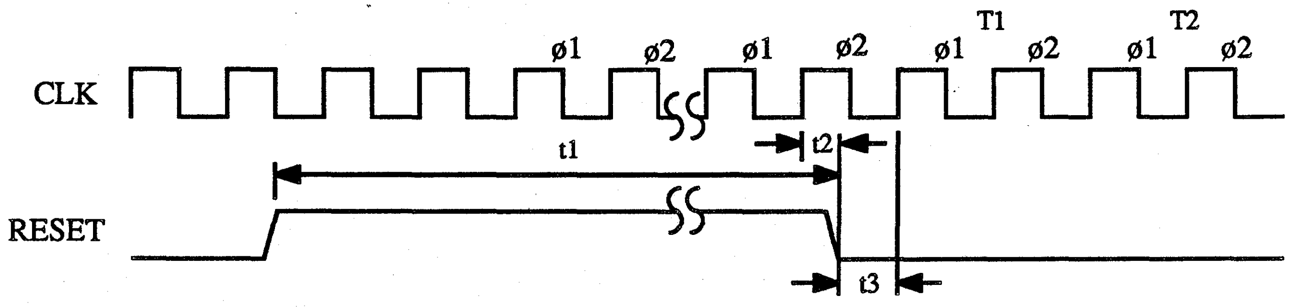
# Appendix A
# Timing

## Clock Timing



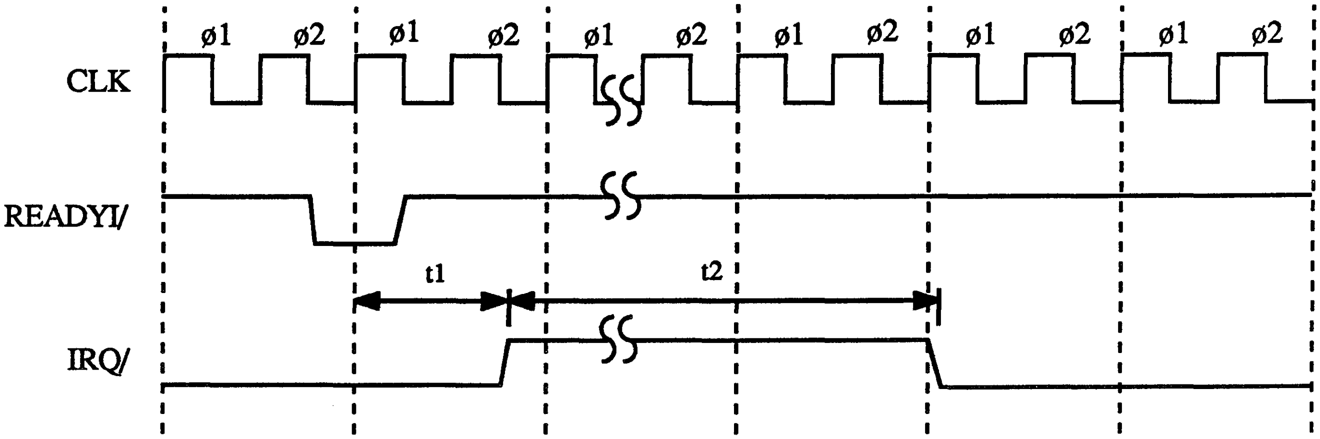
| Clock Timing | | | | |
|---|---|---|---|---|
| Symbol | Description | Min. | Max. | Units |
| t1 | CLK period | 20 | 60 | ns |
| t2a | CLK high (at 3.85V) | 4 | - | ns |
| t2b | CLK high (at 2V) | 7 | - | ns |
| t3a | CLK low (at 1.65V) | 5 | - | ns |
| t3b | CLK low (at 2V) | 7 | - | ns |
| t4 | CLK rise time (1.65V to 3.85V) | - | 7 | ns |
| t5 | CLK fall time (3.85V to 1.65V) | - | 7 | ns |

# Reset and Clock Synchronization



| Reset and Clock Synchronization | | | | |
|---|---|---|---|---|
| Symbol | Description | Min. | Max. | Units |
| t1 | Reset pulse width | 5 | - | CLK |
| t2 | Reset hold from CLK rising edge | 3 | - | ns |
| t3 | Reset inactive to CLK rising edge | 10 | - | ns |

## Interrupt Timing

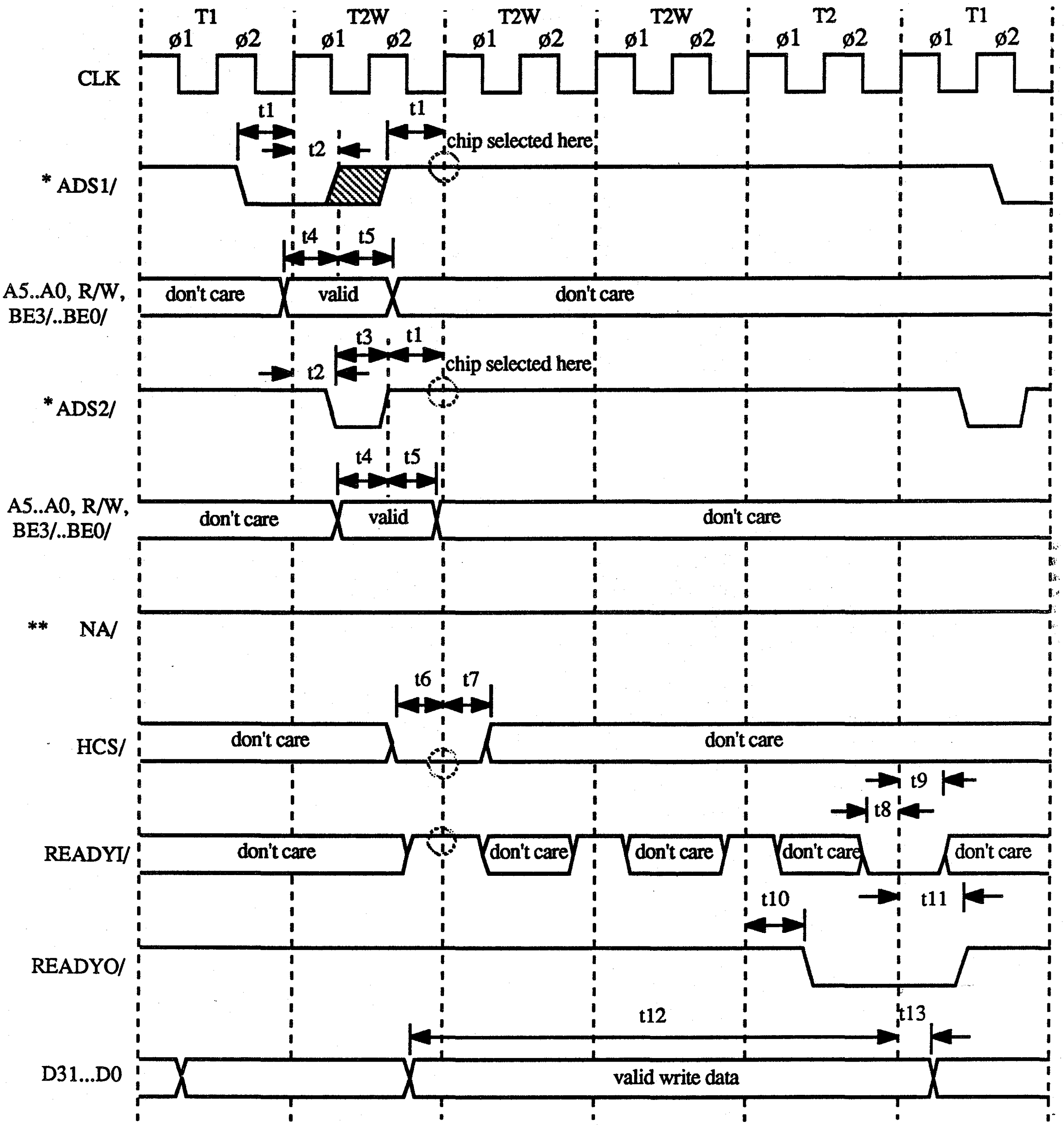


*Note:*
IRQ/ is deasserted after a read of the DSTAT or SSTAT0 register. This occurs within t1 from the rising CLK edge with READYI/ active. The READYI/ timing is shown in the Slave Mode Read Cycle. When multiple or stacked interrupts occur, IRQ/ is deasserted for a minimum of 3 CLKs.

| Interrupt Output Timing | | | | |
|---|---|---|---|---|
| Symbol | Description | Min. | Max. | Units |
| t1 | IRQ/ deasserted from CLK rising | 10 | *50 | ns |
| t2 | IRQ/ deasserted to IRQ/ asserted | 3 | - | CLK |

* This can be shortened by adding an external pull-up resistor.

# Slave Mode Write Cycle (53C700 Register Write)



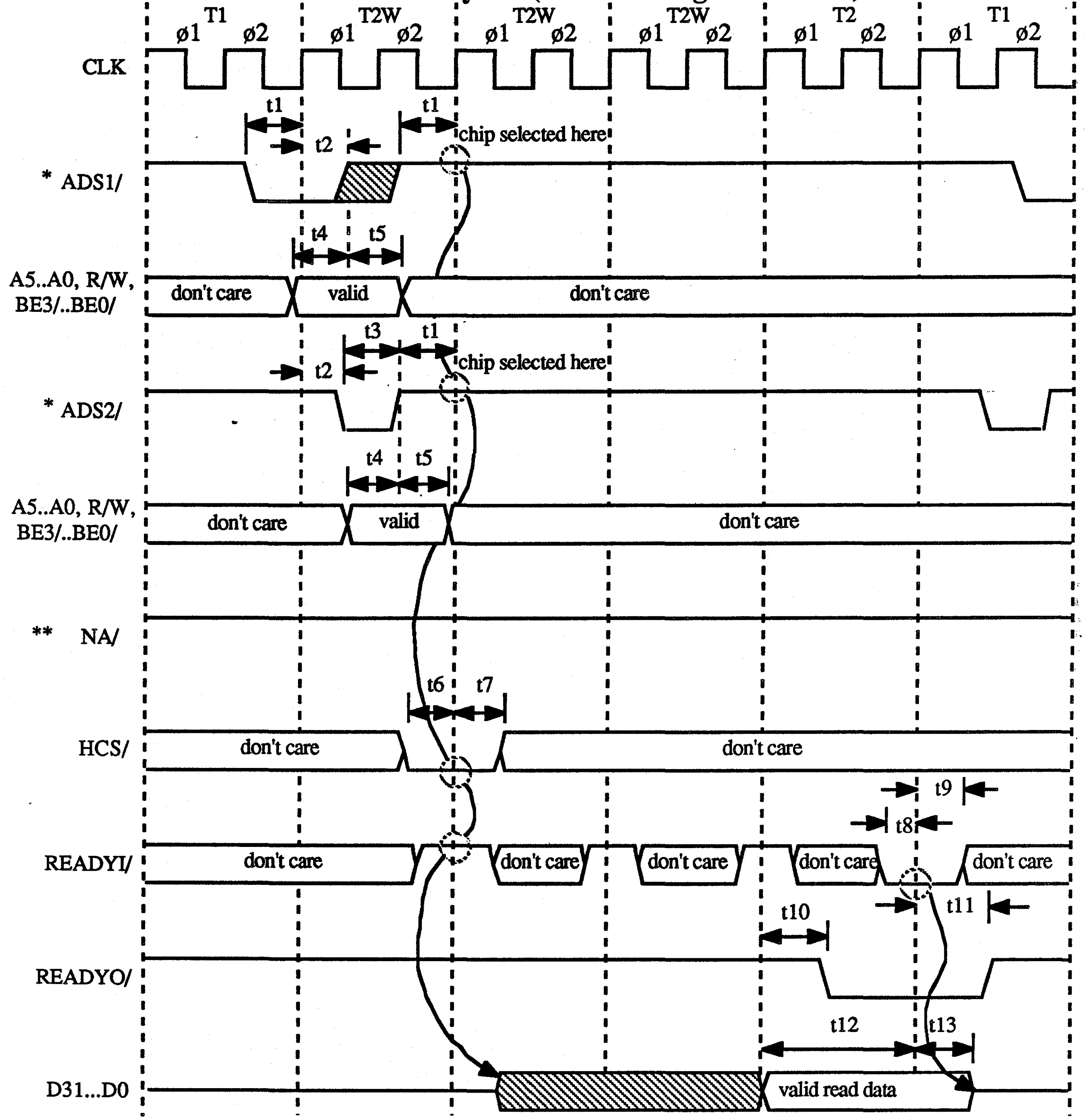
*Notes*:

* 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.

| Slave Mode Write Cycle | | | | |
|---|---|---|---|---|
| Symbol | Description | Min. | Max. | Units |
| t1 | ADS1/ setup or ADS2/ deasserted to CLK rising edge | 7 | - | ns |
| t2 | ADS1/ hold or ADS2/ asserted from CLK rising edge | 5 | - | ns |
| t3 | ADS2/ pulse width | 15 | - | ns |
| t4 | Control and address setup to ADS/ deasserted | 10 | - | ns |
| t5 | Control and address hold from ADS/ deasserted | 10 | - | ns |
| t6 | HCS/ setup to CLK rising edge | 7 | - | ns |
| t7 | HCS/ hold from CLK rising edge | 5 | - | ns |
| t8 | READYI/ setup to CLK rising edge | 9 | - | ns |
| t9 | READYI/ hold from CLK rising edge | 6 | - | ns |
| t10 | READYO/ valid from CLK rising edge | 2 | 17 | ns |
| t11 | READYO/ hold from CLK rising edge | 3 | 19 | ns |
| t12 | Data setup to CLK rising edge | 6 | - | CLK |
| t13 | Data hold from CLK rising edge | 20 | - | ns |

## Slave Mode Read Cycle (53C700 Register Read)



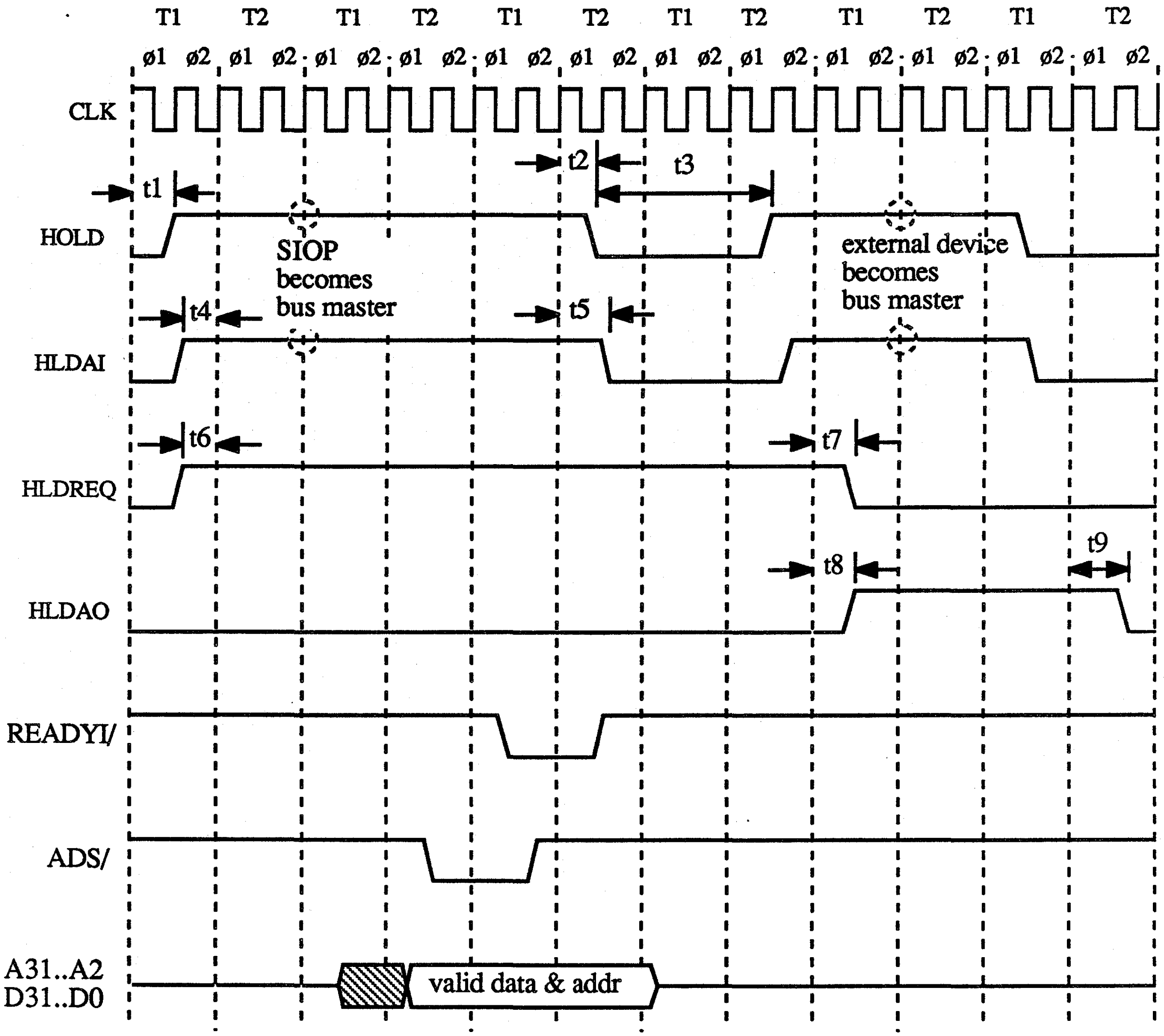
*Notes:*

* 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.

| Slave Mode Read Cycle | | | | |
|---|---|---|---|---|
| Symbol | Description | Min. | Max. | Units |
| t1 | ADS1/ setup or ADS2/ deasserted to CLK rising edge | 7 | - | ns |
| t2 | ADS1/ hold or ADS2/ asserted from CLK rising edge | 5 | - | ns |
| t3 | ADS/ pulse width | 15 | - | ns |
| t4 | Control and address setup to ADS/ deasserted | 10 | - | ns |
| t5 | Control and address hold from ADS/ deasserted | 10 | - | ns |
| t6 | HCS/ setup to CLK rising edge | 7 | - | ns |
| t7 | HCS/ hold from CLK rising edge | 5 | - | ns |
| t8 | READYI/ setup to CLK rising edge | 9 | - | ns |
| t9 | READYI/ hold from CLK rising edge | 6 | - | ns |
| t10 | READYO/ valid from CLK rising edge | 2 | 17 | ns |
| t11 | READYO/ hold from CLK rising edge | 3 | 19 | ns |
| t12 | Data setup to CLK rising edge | 2 | - | CLK |
| t13 | Data hold from CLK rising edge | 5 | 25 | ns |

## Bus Arbitration

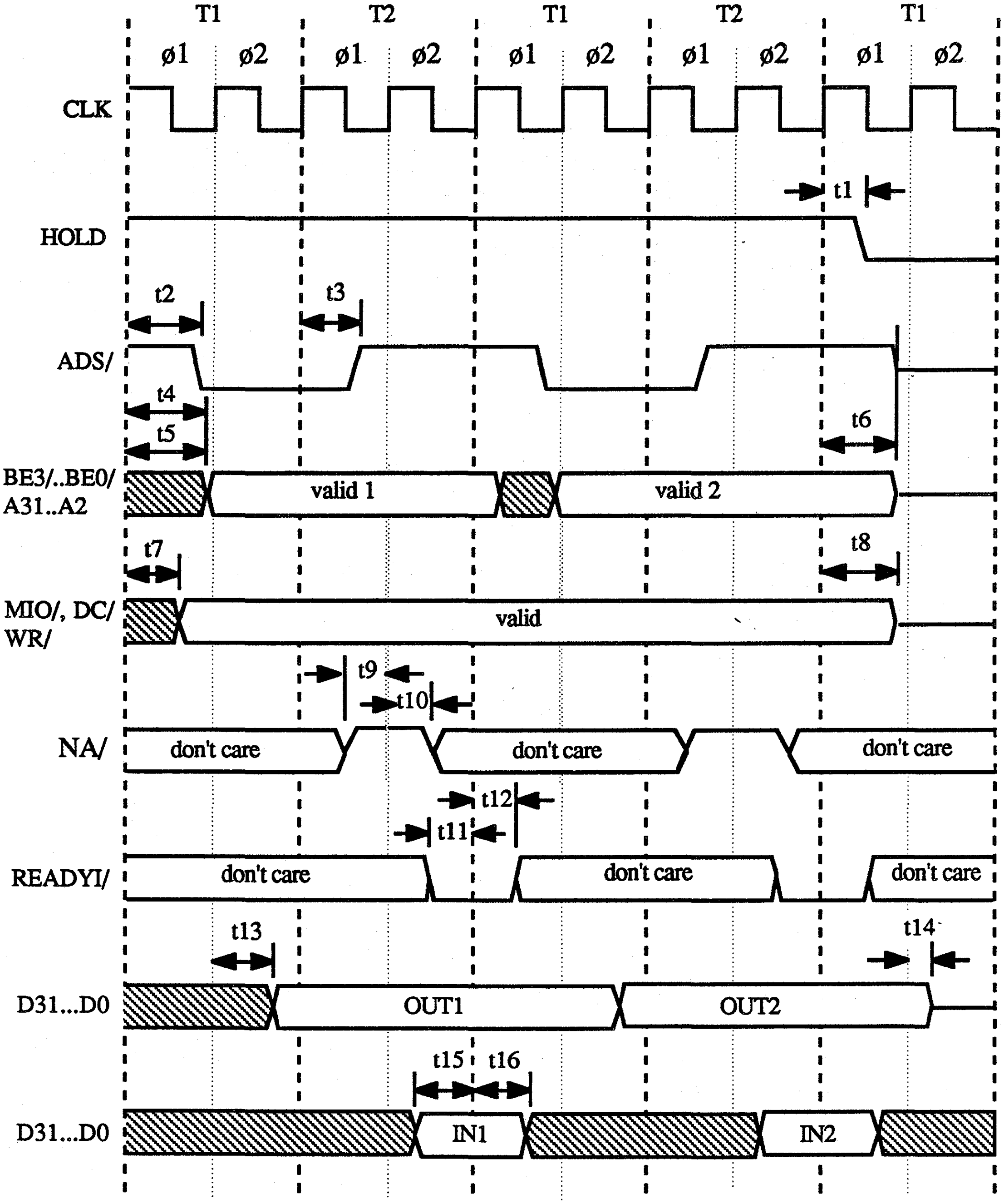


*Notes*:

The SIOP becomes bus master on the next ø1 CLK edge after HOLDAI is sampled active. HLDAI and HLDREQ are asynchronous inputs; timings given are for minimum synchronization delays. The SIOP waits 8 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 2 T states from deasserting HOLD to asserting HOLD.

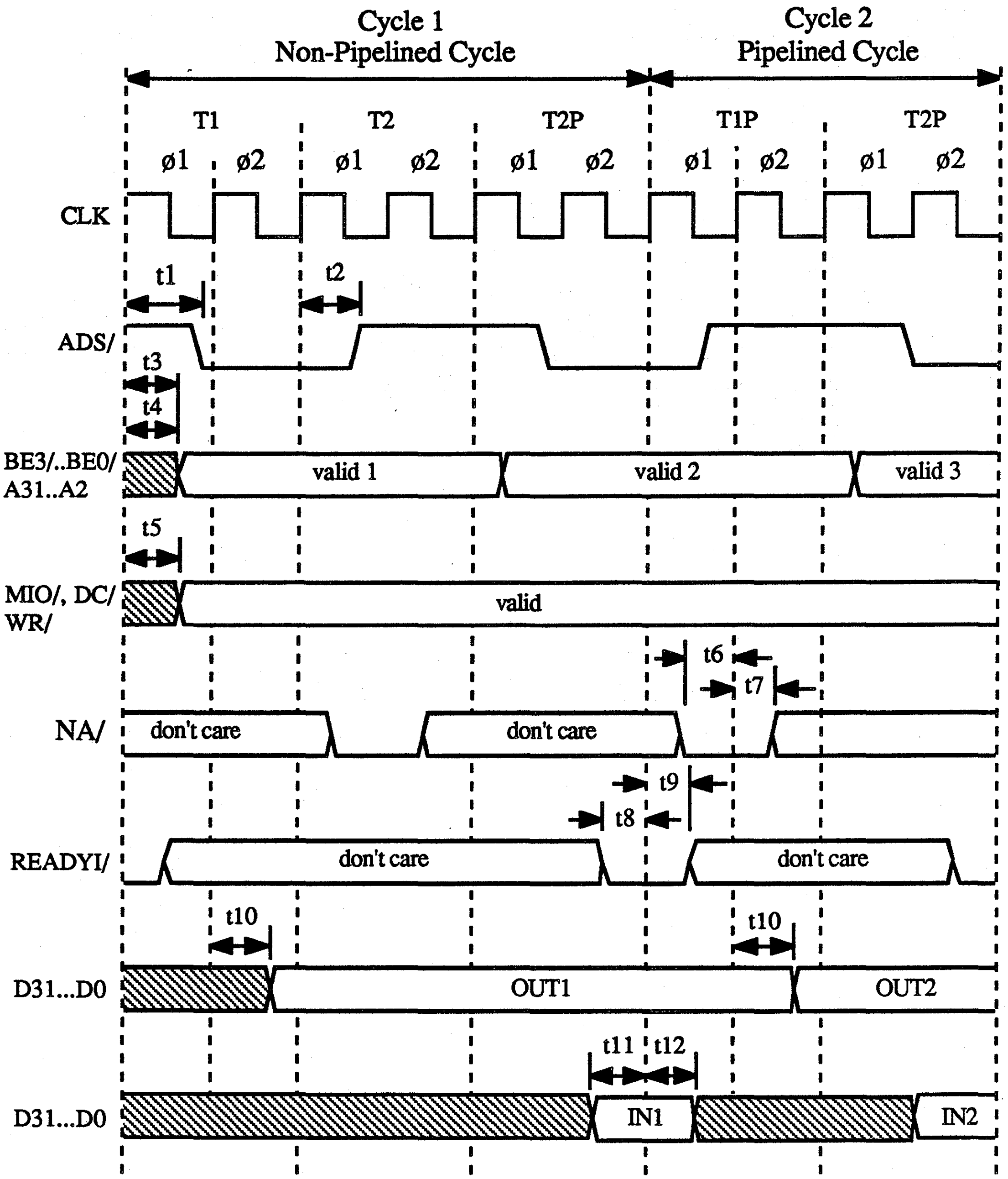
| Bus Arbitration | | | | |
|---|---|---|---|---|
| Symbol | Description | Min. | Max. | Units |
| t1 | HOLD valid from CLK rising edge | 3 | 25 | ns |
| t2 | HOLD hold from CLK rising edge | 4 | 30 | ns |
| t3 | HOLD inactive to HOLD active | 2 | - | CLK |
| t4 | HLDAI setup to CLK rising edge | 3 | - | ns |
| t5 | HLDAI hold from CLK rising edge | 3 | - | ns |
| t6 | HLDREQ setup to CLK rising edge | 3 | - | ns |
| t7 | HLDREQ hold fromCLK rising edge | 3 | - | ns |
| t8 | HLDAO active from CLK rising edge | 3 | 25 | ns |
| t9 | HLDAO inactive from CLK rising edge | 3 | 25 | ns |

# Bus Master Memory Read and Write Cycle Non-Pipelined Next Address



| Bus Master Memory Read and Write Cycle - Non-Pipelined Next Address | | | | |
|---|---|---|---|---|
| Symbol | Description | Min. | Max. | Units |
| t1 | HOLD hold from CLK rising edge | 4 | | ns |
| t2 | ADS/ valid from CLK rising edge | 4 | 21 | ns |
| t3 | ADS/ hold from CLK rising edge | 4 | 30 | ns |
| t4 | Address valid from CLK rising edge | 4 | 23 | ns |
| t5 | BE3/..BE0/ valid from CLK rising edge | 4 | 24 | ns |
| t6 | ADS/, Address & BE3/..BE0/ tristate from CLK rising edge | 4 | 30 | ns |
| t7 | MIO/, DC/, WR/ valid from CLK rising edge | - | 4 | ns |
| t8 | MIO/, DC/, WR/ tristate from CLK rising edge | 4 | 30 | ns |
| t9 | NA/ setup to CLK rising edge | 7 | - | ns |
| t10 | NA/ hold from CLK rising edge | 3 | - | ns |
| t11 | READYI/ setup to CLK rising edge | 9 | - | ns |
| t12 | READYI/ hold from CLK rising edge | 6 | - | ns |
| t13 | Write data setup to CLK rising edge | 5 | 27 | ns |
| t14 | Write data tristate from CLK rising edge | 2 | 22 | ns |
| t15 | Read data setup to CLK rising edge | 7 | - | ns |
| t16 | Read data hold from CLK rising edge | 5 | - | ns |

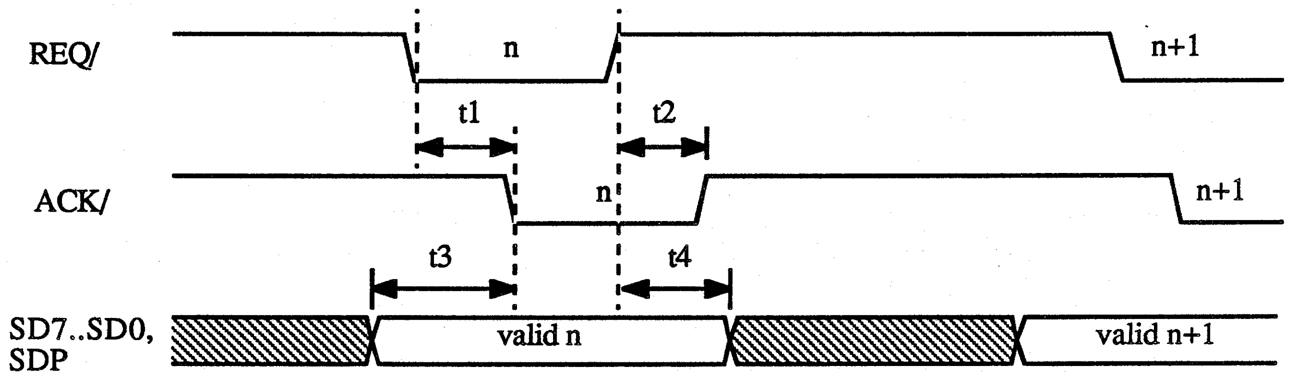
## Bus Master Memory Read anl Write Cycle - Pipelined Next Address



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

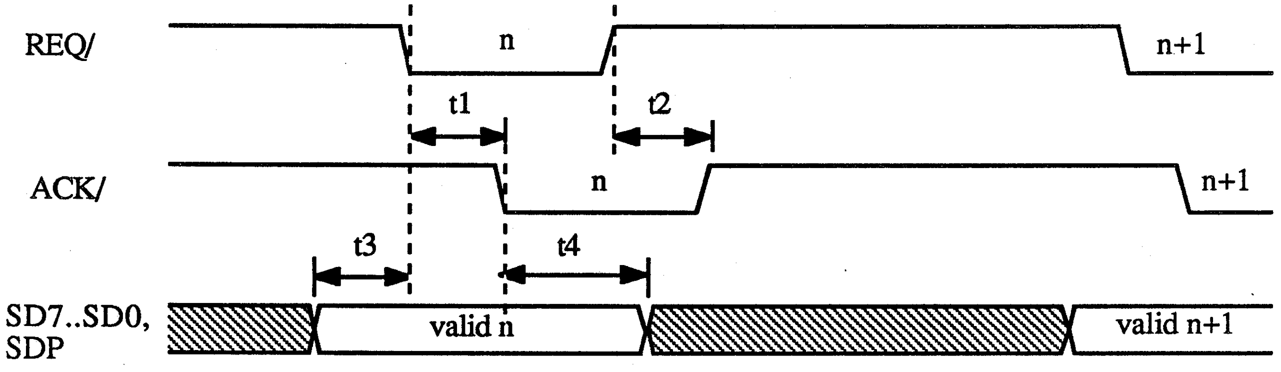
| Bus Master Memory Read and Write Cycle - Pipelined Next Address | | | | |
|---|---|---|---|---|
| Symbol | Description | Min. | Max. | Units |
| t1 | ADS/ valid from CLK rising edge | 4 | 21 | ns |
| t2 | ADS/ hold from CLK rising edge | 4 | 30 | ns |
| t3 | Address valid from CLK rising edge | 4 | 23 | ns |
| t4 | BE3/..BE0/ valid from CLK rising edge | 4 | 24 | ns |
| t5 | MIO/, DC/, WR/ valid from CLK rising edge | - | 4 | ns |
| t6 | NA/ setup to CLK rising edge | 7 | - | ns |
| t7 | NA/ hold from CLK rising edge | 3 | - | ns |
| t8 | READYI/ setup to CLK rising edge | 9 | - | ns |
| t9 | READYI/ hold from CLK rising edge | 6 | - | ns |
| t10 | Data valid from CLK rising edge | 5 | 27 | ns |
| t11 | Data setup to CLK rising edge | 7 | - | ns |
| t12 | Data hold from CLK rising edge | 5 | - | ns |

## Initiator Asynchronous Send



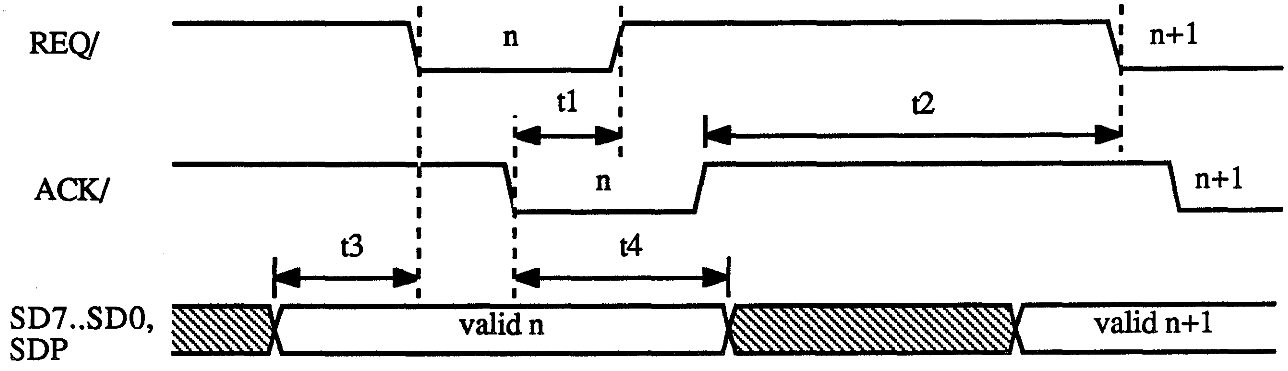
| Initiator Asynchronous Send | | | | |
|---|---|---|---|---|
| Symbol | Description | Min. | Max. | Units |
| t1 | ACK/ asserted from REQ/ asserted | 10 | - | ns |
| t2 | ACK/ deasserted from REQ/ deasserted | 10 | - | ns |
| t3 | Data setup to ACK/ asserted | 55 | - | ns |
| t4 | Data hold from REQ/ deasserted | 20 | - | ns |

## Initiator Asynchronous Receive



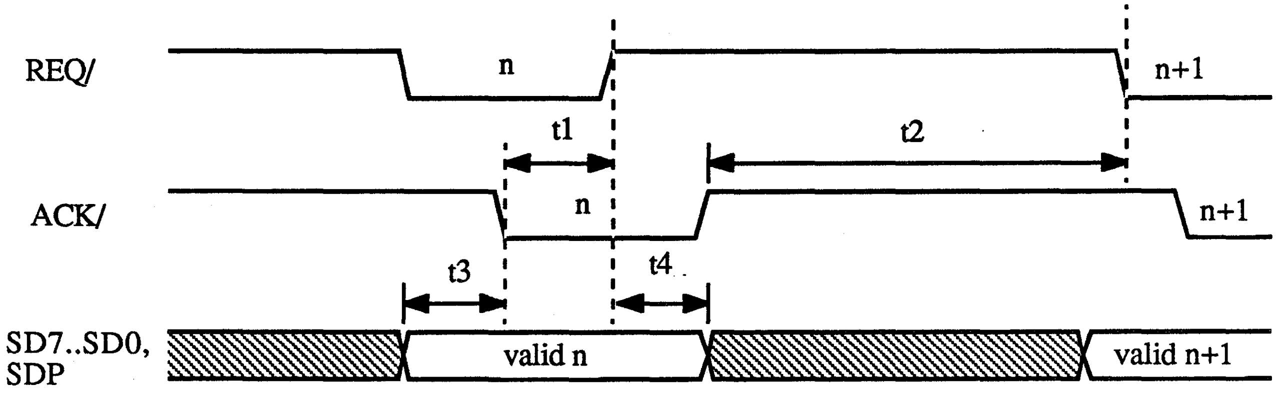
| Initiator Asynchronous Receive | | | | |
|---|---|---|---|---|
| Symbol | Description | Min. | Max. | Units |
| t1 | ACK/ asserted from REQ/ asserted | 10 | - | ns |
| t2 | ACK/ deasserted from REQ/ deasserted | 10 | - | ns |
| t3 | Data setup to REQ/ asserted | 0 | - | ns |
| t4 | Data hold from ACK/ deasserted | 0 | - | ns |

## Target Asynchronous Send



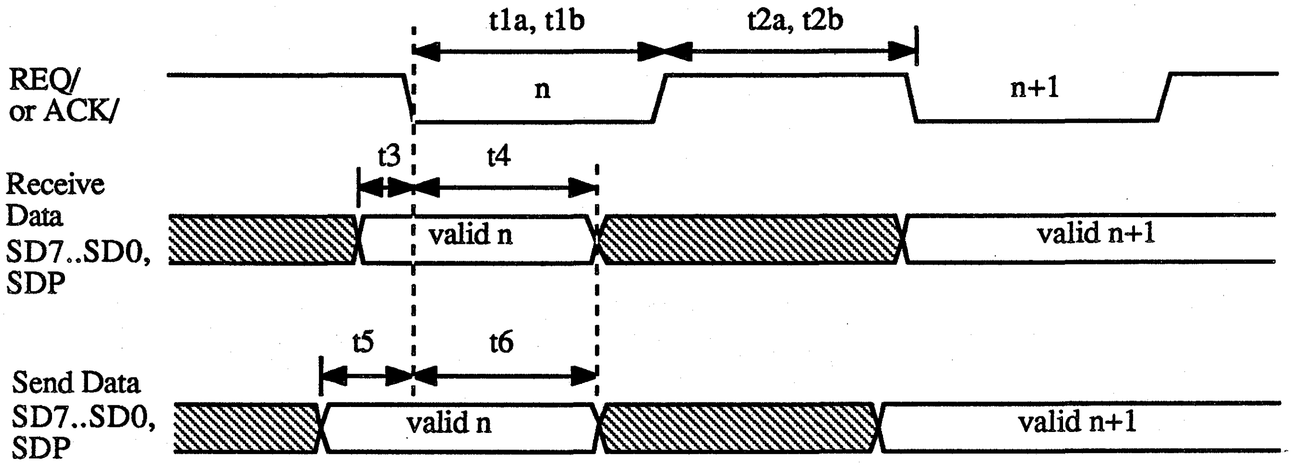
| Target Asynchronous Send | | | | |
|---|---|---|---|---|
| Symbol | Description | Min. | Max. | Units |
| t1 | REQ/ deasserted from ACK/ asserted | 10 | - | ns |
| t2 | REQ/ asserted from ACK/ deasserted | 10 | - | ns |
| t3 | Data setup to REQ/ asserted | 55 | - | ns |
| t4 | Data hold from ACK/ asserted | 20 | - | ns |

## Target Asynchronous Receive



| Target Asynchronous Receive | | | | |
|---|---|---|---|---|
| Symbol | Description | Min. | Max. | Units |
| t1 | REQ/ deasserted from ACK/ asserted | 10 | - | ns |
| t2 | REQ/ asserted from ACK/ deasserted | 10 | - | ns |
| t3 | Data setup to ACK/ asserted | 0 | - | ns |
| t4 | Data hold from REQ/ deasserted | 0 | - | ns |

# Initiator and Target Synchronous Transfers



| Initiator and Target Synchronous Transfers | | | | |
|---|---|---|---|---|
| Symbol | Description | Min. | Max. | Units |
| t1a, t1b | REQ/ or ACK/ assertion pulse width | 90 | - | ns |
| t2a, t2b | REQ/ or ACK/ deassertion pulse width | 90 | - | ns |
| t3 | Receive data setup to REQ/ or ACK/ asserted | 0 | - | ns |
| t4 | Receive data hold from REQ/ or ACK/ asserted | 45 | - | ns |
| t5 | Send data setup to REQ/ or ACK/ asserted | 55 | - | ns |
| t6 | Send data hold from REQ/ or ACK/ asserted | 100 | - | ns |

| Fast Initiator and Target Synchronous Transfers | | | | |
|---|---|---|---|---|
| Symbol | Description | Min. | Max. | Units |
| t1a | Receive data REQ/ or ACK/ assertion pulse width | 30 | - | ns |
| t1b | Send data REQ/ or ACK/ assertion pulse width | 80 | - | ns |
| t2a | Receive data REQ/ or ACK/ deassertion pulse width | 30 | - | ns |
| t2b | Send data REQ/ or ACK/ deassertion pulse width | 80 | - | ns |
| t3 | Receive data setup to REQ/ or ACK/ asserted | 0 | - | ns |
| t4 | Receive data hold from REQ/ or ACK/ asserted | 18 | - | ns |
| t5 | Send data setup to REQ/ or ACK/ asserted | 25 | - | ns |
| t6 | Send data hold from REQ/ or ACK/ asserted | 35 | - | ns |

*Note:*
*The 53C700 can transfer data at 6.25MB/sec for Synchronous Fast SCSI.*

# NCR 53C700 SIOP
# Initiator Register Summary

### 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 |

### DIEN Register R/W (39h)

| | |
|---|---|
| 0 | Enable Illegal Instruction Interrupt |
| 1 | Enable Watchdog Timeout Interrupt |
| 2 | Enable Script Interrupt Instruction Received Interrupt |
| 3 | Enable Script Single Step Interrupt |
| 4 | Enable Aborted Interrupt |
| 5 | Reserved |
| 6 | Reserved |
| 7 | Reserved |

### ISTAT Register R/W (21h)

| | |
|---|---|
| 0 | DMA Interrupt Pending |
| 1 | SCSI Interrupt Pending |
| 2 | Pointer Register Empty |
| 3 | Reserved |
| 4 | Reserved |
| 5 | Reserved |
| 6 | Reserved |
| 7 | Abort Operation |

### SSTAT0 Register R (0Dh)

| | |
|---|---|
| 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 |

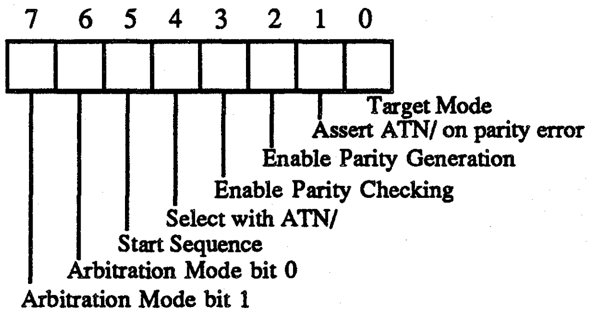
### DSTAT Register R (0Ch)

| | |
|---|---|
| 0 | Illegal Instruction Detected |
| 1 | Watchdog Timeout Interrupt Detected |
| 2 | Script Interrupt Instrucion received |
| 3 | Script Single Step Interrupt |
| 4 | Aborted |
| 5 | Reserved |
| 6 | Reserved |
| 7 | DMA FIFO Empty |

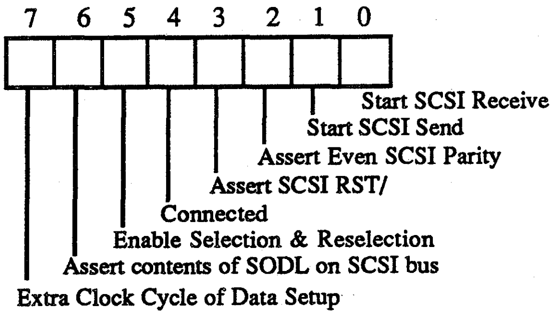
# Appendix B
# NCR 53C700 SIOP386

## Register Summary

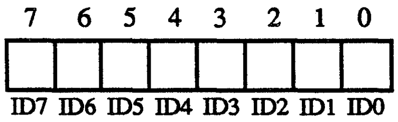
### SCNTL0 Register R/W (00h)



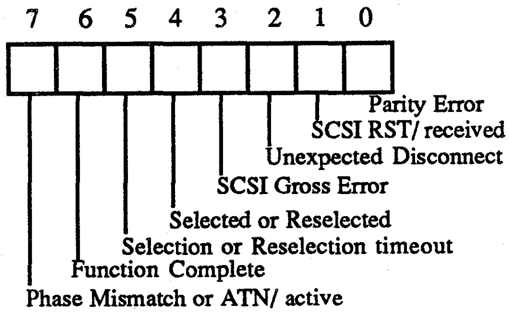
### SCNTL1 Register R/W (01h)


7 6 5 4 3 2 1 0
Start SCSI Receive
Start SCSI Send
Assert Even SCSI Parity
Assert SCSI RST/
Connected
Enable Selection & Reselection
Assert contents of SODL on SCSI bus
Extra Clock Cycle of Data Setup

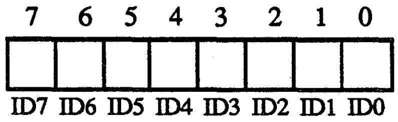
### SDID Register R/W (02h)



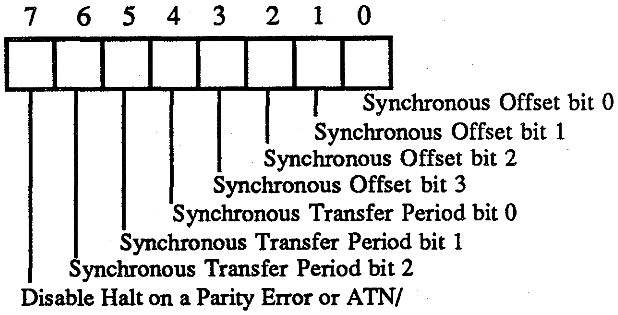
### SIEN Register R/W (03h)



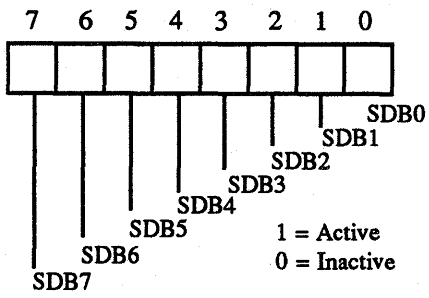
### SCID Register R/W (04h)



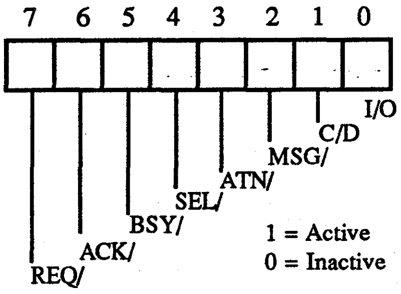
### SXFER Register R/W (05h)


7 6 5 4 3 2 1 0
Synchronous Offset bit 0
Synchronous Offset bit 1
Synchronous Offset bit 2
Synchronous Offset bit 3
Synchronous Transfer Period bit 0
Synchronous Transfer Period bit 1
Synchronous Transfer Period bit 2
Disable Halt on a Parity Error or ATN/

### SODL Register R/W (06h)



### SOCL Register R/W (07h)



# NCR 53C700 SIOP386

## Register Summary

**SFBR Register R (08h)**

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

**SIDL Register R (09h)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SDB7 | SDB6 | SDB5 | SDB4 | SDB3 | SDB2 | SDB1 | SDB0 |

**SBDL Register R (0Ah)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SDB7 | SDB6 | SDB5 | SDB4 | SDB3 | SDB2 | SDB1 | SDB0 |

**SBCL Register R (0Bh)**

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

1 = Active
0 = Inactive

**DSTAT Register R (0Ch)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DMA FIFO Empty | Reserved | Reserved | Aborted | Script Single Step Interrupt | Script Interrupt Instrucion received | Watchdog Timeout Interrupt Detected | Illegal Instruction Detected |

**SSTAT0 Register R (0Dh)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Phase Mismatch or ATN/ active | Function Complete | Selection or Reselection timeout | Selected or Reselected | SCSI Gross Error | Unexpected Disconnect | SCSI RST/ received | Parity Error |

**SSTAT1 Register R (0Eh)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SIDL Register Full | SODR Register Full | SODL Register Full | Arbitration In Progress | Lost Arbitration | Won Arbitration | Status of SCSI RST/ | Status of SCSI Parity signal |

**SSTAT2 Register R (0Fh)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FIFO Flags bit 3 | FIFO Flags bit 2 | FIFO Flags bit 1 | FIFO Flags bit 0 | Latched SCSI Parity signal | Latched MSG/ signal | Latched C/D signal | Latched I/O signal |

# NCR 53C700 SIOP386

## Register Summary

### CTEST0 Register R (14h)

| Bit | Name |
|---|---|
| 7 | Reserved |
| 6 | Reserved |
| 5 | Reserved |
| 4 | Reserved |
| 3 | Reserved |
| 2 | Reserved |
| 1 | Real Target Mode |
| 0 | Data Transfer Direction |

### CTEST1 Register R (15h)

| Bit | Name |
|---|---|
| 7 | Byte Lane 3 Empty in DMA FIFO |
| 6 | Byte Lane 2 Empty in DMA FIFO |
| 5 | Byte Lane 1 Empty in DMA FIFO |
| 4 | Byte Lane 0 Empty in DMA FIFO |
| 3 | Byte Lane 3 Full in DMA FIFO |
| 2 | Byte Lane 2 Full in DMA FIFO |
| 1 | Byte Lane 1 Full in DMA FIFO |
| 0 | Byte Lane 0 Full in DMA FIFO |

### CTEST2 Register R (16h)

| Bit | Name |
|---|---|
| 7 | Reserved |
| 6 | Reserved |
| 5 | SCSI Offset Compare |
| 4 | SCSI FIFO Parity bit |
| 3 | DMA FIFO Parity bit |
| 2 | SCSI TEOP Status |
| 1 | DREQ Status |
| 0 | DACK/ Status |

### CTEST3 Register R (17h)

| Bit | Name |
|---|---|
| 7 | SCSI FIFO bit 7 |
| 6 | SCSI FIFO bit 6 |
| 5 | SCSI FIFO bit 5 |
| 4 | SCSI FIFO bit 4 |
| 3 | SCSI FIFO bit 3 |
| 2 | SCSI FIFO bit 2 |
| 1 | SCSI FIFO bit 1 |
| 0 | SCSI FIFO bit 0 |

### CTEST4 Register R/W (18h)

| Bit | Name |
|---|---|
| 7 | Reserved |
| 6 | Z-Mode |
| 5 | SCSI Z-Mode |
| 4 | SCSI Loopback Enable |
| 3 | SCSI FIFO Write Enable |
| 2 | FIFO Byte Control bit 2 |
| 1 | FIFO Byte Control bit 1 |
| 0 | FIFO Byte Control bit 0 |

### CTEST5 Register R/W (19h)

| Bit | Name |
|---|---|
| 7 | Clock Address Incrementor (DNAD) |
| 6 | Clock Byte Counter (DBC) |
| 5 | Reset SCSI Offset |
| 4 | Master Control for Set or Reset pulses |
| 3 | DMA Direction |
| 2 | EOP |
| 1 | DREQ |
| 0 | DACK/ |

### CTEST6 Register R/W (1Ah)

| Bit | Name |
|---|---|
| 7 | DMA FIFO bit 7 |
| 6 | DMA FIFO bit 6 |
| 5 | DMA FIFO bit 5 |
| 4 | DMA FIFO bit 4 |
| 3 | DMA FIFO bit 3 |
| 2 | DMA FIFO bit 2 |
| 1 | DMA FIFO bit 1 |
| 0 | DMA FIFO bit 0 |

### CTEST7 Register R/W (1Bh)

| Bit | Name |
|---|---|
| 7 | Reserved |
| 6 | Reserved |
| 5 | Reserved |
| 4 | Selection Timeout Disable |
| 3 | DMA FIFO Parity bit |
| 2 | Even Parity - Host bus |
| 1 | DC/ low for Instruction fetch |
| 0 | Differential Mode |

### TEMP Register R/W (1C-1Fh)

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# NCR 53C700 SIOP386

## Register Summary

**DFIFO Register R/W (20h)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Flush DMA FIFO to memory | Clear DMA & SCSI FIFOs | FIFO Byte Offset Counter bit 5 | FIFO Byte Offset Counter bit 4 | FIFO Byte Offset Counter bit 3 | FIFO Byte Offset Counter bit 2 | FIFO Byte Offset Counter bit 1 | FIFO Byte Offset Counter bit 0 |

**ISTAT Register R/W (21h)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Abort Operation | Reserved | Reserved | Reserved | Reserved | Pointer Register Empty | SCSI Interrupt Pending | DMA Interrupt Pending |

**DBC Register R/W (24-26h)**

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | | | | | |

**DCMD Register R/W (27h)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Instruction Op Code bit 1 | Instruction Op Code bit 0 | Function bit 5 | Function bit 4 | Function bit 3 | Function bit 2 | Function bit 1 | Function bit 0 |

**DNAD Register R/W (28-2Bh)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**DSP Register R/W (2C-2Fh)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**DSPS Register R/W (30-33h)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |