NCR

Microelectronics Division
Colorado Springs

NCR 53C700

SCSI I/O Processor

Data Manual

1.0 INTRODUCTION

I/O Performance

The demands on today's I/O interfaces is being pushed by the increasing performance of personal computers and workstations. Extremely fast CPU's, both CISC and RISC, only provide marginal system performance if their I/O interfaces are not properly designed. Faster processors do not equate to higher performance. Amdahl's Law covers this situation. "Assume I/O represents 10% of the system activity and its performance is kept constant. If CPU power is increased by a factor of 10:1, the net improvement is only 5:1. A 100:1 increase in CPU power is valueless if the net improvement in systems performance is only 10:1." Interrupt service routines, which often take in excess of several hundred microseconds to execute, can be a large source of these performance delays. Interrupts may be generated for exception conditions, I/O completion, saving/restoring buffer data pointers (for system check-point/restart), or low probability events available as options in todays SCSI definition. Interrupts can be reduced by using programmed I/O, however, this can be time consuming and requires much of the host computer cycle time. Therefore programmed I/O is not an adequate solution for multi-tasking operations. Another real performance issue is the so called scatter/gather operation. With virtual storage so common today, many I/O's gather the data from several physical addresses in system memory. Latencies inherent in the reinstruct DMA operation can cause serious performance degradation, by allowing the disk drive to slip a latency while the DMA is being reinstructed.

I/O Flexibility

Options in bus protocol allow for increased flexibility. This flexibility is partially responsible for the popularity of the SCSI standard. Flexibility provides users with the ability to configure their systems for a wide range of peripherals (from high performance disk drives to hand held scanners). Additionally, this flexibility offers support for command queueing, asynchronous or synchronous data transfers, caching controllers, peer to peer communication, etc. Unfortunately, flexibility implies firmware complexity, and if these options are not carefully implemented, performance will suffer.

A Better Solution is Required

First generation (NCR5380) SCSI devices were register oriented and required processor intervention even to make the most fundamental protocol decisions. Users liked the flexibility of these devices because the low-level firmware interface provided them with specific real time information about the SCSI bus and improved testability of the SCSI device. This generation of devices typically requires in excess of 4,000 lines of code to specify a SCSI-1 device implementation.

Second generation (NCR53C90) SCSI devices provide on-chip state machines, allowing some complex SCSI sequences to be performed automatically, thereby reducing protocol overhead. However, these devices have no decision making capability, because the internal sequences are fixed in hardware at VLSI design time. Greater than 2,500 lines of driver software is typically required to support this class of SCSI device.

The flexibility of the SCSI bus has caused system integrators and OEM's alike, to struggle with the decision of using these first or second generation SCSI devices standalone or integrating them into intelligent host adapter boards. Non-intelligent SCSI host ports or host bus adapters require a fair amount of processor intervention, however they are relatively inexpensive to implement. Intelligent host adapters, which are an order of magnitude more expensive than non-intelligent adapters, provide slower decision making capabilities (less powerful CPU's), experience interpretation delays (2-8msec required to start any I/O), and suffer from interprocessor communication delays. For these reasons, non-intelligent host adapters outperform their intelligent counterparts in many systems that do not require some type of complex buffering scheme. On the peripheral controller side, space is at a premium and complex peripheral interfaces require powerful microprocessors to transfer data at the high rates available off the peripheral interface. Therefore, SCSI chips that require intense firmware can cause the controller microprocessor to be overworked and unable to perform the required tasks. Because of the limited space available, adding an extra processor, or replacing it with a more powerful one is not always possible.

With MIPS increasing in the system CPU, the delays cause by intelligent host adapter cards and slow peripheral controllers become painfully obvious to the system integrator. The simple solution to this problem is to build complex, versatile, H/W sequences inside the SCSI components or to add additional CPU power in the SCSI device board. Of course both of these solutions are costly (space and component cost) and do not adequately address the problem.

Third Generation Requirements

To adequately accommodate the flexibility requirements of the SCSI bus (reducing interrupts and controlling board cost), an additional level of intelligence and integration is required for next generation silicon. Third generation SCSI chips must be able to make execution decisions based on phase changes on the SCSI bus, as well as compare for specific incoming data values, resulting in a minimum number of interrupts to the external processor. Thus, a truly programmable SCSI chip that executes SCSI oriented commands is required. Additionally, these new chips must be able to reduce interrupt service routine complexities by providing unique status values to the external processor for the few interrupts that do occur. A fully integrated DMA channel must be provided to allow full use of available host bus bandwidth. With todays virtual memory schemes, the ability to support scatter/gather memory operations without processor intervention is key to overall I/O performance. A few hundred lines of driver code is all that's needed to support third generation SCSI devices. This code is required for exception conditions and for passing addresses of the user data buffer to the chip. Error recovery must occur at the high level interface. In second generation chips, the firmware is required to manage every detail of the error recovery mechanism, because the high level interface is fixed and has only one entry point. Programmable SCSI chips allow error recovery using the high level interface, because the algorithm can be entered at any command, and error specific SCSI SCRIPTS™ can be developed.

2.0 The NCR SCSI I/O Processor

The NCR 53C700 is the first truly intelligent SCSI host adapter on a chip. A high-performance reusable SCSI core and an intelligent 32-bit bus master DMA have been integrated with a SCSI SCRIPTS processor to accommodate the flexibility requirements of SCSI-1, 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 that is in the 53C700 is reusable and designed to migrate to SCSI-2 wide and fast requirements. As part of the 53C700 it offers synchronous transfers up to 6.25MBytes/sec with asynchronous transfers greater than 5MBytes/sec. Synchronous offsets up to 8 are supported.

The SCSI core offers low-level register access as well as the high-level control interface. Like first generation SCSI devices, the 53C700 SCSI core can be accessed as a register oriented device. The ability to sample and assert any signal on the SCSI bus can be useful in manufacturing test as well as in diagnostic procedures. Loopback diagnostics are supported to the extent that the SCSI core may perform a self-selection and operate as both an initiator and a target, verifying that internal data paths are operational. Another important feature is the ability to test the SCSI pins for physical connection to the board or the SCSI bus.

Unlike previous generation devices, the SCSI core is controlled by the integrated DMA through a high-level logical interface. High level programming language commands that control the SCSI core may be chained from main host memory. These commands instruct the SCSI core to select, reselect, disconnect, wait for a disconnect, transfer user data, transfer SCSI information, change bus phases, and in general, implement all aspects of the SCSI protocol. Also, the SCSI SCRIPTS processor will transfer execution control (jump, call,return and interrupt) based on SCSI bus phase comparisons. A value in the SCSI SCRIPTS command can be compared to the actual data in value on the SCSI bus, allowing the same transfer of control based on input data compares. A special on-chip 2MIPS processor referred to as the SCSI SCRIPTS processor provides this capability.

DMA COMPONENT

The DMA component is a bus master DMA device that is easily attached to the 80486, 80386, 80286, 80386SX, and 80376 processors. It has been designed to 25 Mhz 80386 bus timings and may be externally adapted to ISA (AT), EISA, Micro Channel™ , etc.

The chip 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. An on-chip 32 byte FIFO allows 2,4, or 8 long words to be burst across the memory bus interface providing memory transfer rates in excess of 50 MBytes/sec.

Since the DMA is tightly coupled to the SCSI core through the SCSI SCRIPTS processor, uninterrupted scatter/gather memory operations are supported, with only a 500 nano second delay between memory segment transfers.

A Watchdog Timer is included as a "bus safety" feature and a flexible arbitration scheme allows for either daisy chained or "ored" memory bus request implementations.

SCSI SCRIPTS™ Processor

The SCSI SCRIPTS processor is a specially designed 2 MIPS processor that allows both DMA and SCSI instructions to be fetched from host memory. Algorithms written in the SCSI SCRIPTS language and then compiled, can control the actions of the SCSI and DMA cores and are executed from 16 or 32-bit system memory. This allows complex SCSI bus sequences to be executed independently of the host CPU.

One of the powerful aspects of the SCSI SCRIPTS processor is the ability to begin a SCSI I/O operation in 500nsec. This compares to the 2-8msec required for traditional intelligent adapters. The SCSI SCRIPTS processor not only offers performance but allows algorithms to be tailored to tune SCSI bus performance, adjust to new bus device types (i.e. scanners, communication gateways, etc.), or adapt to changes in the SCSI logical bus definitions and quickly incorporate new or popular options. Therefore, SCSI flexibility can be implemented without sacrificing I/O performance.

SCSI SCRIPTS are entirely independent of the CPU and system bus being used. This means that scripts for an EISA implementation of a 80386 can be identical to the scripts for a 80386SX Micro Channel™ implementation.

NCR SCSI SCRIPTS™ Description

After power up and initialization of the 53C700, the user may operate the chip in one of two modes; 1) low level register interface, 2) SCSI SCRIPTS chained mode.

In the low level register interface, the user has access to the DMA control logic and the SCSI bus control logic and is able to operate the chip much like an NCR 53C80. An external processor has access to the SCSI bus signals and the low level DMA signals, allowing a user to devise a complicated board level test algorithm. The interface is useful for backward compatibility with SCSI devices that require certain unique timings or bus sequences to operate properly. Another feature allowed at the low level is loop back testing. With the loop back mode, a user can direct the SCSI core to talk to the DMA core for testing internal data paths all the way out to the chip's pad.

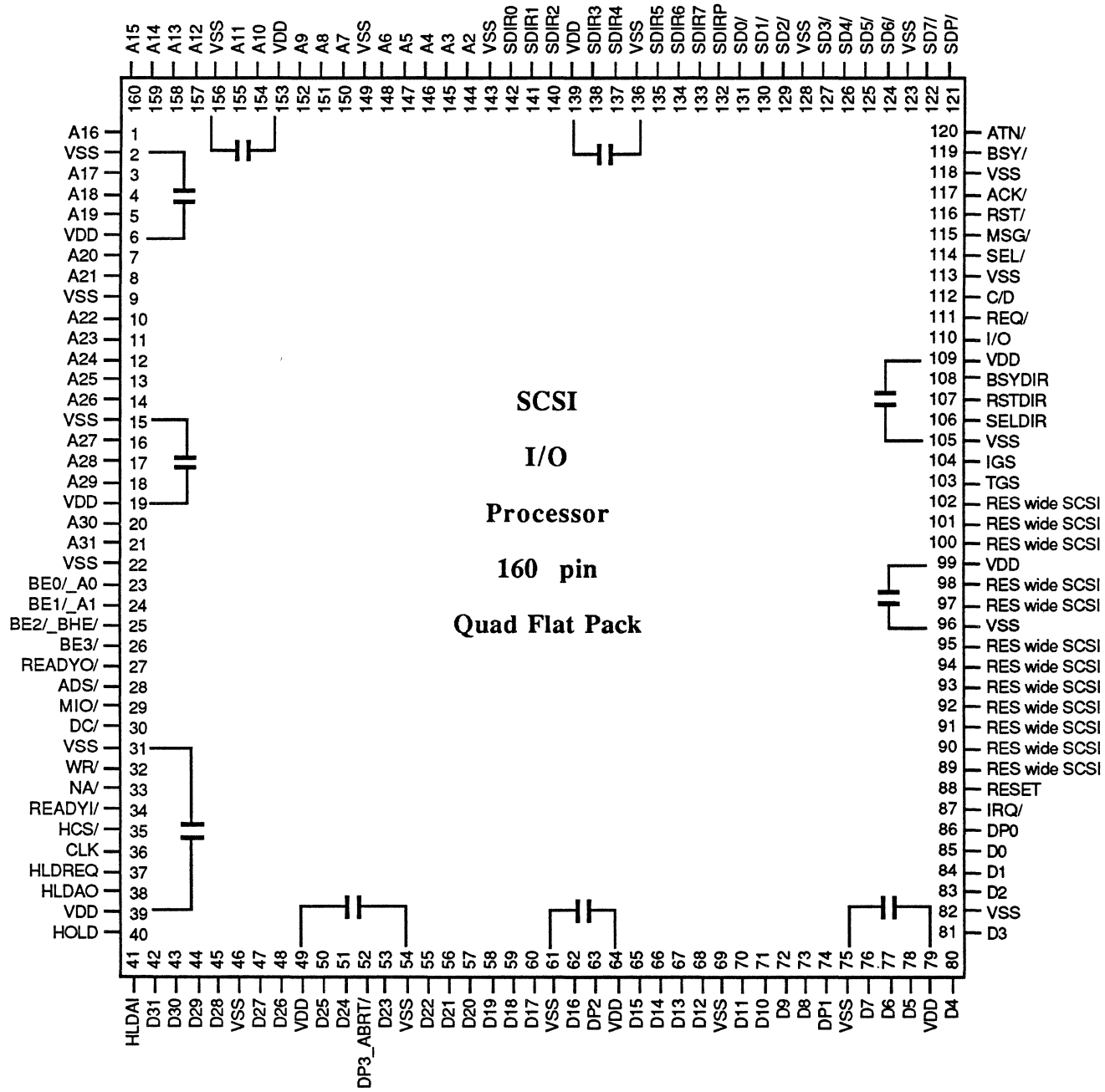
Operating in the chained mode, the 53C700 requires only a SCSI SCRIPTS start address and then all subsequent commands are fetched from external memory. Four bytes (or optionally two) at a time are fetched across the iAPX 286/386 DMA interface and loaded into the command register. Command fetch and decode time is minimal (about 500 ns), so little performance penalty is paid for this feature. Commands will be fetched until an interrupt command is encountered, or until some external, unexpected event (e.g. hardware error detected) causes an interrupt to the external processor. Given the rich set of SCSI oriented features included in the command set, and the ability of the user to re-enter the SCSI algorithm at any point, this high level interface is all that is required for both normal and exception conditions. Therefore the user is never required to switch to a low level mode for error recovery as is the case with today's second generation SCSI VLSI.

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 ns to Begin Compared to 2 - 8 ms
	- Performs Complex Bus Sequences without Interrupts Including Restore Data Pointers
	- Unique Interrupt Status Reporting - Reduces ISR Overhead
	- High-Speed Async/Sync SCSI Bus Transfers 5.0 Mbytes/s asynchronous 6.25 Mbytes/s synchronous(with future migration to 10Mbytes/s)
	- Memory transfers in excess of 50Mbytes/s
• INTEGRATION	- Full 32-Bit DMA Bus Master
	- High Performance SCSI Core
	- Integrated SCRIPTS PROCESSOR
	- Allows For 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 busses
	- 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

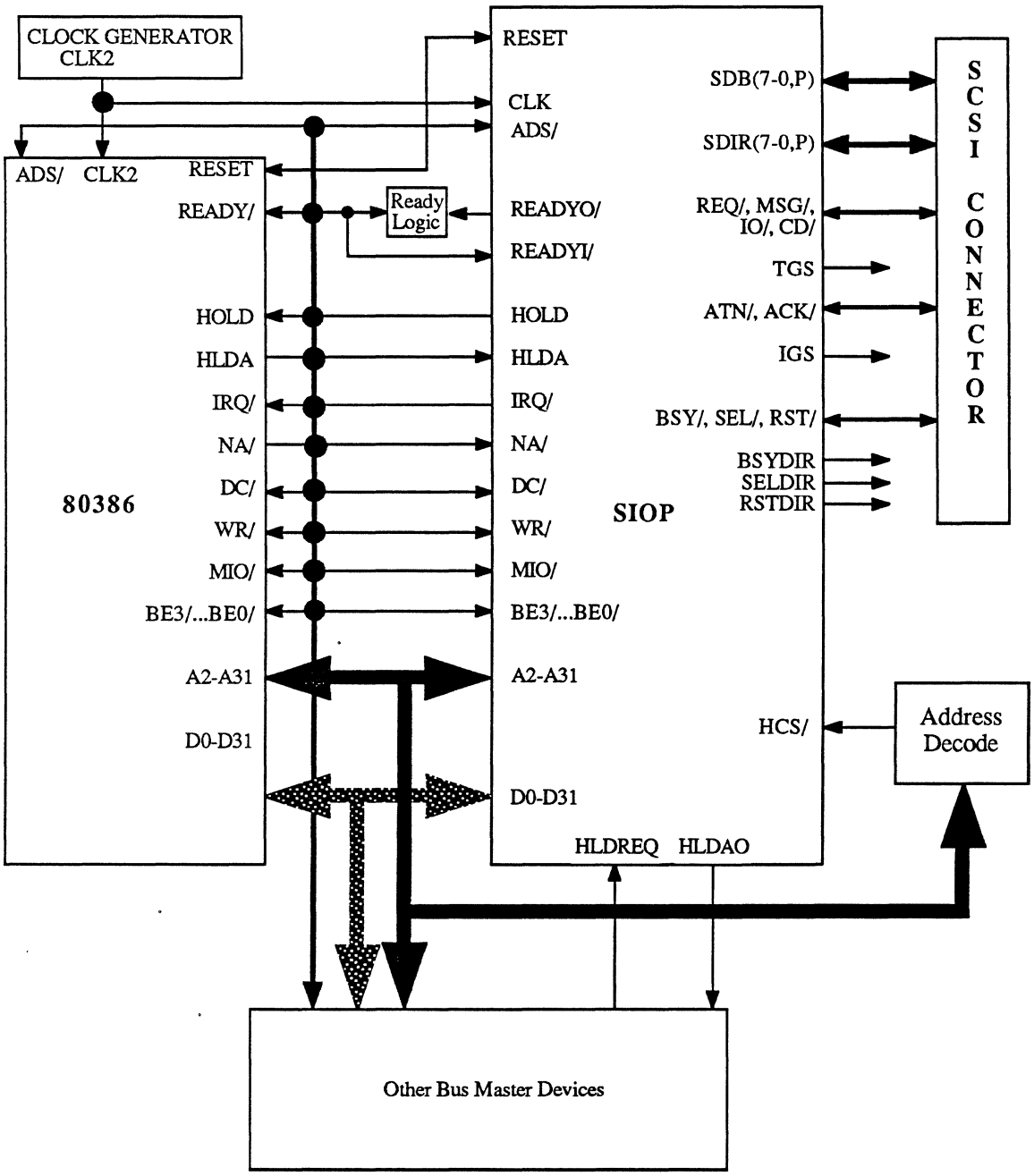
- RELIABILITY
  - 10K volts ESD protection SCSI Signals
  - Greater than 350 mV SCSI Bus Hysteresis
  - Protection against Bus Reflections due to Impedance Mismatches
  - Controlled Bus Assertion Times (reduces RFI, improves reliability, and eases FCC certification)
  - Latch-up protection greater than 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

NCR 53C700 SIOP386 Pinout



* note: The above decoupling is recommended for optimum noise isolation. This configuration takes advantage of the seperate ground planes contained in the SIOP for address and data lines, control lines and SCSI pads.

SCSI I/O Processor Block Diagram



SCSI I/O Processor Pinouts

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

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

3.0 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.

3.1 80386/80286 interface signals

Symbol	Label	Type	Pin#
A31-A2	Host Address Bus	Input/Output	21-20, 18-16, 14-10, 8-7, 5-3, 160-157, 155-154, 152-150, 148-144

These three-state signals provide physical memory addresses or I/O port addresses depending on the state of the MIO/ signal. They are driven by the 386 when the chip is in slave mode and by the SIOP during a DMA transfer. In the 286 mode the A31-24 signals are driven to reflect the value contained in the **DNAD** Register. These upper address signals can be used for memory paging. The address outputs are synchronized with the CLK signal. When transferring data to/from the SIOP's registers, A5-A2 are used in 80386 mode, and A5-A0 are used in 80286 mode.

D31-D0	Host Data Bus	Input/Output	42-45, 47-48, 50-51, 53, 55-60, 62, 65-68, 70-73, 76-78, 80-81, 83-85

These three-state signals provide the general purpose data path between the 80386/80286 and the SIOP. Data bus inputs and outputs are active high. Data can be transferred on 32-bit and 16-bit busses. 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.

DP3_ABRT/	Host Data Parity Bit 3/Abort	Input/Output	52

This three-state signal can be used as the parity bit for D31-D24. When the pass parity option is not desired, this signal can be used as an Abort Transfer pin. Parity generation must be enabled(bit 2 of the **SCNTL0** Register must be set to 1) for this signal to function as the ABRT/ signal. When this signal is asserted the DMA transfer is aborted. When using this signal as a parity signal, this output is synchronized with the CLK signal.

DP2-DP0	Host Data Parity Bits	Input/Output	63, 74, 86

These three-state signals represent 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, these signals are placed in a high-impedance state. Parity outputs are synchronized with the CLK signal.

BE3/	Byte Enable 3	Input/Output	26

This three-state signal is used to enable data transfer in the D24-D31 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. This output is synchronized with the CLK signal.

BE2/_BHE/	Byte Enable 2 - High Byte Enable	Input/Output	25

In 386 mode, this three-state signal is used to enable data transfer in the D23-D16 data byte lane. In 286 mode it is used to distinguish between 8-bit and 16-bit data transfers. Bit 4 of the **DMODE** Register determines whether the chip operates in 286 mode or 386 mode. This output is synchronized with the CLK signal.

Symbol	Label	Type	Pin#
BE1/_A1	Byte Enable 1 - Address A1	Input/Output	24

In 386 mode, this three-state signal is used to enable data transfer in the D15-D8 data byte lane. In 286 mode, this pin is used as 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 mode or 386 mode. This output is synchronized with the CLK signal.

BE0/_A0	Byte Enable 0 - Address A0	Input/Output	23

In 386 mode, this three-state signal is used to enable data transfer in the D7-D0 data byte lane. In 286 mode, this pin is used as address line A0 and should be connected to the A0 pin of the 286. Bit 4 of the **DMODE** Register determines whether the chip operates in 286 mode or 386 mode. This output is synchronized with the CLK signal.

WR/	Write - Read Control signal	Input/Output	32

This three-state signal defines the type of bus cycle being performed. When the SIOP is in Slave mode, high equals a write to the chip, and low equals a read from the chip. When the SIOP is in Master mode, high equals a write to the system bus, and low equals a read from the system bus. This output is synchronized with the CLK signal.

DC/	Data - Control signal	Output	30

This three-state signal defines the type of bus cycle being performed. A high signal, indicates that 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. This output is synchronized with the CLK signal.

MIO/	Memory - I/O signal	Output	29

This three-state signal defines the type of bus cycle being performed. When high, transfer is to a memory address, and when low, transfer is to an I/O address. This output is synchronized with the CLK signal.

ADS/	Address Status	Input/Output	28

This three-state signal 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. This output is synchronized with the CLK signal.

NA/	Next Address Request	Input	33

When the SIOP is in Master mode, this input is used to indicate 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 prepared to accept new values of BE3/, BE2/_BHE/, BE1/_A1, BE0/_A0, A31-A2, WR/, DC/, and MIO/. It is monitored only when in master mode. This signal should not be driven active during a slave access to the SIOP.

READYI/	Transfer Acknowledge In	Input	34

In master mode, READYI/ is an input which 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.

Symbol	Label	Type	Pin#
READYO/	Transfer Acknowledge Out	Output	27

When the SIOP is in Slave mode, it asserts READYO/ to acknowledge the completion of a bus cycle. When the SIOP is selected as a slave device, the READYO/ will be deasserted for a minimum of 2 bus cycles before becoming asserted. This signal is not used in Master mode. This output is synchronized with the CLK signal.

HOLD	Bus Hold Request Output	Output	40

This output only signal is asserted when the SIOP needs to gain 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. This output is synchronized with the CLK signal.

HLDREQ	Bus Hold Request Input	Input	37

This signal is a system bus request input which indicates that another bus master device is requesting use of the host bus. This allows the system to incorporate a daisy-chaining* technique for handling requests for use of the system bus. If another bus master device requests the bus at the same time as the SIOP, the SIOP will have priority over the other bus master device. This input signal can be asserted asynchronously by another bus master device.

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

HLDAI	Bus Hold Acknowledge In	Input	41

This input only signal is asserted by the host CPU to indicate that it has given up the system bus. This signal will be passed through to the HLDAO pin unless the SIOP requires use of the bus.

HLDAO	Bus Hold Acknowledge Out	Output	38

This output signal is a copy of HLDAI signal 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. This output is synchronized with the CLK signal.

RESET	Hardware Reset	Input	88

When this active-high signal is asserted, all registers will be set to the default values as described in the register sections. It is connected to the 80386 RESET line. This signal is also used to define the ø1 and ø2 clock phases.

IRQ/	Interrupt request	Open-drain output	87

This signal will be asserted in response to an interrupting condition or by issuing a SCSI SCRIPTS™ Interrupt instruction. It is an open-drain signal with an internal pull-up resistor. This output is synchronized with the CLK signal.

HCS/	Host Chip Select	Input	35

This signal is generated by external address decoding to allow the SIOP's registers to be read/written. The external address decode logic generating this signal should include the MIO/ signal to allow the SIOP to be memory or I/O mapped.

CLK	System Clock	Input	36

This signal 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 16.7 MHz to 50 MHz.

3.2 SCSI Bus Signals

Symbol	Label	Type	Pin#
SDIR7-SDIR0	SCSI Data Direction	Output	133-135, 137-138, 140-142

In differential mode, these signals are used to 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 this signal is low, the direction is from the SCSI bus to the SIOP. These signals are always valid even in single-ended mode.

SDIRP	SCSI Parity Direction	Output	132

In differential mode, this signal is used to control the direction of an external differential pair transceiver for the SDP/ signal. When this signal is high, the direction is from the SIOP to the SCSI bus. When this signal is low, the direction is from the SCSI bus to the SIOP. This signal is always valid even in single-ended mode.

ATN/	SCSI Attention	Input/Output	120

This active low signal is asserted by the initiator to indicate to the target that a Message Out phase is desired. This open-drain signal can be directly connected to the single-ended SCSI ATN line. In differential mode, the IGS output is used to control the direction of this signal.

MSG/	SCSI Message Phase signal	Input/Output	115

This active-low signal is asserted with the I/O and C/D signals by the target to determine the information transfer phase. This open-drain signal can be directly connected to the single-ended SCSI MSG line. In differential mode, the TGS output is used to control the direction of this signal.

I/O	SCSI Input-Output Phase	Input/Output	110

This active-low 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 with respect to the initiator. This open-drain signal can be directly connected to the single-ended SCSI I/O line. In differential mode, the TGS output is used to control the direction of this signal.

C/D	SCSI Control - Data Phase	Input/Output	112

This active-low signal is asserted with the MSG/ and I/O signals by the target to determine the information transfer phase. This open-drain signal can be directly connected to the single-ended SCSI C/D line. In differential mode, the TGS output is used to control the direction of this signal.

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

Key: "0" not asserted
"1" asserted

Table of SCSI Phases

Symbol	Label	Type	Pin#
REQ/	SCSI Data Transfer Request	Input/Output	111

This active-low signal is asserted by the target request a data transfer. When this signal becomes active, the MSG/, C/D, and I/O phase lines are considered valid. This open-drain signal can be directly connected to the single-ended SCSI REQ line. In differential mode, the TGS output is used to control the direction of this signal.

ACK/	SCSI Acknowledge	Input/Output	117

This active-low signal is asserted by the initiator acknowledge a data transfer. It is asserted in response to the REQ/ signal. This open-drain signal can be directly connected to the single-ended SCSI ACK line. In differential mode, the IGS output is used to control the direction of this signal.

BSY/	SCSI Busy	Input/Output	119

This active-low signal is asserted to indicate that the SCSI bus is busy. It is driven active by a device wanting to arbitrate for use of the SCSI bus. Once the arbitration and selection phases are complete, the target will drive this signal active. This open-drain signal can be directly connected to the single-ended SCSI BSY line. In differential mode, this signal is an input only and the BSYDIR signal is used to assert BSY/ on the SCSI bus.

RST/	SCSI Reset	Input/Output	116

This active-low signal is asserted to perform a SCSI bus reset. This open-drain signal can be directly connected to the single-ended SCSI RST line. In differential mode, this signal is an input only and the RSTDIR signal is used to assert RST/ on the SCSI bus. When the Reset SCSI bus bit in the **SCNTL1** Register is set to 1, the RST/ signal is asserted and remains asserted until this bit is reset to 0.

SEL/	SCSI Select	Input/Output	114

This active-low signal is asserted to select or reselect another SCSI device. This open-drain signal can be directly connected to the single-ended SCSI SEL line. In differential mode, this signal is an input only and the SELDIR signal is used to assert SEL/ on the SCSI bus.

BSYDIR	SCSI BSY/ Direction	Output	108

This signal is used to control the assertion of the SCSI BSY/ signal in differential mode. This signal should be connected to the driver enable of the differential pair transceiver. When this signal is high, BSY/ is asserted on the SCSI bus. When this signal is low, the differential pair driver is disabled. The SCSI termination resistors will then deassert BSY/ by pulling BSY/ high. This signal is always valid even in single-ended mode.

RSTDIR	SCSI RST/ Direction	Output	107

This signal is used to control the assertion of the SCSI RST/ signal in differential mode. This signal should be connected to the driver enable of the differential pair transceiver. When this signal is high, RST/ is asserted on the SCSI bus. When this signal 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.

SELDIR	SCSI SEL/ Direction	Output	106

This signal is used to control the direction of the SCSI SEL/ signal in differential mode. This signal should be connected to the driver enable of the differential pair transceiver. When this signal is high, SEL/ is asserted on the SCSI bus. When this signal 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.

Symbol	Label	Type	Pin#
TGS	Target Group Select	Output	103

This signal is used to enable the external tranceivers 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 this signal is low, REQ/, MSG/, C/D and I/O are inputs. This signal is always valid even in single-ended mode.

IGS	Initiator Group Select	Output	104

This signal is used to enable the external tranceivers to drive SCSI ACK/ and ATN/ when the SIOP is operating 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.

4.0 REGISTERS

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

NCR 53C700 SIOP386 Register Address Map

								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							(R/W)	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

4.1 53C700 SIOP386 REGISTER DESCRIPTIONS

Notes: NCR reserved bits are designated as "RES" in each register map. These bits should always be written to 0, and all information read from them should be discarded. Unless otherwise indicated, all bits in 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 depending on the data buswidth(8, 16 or 32bits).

4.2 SCSI Registers

Register 00 --- SCSI Control Register 0 (**SCNTL0**), Read/Write

ARB1 7	ARB0 6	START 5	WATN/ 4	EPC 3	EPG 2	AAP 1	TRG 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
0	1	Selection Only
1	0	Full Arbitration
1	1	Full Arbitration, Selection or Reselection

An arbitration or selection sequence is started by setting the Start Sequence bit in this register. The sequence can be aborted by resetting the Start Sequence bit. Before starting any of the arbitration sequences in Low-Level mode, the Connected bit in the **SSTAT1** Register should be checked to see if the SIOP is connected to the SCSI bus. If the Connected bit is set to 1, the SIOP has been selected or reselected and is already connected to the SCSI bus.

Simple Arbitration - In this mode, the SIOP waits for a bus free condition to occur, asserts BSY/ and asserts 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 check to see if a higher priority SCSI ID is present by reading the **SBDL** Register.

Full Arbitration - In this mode, the SIOP waits for a bus free condition to occur, asserts BSY/ and asserts the highest priority ID stored in the **SCID** Register onto the SCSI bus. If the SEL/ signal is asserted by another SCSI device or a higher priority SCSI ID is detected, 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. At this time, the Won Arbitration Bit is set in the **SSTAT1** Register.

Selection Only - In this mode, the chip does not arbitrate for the SCSI bus. Full Arbitration must be performed prior to Selection Only. The SIOP performs selection by asserting SEL/, the target's ID(stored in the **SDID** Register) and the SIOP's ID(the highest priority ID stored in the **SCID** Register) onto the SCSI bus. 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.

Full Arbitration, Selection & Reselection - In this mode, the SIOP waits for a bus free condition, 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. At this time, 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™, Start Sequence is controlled by the SCRIPTS PROCESSOR. The Start Sequence bit is intended to be used in Low-Level mode. This bit is cleared automatically when the selection sequence is complete. The arbitration sequence can be aborted by resetting this bit to 0. If the sequence is aborted, the Connected bit in the **SSTAT0** Register should be checked to make sure 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. It is intended to be used in Low-Level mode. If 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 receiving data 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 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 an interrupt can optionally be generated. If the SIOP is operating in target mode, bytes with parity errors written into the DMA FIFO can be stopped from being written to the SCSI bus. A phase change to Message In phase must be generated to indicate this condition. If the SIOP is operating in initiator mode and a parity error is detected, ATN/ can optionally be asserted 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

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. In order to enable the DP0_ABRT/ signal to function as an Abort signal (ABRT/), this bit should be written 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. 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 will default 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 be used to 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 will assert the SCSI data bus with the contents of the SCSI Output Data Register (**SODL**). When the SIOP is an initiator, the SCSI I/O signal must be inactive for the **SODL** contents to be asserted 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. The contents of the **SODL** Register can be asserted at anytime, even before the SIOP becomes connected as an initiator or target. This bit should be written to 0 when executing SCSI SCRIPTS™. It can be used when doing diagnostics testing or when operating in Low-Level mode.

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

When this bit is set to 1, the SIOP becomes enabled to respond to bus-initiated selections and reselections. The SIOP can respond to selections and reselections in both initiator and target roles. If the disconnect - reconnect feature is to be supported, this bit should be written to 1 as part of the initialization routine.

Bit 4 CON Connected

This bit will automatically be set anytime the SIOP becomes connected as an initiator or connected as a target. It becomes 1 after successfully completing arbitration. It also will be set to 1 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 currently connected on the SCSI bus.

Bit 3 RST Assert SCSI RST/ signal

Writing this bit to 1 will cause the SCSI RST/ signal to be asserted. The RST/ output remains asserted until this bit is written to 0. The 25 µs 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 will check data received for even parity. This bit is intended for use in diagnostic testing and should be 0 during normal operation. It can be used 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. 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. It is automatically set to 1 by the SCRIPTS PROCESSOR to start a SCSI receive operation when executing SCSI SCRIPTS™. It is intended for Low-Level operation.

Register 02 --- SCSI Destination ID Register (**SDID**), Read/Write

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

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

This register is used 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. This register is written by the SCRIPTS PROCESSOR with the destination SCSI ID when executing SCSI SCRIPTS™. 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 Phase mismatch - Initiator mode
or ATN/ active - Target mode

In initiator mode, this bit is defined as 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 becomes 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 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. For this case, 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 selection only or full arbitration 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 250ms 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 diconnects. 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. 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 7	ID6 6	ID5 5	ID4 4	ID3 3	ID2 2	ID1 1	ID0 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 is used to set 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 7	TP2 6	TP1 5	TP0 4	MO3 3	MO2 2	MO1 1	MO0 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 initiator mode, this bit is defined as Disable Halt on Parity Error. 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. 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. 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. While sending data in target mode with pass parity enabled, the byte with the parity error 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
= TCP * (4 + XFERP), if Bit 7 = 0 in the **SCNTL1** Register

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
TCP = 1 ÷ (CLK input frequency ÷ 1.5), if Bit 7 = 0 & Bit 6 = 1 in the **DCNTL** Register
TCP = 1 ÷ (CLK input frequency ÷ 1), if Bit 7 = 1 & Bit 6 = 0 in the **DCNTL** Register

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

Bit 3 MO3 Maximum SCSI Synchronous Offset Bit 3
Bit 2 MO2 Maximum SCSI Synchronous Offset Bit 2
Bit 1 MO1 Maximum SCSI Synchronous Offset Bit 1
Bit 0 MO0 Maximum SCSI Synchronous Offset Bit 0

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

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

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

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 in the **SCNTL1** Register. This register should be used when sending data via programmed I/O. Data flows through this register when sending data in any mode. It is also used to allow the user 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™. It should not be written once the SIOP becomes connected and starts executing SCSI SCRIPTS™.

Register 08 --- SCSI First Byte Received Register **(SFBR)**, Read Only

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

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

This register contains the first byte received in any asynchronous information transfer phase. For example, when the SIOP is operating in initiator mode, this register contains the first byte received in any of the following phases: Message In, Status, or Data In. When a Block Move Instruction is executed for a particular phase, then the next input phase will have the first byte received stored in this register - even if the next input phase is the *same* phase. Input here refers to the phase that is an input to the SIOP. For example, if a Block Move instruction is executed for a Data In phase, the SIOP completes the data transfer for that instruction, the target continues to assert the Data In phase, then the next byte received in response to a Block Move instruction is stored in this register. This register is compared to the Data to be compared field in a Transfer Control instruction.

Register 09 --- SCSI Input Data Latch Register **(SIDL)**, Read Only

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

Bits 7 - 0 SD7 - SD0 SCSI Input Data Latch Register Bit 7 - SCSI Input Data Latch Register Bit 0

This register is used primarily for diagnostics testing, programmed I/O operation or error recovery. Data received from the SCSI bus can be read from this register. Data can be written to the **SODL** Register and then read back into the SIOP by reading this register to provide "loopback" testing. When receiving SCSI data, the data will flow into this register and out to the host FIFO. This register differs from the **SBDL** Register by the fact that 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 7	DB6 6	DB5 5	DB4 4	DB3 3	DB2 2	DB1 1	DB0 0
Default >>>	0	0	0	0	0	0	0	0

Bits 7 - 0 SD7 - SD0 SCSI Data Bit 7 - SCSI Data Bit 0

This register contains the status of the SCSI data bus. 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 the register is read. This register should be used when receiving data 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 7	ACK 6	BSY 5	SEL 4	ATN 3	MSG 2	C/D 1	I/O 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 status of the SCSI control bus. 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 7	RES 6	RES 5	ABRT 4	SSI 3	SIR 2	WTD 1	OPC 0
Default >>>	1	0	0	0	0	0	0	0

Reading this register will clear any DMA interrupts that may have caused the IRQ/ signal to be asserted.

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. It can be used to determine if any data resides in the FIFO when an error occurs and an interrupt has been 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 SPI SCRIPT Pipeline Interrupt

This status bit is set when an interrupt condition occurs during Pipeline operation. The Pipeline Mode bit in the **DCNTL** Register must be 1 for the SCRIPT Pipeline Interrupt bit to be set. When executing SCSI SCRIPTS™ automatically, this bit is never set to 1. The following conditions can cause a SCRIPT Pipeline Interrupt:

1) The Single Step Mode bit in the **DCNTL** Register is equal to 1, then there will be a SCRIPT Pipeline Interrupt after successfully completing each instruction.
2) The SIOP encounters a branch condition while executing pipelined instructions

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. The Watchdog Timer is only used for the 80386 memory interface. If this counter decrements to zero, it indicates that the memory device did not assert its READY/ signal within the specified timeout period.

Bit 0 OPC Illegal Instruction Detected

This status bit is set anytime an illegal instruction is decoded. This bit may be set when the SIOP is operating in single step mode or when the SIOP is automatically executing SCSI SCRIPTS™.

Register 0D --- SCSI Status Register 0 **(SSTAT0)**, Read Only

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

Each of these bits correspond to a SCSI condition that can cause an interrupt to be generated by the SIOP. 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. 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. This status bit is used in diagnostics testing or in Low-Level mode.

Bit 6 CMP Function Complete

This bit is set to 1 when an arbitration only, selection only, or full arbitration sequence has completed.

Bit 5 STO Selection or Reselection Timeout

This bit is set to 1 when a selection or reselection timeout occurs. This means that the device trying to be selected or reselected did not respond within the suggested 250 ms 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 & Reselection bit must have been written to 1 in the **SCNTL1** Register for the SIOP to respond to selection & 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 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

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 is operating 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 is operating in Low-Level Mode, then 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).

Register 0E --- SCSI Status Register 1 **(SSTAT1)**, Read Only

	ILF 7	OLF 6	ORF 5	AIP 4	LOA 3	WOA 2	RST/ 1	SDP/ 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 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 5 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 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. The Arbitration Mode selected in the **SCNTL0** Register must be the arbitration only mode or the full arbitration & selection mode for this bit to be set.

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

Bit 7 FF3 FIFO Flags bit 3
Bit 6 FF2 FIFO Flags bit 2
Bit 5 FF1 FIFO Flags bit 1
Bit 4 FF0 FIFO Flags bit 0

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

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 whenever a new byte is latched into the **SIDL** Register. When this bit is 1, the parity signal is active, and 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 7	RES 6	RES 5	RES 4	RES 3	RES 2	RTRG 1	DDIR 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 whether the chip "thinks" it is in target mode or initiator mode. It does not reflect the status of the Target Mode bit in the **SCNTL0** Register. It reflects the operating mode of the logic inside the SIOP. For example, the Target Mode bit in the **SCNTL0** Register might be written to 0 indicating that the SIOP will be operating in initiator mode. However, if the SIOP then becomes selected as a target, this bit will reflect the fact 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 7	FMT2 6	FMT1 5	FMT0 4	FFL3 3	FFL2 2	FFL1 1	FFL0 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 indicate whether the bottom bytes in the DMA FIFO 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 indicate whether the top bytes in the DMA FIFO 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 7	RES 6	SOFF 5	SFP 4	DFP 3	TEOP 2	DREQ 1	DACK 0
Default >>> 0	0	1	0	0	0	0	1

Bit 7	RES	Reserved
Bit 6	RES	Reserved

Bit 5 SOFF SCSI Offset Compare

This bit operates differently depending on whether the chip is currently an initiator or a target. 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 data is read out of the FIFO by reading the **CTEST6** Register. Reading the **CTEST6** Register unloads one data byte currently residing in 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 will fall 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, and 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, and 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, and when this bit is 0, DACK/ is active.

Register 17 --- Chip Test Register 3 (**CTEST3**) Read Only

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

Bits 7 - 0 SF7 - SF0 SCSI FIFO Bit 7 - SCSI FIFO Bit 0

Reading this register will unload the bottom byte of the eight-deep SCSI Synchronous FIFO. Reading this register will also latch 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 7	ZMOD 6	SZM 5	SLOB 4	SFWR 3	FBL2 2	FBL1 1	FBL0 0
Default >>>	0	0	0	0	0	0	0	0

Bit 7 RES Reserved

Bit 6 ZMOD Z Mode - High-Impedance Mode

Writing this bit to 1 will cause the SIOP to place <u>all</u> outputs into the high-impedance state. In order to read data out of the SIOP, this bit must be written to 0.

Bit 5 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.

Bit 4 SLBE SCSI Loopback Enable

Setting this bit to 1 enables "Loopback" Mode. Loopback allows the user to assert any SCSI signal regardless of whether the SIOP is 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 reference the "Loopback Mode" section.

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

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 are used to steer 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 be equal to 0.

Register 19 --- Chip Test Register 5 (**CTEST5**) Read/Write

ADCK 7	BBCK 6	ROFF 5	MASR 4	DDIR 3	EOP 2	DREQ 1	DACK 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 amount that the **DNAD** Register is incremented is dependent on the **DNAD** contents, the bus width, and the host processor being used (80286 or 80386). 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 amount that the **DBC** Register is decremented is dependent on the **DBC** contents, the bus width, and the host processor is used (80286 or 80386). 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(i.e. bits 3 - 0 become active high). When this bit is reset, bits 3 - 0 deassert the corresponding signals(i.e. bits 3 - 0 become active low).

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 indicates that data will be transferred 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 will automatically clear 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 is requesting 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 will automatically clear itself after asserting the DREQ signal.

Bit 0 DACK Data Acknowledge

Setting this bit either asserts or deasserts the internal DACK/ data request signal depending on the current status of the MASR bit in this register. Asserting the DACK/ signal indicates that the DMA portion of the SIOP is acknowledging 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 will automatically clear itself after asserting the DACK/ signal.

Register 1A --- Chip Test Register 6 **(CTEST6)** Read/Write

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

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

Writing to this register causes data to be written to the appropriate byte lane of the DMA FIFO as determined by the FBL bits in the **CTEST4** Register. Reading this register causes data to be unloaded 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, and data read out of the FIFO is read from the bottom of the FIFO. When data is read from the DMA FIFO, the parity bit for that byte is also latched and stored in the DMA FIFO parity bit in the **CTEST2** Register.

Register 1B --- Chip Test Register 7 **(CTEST7)** Read Only

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

Bit 7 RES Reserved
Bit 6 RES Reserved
Bit 5 RES Reserved
Bit 4 RES Reserved

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, a read or a write of the **CTEST6** Register should be performed. When loading data into the FIFO via programmed I/O, this bit will be written to the FIFO as the parity bit for each byte loaded. When writing data to the DMA FIFO, this bit must be set with the appropriate 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 reference the section on "Diagnostics".

Bit 2 EVP Even Parity

Setting this bit to 1 causes the SIOP to send out even parity when sending data to the 80386 bus. The SIOP accomplishes this by inverting the parity bit received from the SCSI bus. In addition, the even parity received from

the 80386 bus is inverted to become odd parity before the SIOP checks parity and sends the data to the SCSI 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 allowing SIOP instructions to be stored in a cache or forcing them to be read out of memory directly. Resetting this bit to 0 causes the SIOP to drive the DC/ signal high when fetching instructions from memory. 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 to 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, reference 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 is used to store 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. **TEMP** should not be written while the SIOP is executing SCSI SCRIPTS™.

Register 20 --- DMA FIFO Register **(DFIFO)**, Read/Write

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

Bit 7 FLF Flush DMA FIFO

When this bit is set to 1, data residing in the DMA FIFO is transferred to or from(according to the DMAWR signal) memory starting at the current address contained in the **DNAD** Register. The internal DMAWR signal controlled by the **CTEST5** Register 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 FIFO Byte Offset Counter Bit 5 - FIFO Byte Offset Counter Bit 0

These six bits indicate the amount of data that has been transferred between the SCSI core and the DMA core. It can be used to determine the number of bytes that reside in the DMA FIFO when an error occurs. These bits are changing when data is being transferred between the two cores. Once the chip has stopped transferring data, these bits are stable. If the Byte Offset Counter bits need to be preserved, then the same value read from these bits should be written to these bits.

To determine how many bytes reside in the DMA FIFO when an error occurs, the following steps must be performed:

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** Register
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 7	RES 6	RES 5	RES 4	RES 3	PRE 2	SIP 1	DIP 0
Default >>>	0	0	0	0	0	1	0	0

This register can be read at any time without interfering in the operation of the SIOP. It can be used as a polling register if interrupts are not enabled.

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, this bit should be reset to 0 before reading the **DSTAT** Register to prevent further Aborted interrupts from being generated. The sequence to abort any operation is described below.

1) Write this bit to 1
2) Wait for an interrupt
3) Read this **ISTAT** Register
4) If the SCSI Interrupt Pending bit is 1, then read the **SSTAT0** Register to determine the cause of the SCSI Interrupt and go back to step 2
5) If the SCSI Interrupt Pending bit is 0, and the DMA Interrupt Pending bit is 1, then write 00h value to this register
6) Read the **DSTAT** Register to verify the aborted interrupt and to see if any other interrupting conditions have occurred.

Bits 6 - 3 RES Reserved

Bit 2 PRE Pointer Register Empty

This status bit is set 1 when the **DSPS** Register & the **DSP** Register are empty. In pipeline mode, this register should be polled 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) DMA interrupt received
3) Interrupt instruction received in SCSI SCRIPTS™ operation
4) Watchdog timer counter decremented to zero, indicating that a host memory timeout occurred
5) illegal instruction detected

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. The function of the bits in this register will be different for each instruction. For a complete description, please reference 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 address pointer for the next transfer that occurs on the 80386 interface. Compare instructions use this register to point to the jump address if the compare was not successful. Block Move instructions use this register to point to the address where data is to be moved.

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 this register is written with the starting address of the SCSI SCRIPTS™, the SCRIPTS™ are automatically fetched and executed until an interrupt condition occurs. In single step mode, there is a SCRIPT Pipeline 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 pipeline interrupt occurs to fetch and execute the next SCSI SCRIPT. In pipeline mode, this register becomes the **DCMD** and **DBC** register. When writing this register 8-bits at a time or 16 bits at a time, writing the upper 8 bits causes the SCSI SCRIPTS™ execution to begin.

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

Default >>> all zeros

If the SIOP is executing SCSI SCRIPTS™, this register should not be written. When executing pipelined instructions, this register becomes the DNAD register and should be loaded with the 32-bit address of the pipelined command.

Register 34 --- DMA Mode (**DMODE**) Read/Write

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

Bit 7 BL1 Burst Length Bit 1
Bit 6 BL0 Burst Length Bit 1

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

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 to be transferred on odd-byte boundaries. The operation will be described for the case where the host bus width is 32-bits and the burst length is 2. The SIOP will assert its HOLD output only when there are two 32-bit words available to transfer. Once these two words have been transferred, then the SIOP will deassert HOLD until there are two more 32-bit words to transfer. If the SIOP accumulates more than two 32-bit words to transfer, then the SIOP will transfer two words, deassert HOLD for a minimum of two bus cycles, and then reassert HOLD to request the bus for another transfer.

Bit 5 BW16 Host Bus Width Equal to 16 bits

When this bit is set to 1, the SIOP will execute 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. The following signals will change function: BE2/ becomes BHE/, BE1/ becomes A1, and BE0/ becomes A0. Block Move instructions transfer data 16-bits at a time. Writing this bit to 1 does not cause SCSI SCRIPTS™ to be loaded 16-bits at a time. The SCSI SCRIPTS™ 16 bit in the **DCNTL** Register will control how SCSI SCRIPTS™ are loaded. This bit should be initialized 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, but only applies to data being transferred to/from memory. Writing this bit to 1 will cause the SIOP to transfer data to an I/O-mapped device. Writing this bit to 0 will cause 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 will disable the address pointer from incrementing after each data transfer. The address pointer is contained in the **DNAD** Register. If this bit is 0, this pointer will be incremented after each data transfer. If this bit is 1, this pointer will not increment after each data transfer. This can be used to allow data to be transferred 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** Register and the **DSPS** Register have different functions. The **DSP** Register will operate as the first 32-bit word of a pipelined instruction. The **DSPS** Register will operate as the second 32-bit word of a pipelined instruction. The execution of pipelined commands operate as follows:

1) Write the Start DMA bit to 1 in the **DCNTL** Register.
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) Poll the Pipeline Register Empty bit in the **ISTAT** Register until it is 1.
5) Load the **DSPS** Register with the second 32-bit word of the next instruction.
6) Load the **DSP** Register with the first 32-bit word of the next instruction. .
7) Go to step 4.

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

Bit 0 MAN Manual Start Mode

Writing this bit to 0 will cause the SIOP to automatically fetch and execute SCSI SCRIPTS™ after the **DSP** Register is written. Writing this bit to 1 disables the SIOP from automatically fetching and executing SCSI SCRIPTS™ after the **DSP** Register is written. Instead, the Start DMA bit in the **DCNTL** Register controls when the operation begins. Once the Start DMA bit in the **DCNTL** Register is set to 1, the SIOP will automatically fetch and execute each instruction.

Register 39 --- DMA Interrupt Enable Register (**DIEN**) Read/Write

RES 7	RES 6	RES 5	ABRT 4	SPI 3	SIR 2	WTD 1	OPC 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 will allow any abort condition to assert the IRQ/ signal. There are two ways that an abort condition can occur: the DP3_ABRT/ input signal was asserted by another device (Parity Generation mode), or a software abort command was issued by writing 1 to Bit 7 of the **ISTAT** Register. Writing 0 to this bit will disable the assertion of IRQ/ when an abort condition occurs.

Bit 3 SPI Enable SCRIPT Pipeline Interrupt

Writing this bit to 1 causes the IRQ/ signal to be asserted when the SCRIPT Pipeline Interrupt bit is set to in the **DSTAT** Register. The Pipeline Mode bit in the **DCNTL** Register must be 1 for the SCRIPT Pipeline Interrupt bit to ever become 1. Writing 0 to this bit will disable the assertion of IRQ/ when a SCRIPT Pipeline Interrupt condition occurs. The following conditions can cause an interrupt when the SIOP is executing Single Step instructions.

1) If the Single Step Mode bit in the **DCNTL** Register is equal to 1, then there will be a SCRIPT Pipeline Interrupt after successfully completing each instruction.
2) If the SIOP encounters a branch condition while executing pipelined instructions

Bit 2 SIR Enable SCRIPT Interrupt Instruction Received Interrupt

Writing this bit to 1 causes the IRQ/ signal to be asserted when the SCRIPT Interrupt Instruction Received bit is set to 1 in the **DSTAT** Register. The SCRIPT Interrupt Instruction Received status bit becomes set when an interrupt instruction is encountered when executing SCSI SCRIPTS™. Writing 0 to this bit will disable the assertion of IRQ/ when a SCRIPT Interrupt instruction is received.

Bit 1 WTD Enable Watchdog Timeout Interrupt

Writing this bit to 1 will cause the IRQ/ signal to be asserted whenever the Watchdog Timer Counter has decremented to zero. The Watchdog Timer is only used for the 80386 or 80286 bus interface. If this counter decrements to zero, it indicates that the memory device did not assert its READY/ output signal within the specified timeout period. Writing 0 to this bit will disable the assertion of IRQ/ when a Watchdog Timeout condition occurs.

Bit 0 OPC Enable Illegal Instruction Interrupt

Writing this bit to 1 will cause the IRQ/ signal to be asserted anytime that an illegal instruction is decoded. This bit can be set when the SIOP is operating in either SCSI SCRIPTS™ mode or Single Step mode. Writing 0 to this bit will disable 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 can be used to provide a timeout mechanism when transferring data 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. This register should be written with the appropriate timeout value during initialization. Every time that the SIOP transfers data to/from memory, the value stored in this register will be loaded into the counter. This timeout feature can be disabled by writing a 00h to this register. The unit time base for this register is 16 CLK input periods. For example, at 50 MHz, the time base for this register would be 16 x 20 ns = 320 ns. If a desired timeout of 50 µs 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 7	CF0 6	S16 5	SSM 4	LLM 3	STD 2	RES 1	RST 0
Default >>>	0	0	0	0	0	0	0	0

Bit 7 CF1 Clock Frequency bit 1
Bit 6 CF0 Clock Frequency bit 0

These two bits are set 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 in order to guarantee that the SIOP meets the SCSI timings defined by the ANSI specification.

CF1	CF0	Clock Frequency
0	0	37.51 - 50 MHz
0	1	25.01 - 37.50 MHz
1	0	16.67 - 25.00 MHz
1	1	Reserved

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

Writing this bit to 1 will cause the SIOP to fetch SCSI SCRIPTS™ instructions 16-bits at a time. SCSI SCRIPTS™ instruction fetches will involve four 16-bit transfers. This bit applies only to SCSI SCRIPTS™ operations and has no effect on data transfers for Block Move instructions. Writing this bit to 0 will cause the SIOP to fetch SCSI SCRIPTS™ instructions 32-bits at a time.

Bit 4 SSM Single Step Mode

Writing this bit to 1 will cause the SIOP to stop after completing each instruction. The SCRIPT Pipeline Interrupt bit in the **DSTAT** Register will become 1 after each instruction. If the Enable SCRIPT Pipeline Interrupt bit is 1 in the **DIEN** Register, then the IRQ/ signal will be asserted after each instruction has been executed. If this bit is 0, then the SIOP will not stop after each instruction, instead it will continue fetching and executing instructions until an interrupting condition occurs. For normal SCSI SCRIPTS™ operation, this bit should be 0. To (re)start the SIOP in this mode, read the **ISTAT** and **DSTAT** registers to clear the SCRIPT Pipeline Interrupt, and then set the START DMA bit in this register.

Bit 3 LLM Enable SCSI Low-Level Mode

Writing this bit to 1 will place the SIOP in the Low-Level Mode. In this mode, no DMA operations can occur, and no instructions can be executed. The Arbitration and Selection Modes can be executed by writing 1 to the Start Sequence bit as described in the **SCNTL0** Register. SCSI bus transfers can be performed manually by asserting and polling SCSI signals. Writing a 0 to this register allows instructions to be executed in either SCSI SCRIPTS™ mode.

Bit 2 STD Start DMA Operation

Setting this bit to 1 causes the SIOP to fetch a SCSI SCRIPT™ instruction from the address contained in the **DNAD** Register. 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

When the SIOP is in Manual Start Mode or Pipeline Mode, once the Start DMA bit is written to 1, this bit does not need to be written to 1 again until an interrupt occurs. When the SIOP is in Single Step Mode, the Start DMA bit needs to be written to 1 to start execution of each instruction.

Bit 1 RES Reserved

Bit 0 RST Software Reset

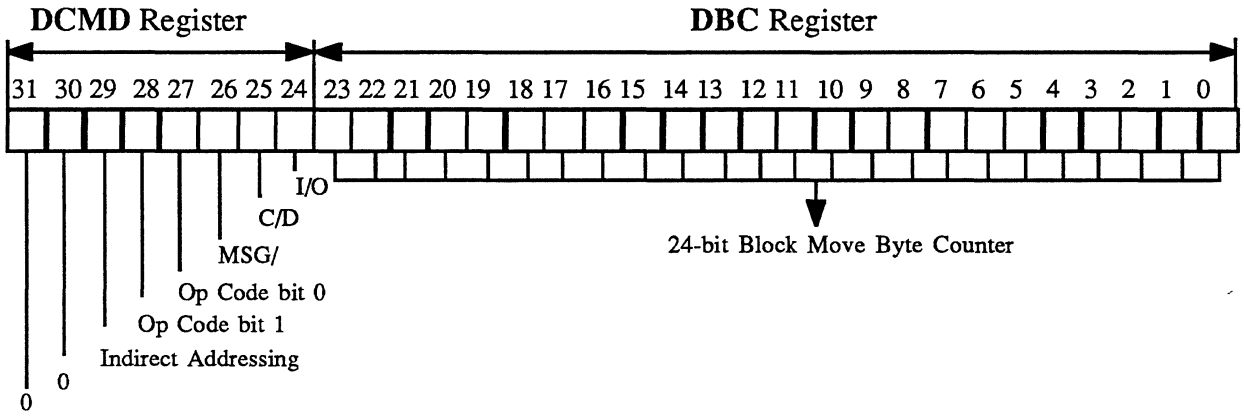
Writing this bit to 1 will reset the SIOP. All registers will be cleared to their respective default values and all SCSI signals will be deasserted. Writing this bit to 1 does not cause the SCSI RST/ signal to become asserted. This bit is not self-clearing and must be written to 0 in order to clear the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

5.0 Instruction Set of the SCSI I/O Processor

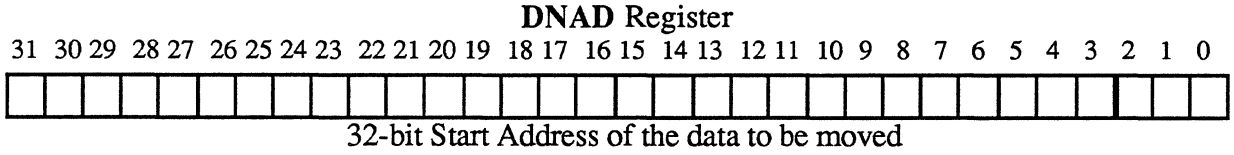
The SCSI I/O Processor fetches and executes its own instructions by becoming a bus master, and loading two 32-bit words into its registers. This method is referred to as executing SCSI SCRIPTS™. The SCSI SCRIPTS™ mode of executing instructions allows SIOP to make decisions based on the status of the SCSI bus. It also off-loads the microprocessor from having to service numerous interrupts. Instructions can also be executed by pipelining them in one at a time using pipeline mode.

There are four types of instructions implemented. Each instruction consists of two 32-bit words.

Block Move Instructions



First 32-bit word of the Block Move instruction



Second 32-bit word of the Block Move instruction

Indirect Addressing Field

If this Indirect bit is 0, then the 32-bit Address field points to the starting address of the data to be moved to/from memory. If this Indirect bit is 1, data is not moved to/from the address specified in the 32-bit Address field. The value contained in the memory location pointed to by the **DNAD** Register will be the actual starting address for data to be moved to/from memory. The SIOP will fetch the data from the address pointed to by the **DNAD** Register. This data will be loaded into the **DNAD** Register and this data will be the starting address for the data to be moved to/from memory. If the Indirect bit is 1, the SIOP will fetch the starting address described above before executing the instruction.

Op Code Field

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 will fetch the starting address from the location pointed to by the **DNAD** Register, and store that value 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 will assert 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. If the SCSI Group Code is either Group 0, Group 1, Group 2, or Group 5, then the SIOP will overwrite the **DBC** Register with the length of the Command Descriptor Block - 6,10, or 12 bytes. 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. If the Group code is not one of the Group codes defined above and the **DBC** Register contains 000000h, then an illegal instruction Interrupt will be generated.

(5) The SIOP will transfer 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 can optionally be 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** - Block Move Instruction
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

MOVE Instruction

(1) If the Indirect Addressing bit is 1, the SIOP will fetch the starting address from the location pointed to by the **DNAD** Register, and store that value 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 starting to execute this instruction.

(3) 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.

(4) 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.

(5) If the SCSI phase bits do not match the value stored in the **SSTAT2** Register, the SIOP will generate a phase mismatch interrupt and the command will not be executed.

WMOV Instruction

(1) If the Indirect Addressing bit is 1, the SIOP will fetch the starting address from the location pointed to by the **DNAD** Register, and store that value 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 starting to execute this instruction.

(3) The SIOP will wait for a previously unserviced phase to occur. A previously unserviced phase is defined to be any phase with REQ/ asserted. It simply means that the SIOP has not yet 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 will generate a phase mismatch interrupt and the command will not be executed.

The difference between the Block Move instruction and the Wait Block Move instruction is very subtle. Any time a Block Move instruction is executed, the Phase Field in the **DCMD** Register will be compared with the Latched SCSI phase lines stored in the **SSTAT2** Register. If REQ/ has not yet been asserted by the target for the next information transfer phase, then the value in the Latched SCSI phase lines bits will be the last serviced phase. When the Block Move instruction is executed, the phase defined in the **DCMD** Register may not be identical to the phase stored in the **SSTAT2** Register and a Phase Mismatch Interrupt will be generated. If the Wait Block Move instruction had been executed instead of the Block Move instruction, then the SIOP would have waited for the next information transfer phase to occur before comparing for the proper phase. There is no performance penalty when the SIOP is executing a Wait Block Move instruction compared with the Block Move instruction. If the phase has been qualified by REQ/ when the SIOP starts executing the instruction, both the Wait Block Move and Block Move instructions execute at the same speed. Therefore it is recommended that the Wait Block Move instruction be used primarily.

Phase Field (MSG, C/D, & I/O)

This three bit field defines the desired SCSI information transfer phase. When the SIOP is operating in initiator mode, these bits are compared with the Latched SCSI phase bits in the **SSTAT2** Register. When the SIOP is operating in target mode, the SIOP will assert the phase defined in this field. The following table describes the possible combinations and their corresponding SCSI 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

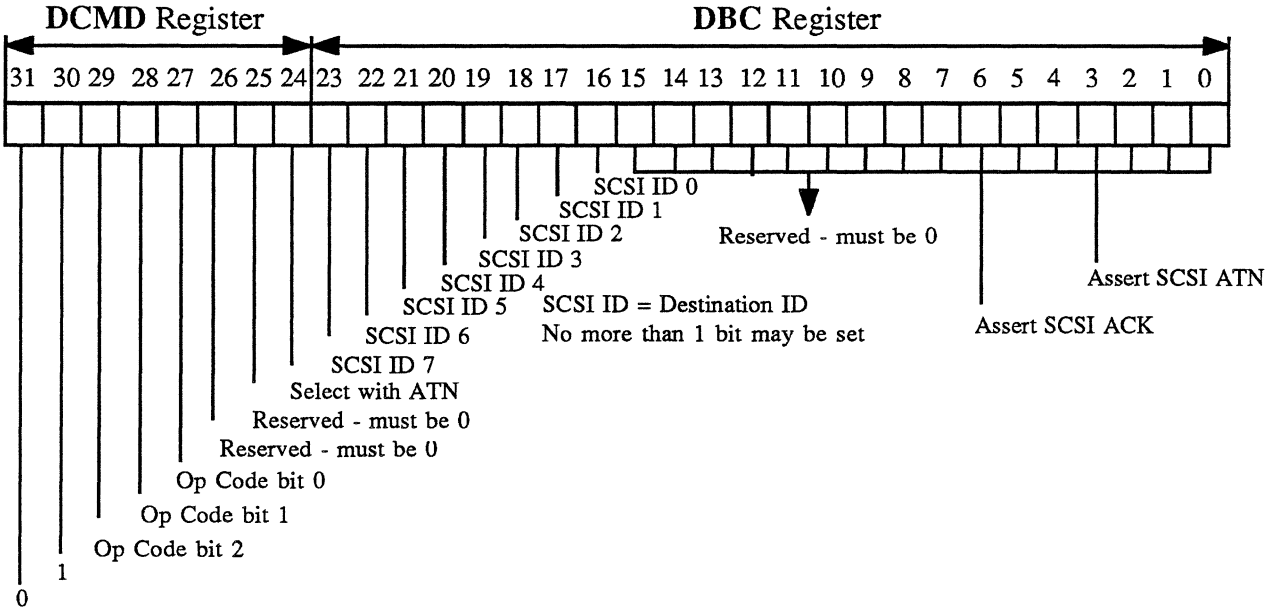
Transfer Counter Field

This twenty-four bit field specifies the number of data bytes to be moved between the SIOP and system memory. This 24-bit 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 **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 will fetch the next instruction. Once the SIOP has started executing SCSI SCRIPTS™ instructions, the user should not write to the **DBC** Register .

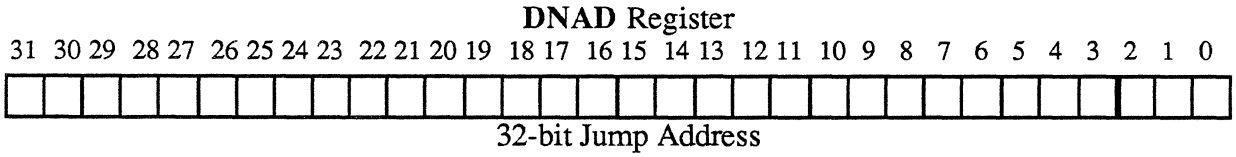
Start Address Field

This 32-bit field specifies the starting address of the data to be moved to/from memory. This 32-bit 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, the user should not write to the **DNAD** Register .

I/O Instructions



First 32-bit word of the I/O instruction



Second 32-bit word of the I/O instruction

Op Code Field

This three bit field is used to specify the event required to develop before execution continues. The Op Code Field bits have different meaning depending on whether the SIOP is operating in initiator or target mode.

Target Mode

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 will arbitrate for the SCSI bus by asserting the SCSI ID stored in the **SCID** Register. If the SIOP loses arbitration, then it will try 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 will fetches the next instruction from the address pointed to by the 32-bit jump address field stored in the **DNAD** Register. The SIOP automatically configures itself to be in the initiator mode if it was reselected, or the target mode if it was selected.

DISCONNECT Instruction

(1) The SIOP disconnects from the SCSI bus by deasserting all SCSI signal outputs. The SCSI direction control signals are deasserted causing the differential pair output drivers to become disabled.

WAIT SELECT Instruction

(1) If the SIOP has already been 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 address pointed to by the 32-bit jump address field stored in the **DNAD** Register. The SIOP automatically configures itself to be in 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. This instruction should not be used 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. This instruction should not be used in target mode.

Initiator Mode

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

(2) The SIOP arbitrates for the SCSI bus by asserting the SCSI ID stored in the **SCID** Register. If the SIOP loses arbitration, it will try again during the next available arbitration cycle without reporting any lost arbitration status.

(3) If the SIOP wins arbitration, it attempts to select 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.

(4) If the SIOP is selected or reselected before winning arbitration, it fetches the next instruction from the address pointed to by the 32-bit jump address field stored in the **DNAD** Register. The SIOP will automatically configure itself to be in initiator mode if it was reselected, or in target mode if it was selected.

(5) 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 is defined to be when BSY/ and SEL/ are inactive for a minimum of a Bus Free Delay (400 ns) after the SIOP has received a Disconnect Message or a Command Complete Message.

WAIT RESELECT Instruction

(1) If the SIOP is selected before being reselected, it fetches the next instruction from the address pointed to by the 32-bit jump address field stored in the **DNAD** Register. The SIOP automatically configures itself to be in 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

This bit specifies whether ATN/ is asserted during the selection phase when the SIOP is executing a **SELECT** instruction. It should be set to 1 only for the **SELECT** instruction when operating in initiator mode. If this bit is set to 1 on *any* other I/O instruction, an illegal instruction interrupt will be generated.

SCSI Destination ID Field

This eight bit field specifies the destination SCSI ID for an I/O instruction. Only one bit in this field should be set to 1.

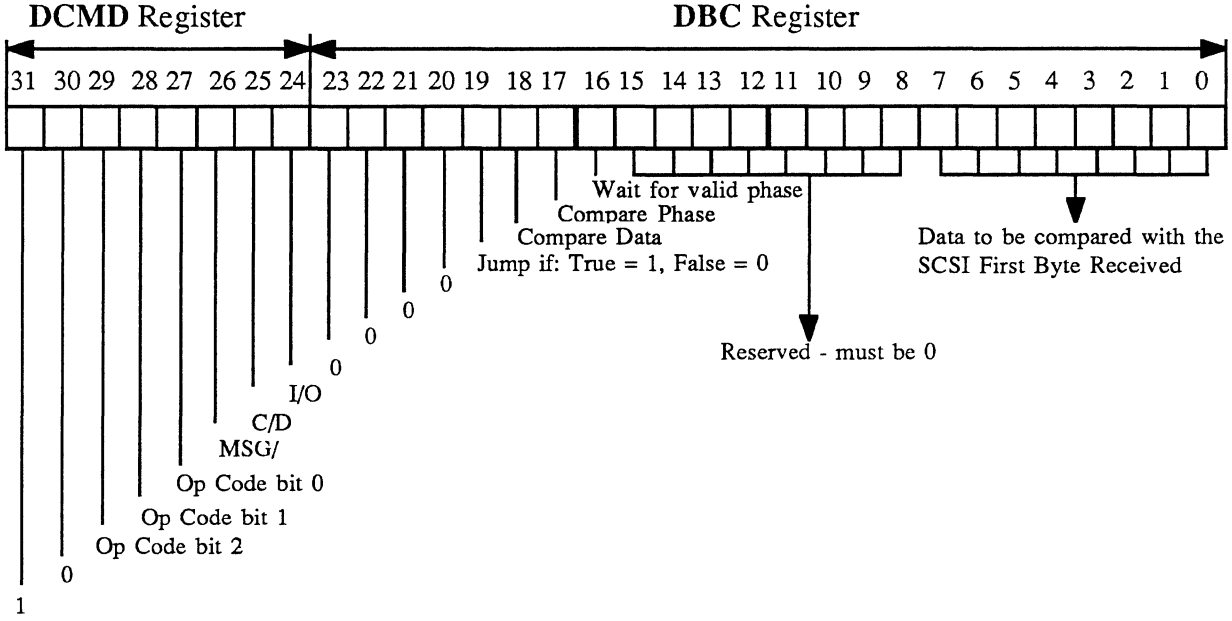
Assert ACK/ and Assert ATN/ Field

This two bit field specifies which of the two SCSI signals to assert or deassert. Writing either of these bits to 1 will cause the SIOP to set or reset the corresponding bits in the **SOCL** Register. The **SET** instruction can be used to assert ACK/ and/or ATN/ on the SCSI bus. The **CLEAR** instruction can be used to deassert ACK/ and/or ATN/ on the SCSI bus. 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.

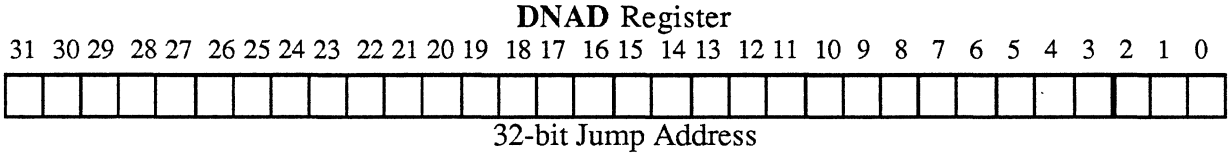
Jump Address Field

This thirty-two bit field specifies the address of the instruction to be fetched when the SIOP encounters a jump condition. The SIOP will fetch 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, in initiator mode during the execution of a **SELECT** instruction, if the SIOP is reselected instead, the next instruction is fetched from the address pointed to by this 32-bit jump address field. For a complete description of the different jump conditions, reference each instruction description.

Transfer Control Instructions



First 32-bit word of the Transfer Control instructions



Second 32-bit word of the Transfer Control instructions

Op Code Field

This three bit 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 can operate in initiator or target mode.

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

(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(depending on the Jump if True/False bit), the SIOP loads the **DSP** Register with the contents of the **DNAD** Register. The **DSP** register now contains the address of the next instruction.

(2) If the comparisons are false(depending on the Jump if True/False bit), 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 loads the **DSP** Register with the contents of the **DNAD** Register and that address value becomes the address of the next instruction. When the SIOP executes a **CALL** instruction, the current instruction pointer contained in the **DSP** Register is stored in the **TEMP** Register. 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 and the instruction pointer is not modified.

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 will load the **DSP** Register with the contents of the **DNAD** Register and that address value becomes the address of the next instruction. When the SIOP executes a **CALL** instruction, the current instruction pointer contained in the **DSP** Register will be stored in the **TEMP** Register. When a **RETURN** instruction is executed, then the value stored in the **TEMP** Register will be returned to the **DSP** Register. The SIOP does not check to see that a **CALL** instruction has already been executed and 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 will fetch the next instruction from the address pointed to by the **DSP** Register and the instruction pointer will not be modified.

INT Instruction

(1) The SIOP will compare 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 will generate an interrupt by asserting the IRQ/ signal. The 32-bit address field that is stored in the **DNAD** Register can contain a unique interrupt service vector. When servicing the interrupt, this unique status code can allow the ISR to quickly identify the point at which the interrupt occurred.

(2) The SIOP will be halted and the **DSP** Register must be written to start any further operation.

Phase Field

This three bit field corresponds to the three SCSI bus phase signals and is used to compare 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 reset to 0.

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

This field describes whether the SIOP should branch when a comparison is true or when a comparison is false. If this bit is 1, the SIOP executes the Transfer Control instruction (**JUMP, CALL, RETURN,** or **INT**) when the comparison is true. 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 and the instruction pointer does not change. If this bit is 1 and the comparison is true, the SIOP fetches the next instruction from the address pointed to by the **DNAD** Register and the instruction pointer contains this new address. If this bit is 0, the SIOP executes the Transfer Control instruction (**JUMP, CALL, RETURN,** or **INT**) when the comparison is false. If this bit is 0 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 fetches the next instruction from the address pointed to by the **DSP** Register. This bit applies to both Phase Compares and Data Compares. If both the Phase Compare and Data Compare bits are set to 1, both compares must be true to branch on a true condition, and both compares must be false to branch on a false condition.

Compare Data

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

Compare Phase

In initiator mode, this field is used to enable a compare of the SCSI phase. If 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, the comparison is true. In target mode, if this bit is 1, this field is used to test for an active SCSI ATN/ signal. The Wait for a valid phase controls when the compare will occur. The Jump if True/False bit determines the condition(true or false) on which to branch.

Wait for Valid Phase

The Wait for a valid phase controls when a compare will occur. If the Wait for valid Phase bit is set to 1, the SIOP waits for the next valid phase before comparing the phase and/or data. If the Wait for valid Phase bit is 0, the SIOP compares immediately.

Data to be Compared Field

This 8-bit field is used as the data to be compared with the SCSI First Byte Received Register. This bit should be used with the Compare Data Field in order to compare for a particular data pattern.

Jump Address Field

This 32-bit field is used as the address of the next instruction when the compare operations are successful. For example, if a **JUMP** instruction is issued and the Compare Data & True/False bits are 1, then the SIOP will fetch the next instruction from this 32-bit address if the SCSI First Byte Received is equal to the Data to be Compared Field. Once the SIOP has fetched the instruction from the address pointed to by these 32-bits, this address is incremented by 4, loaded into the **DSP** Register and becomes the current instruction pointer.

6.0 Functional Description

6.1 SCSI SCRIPTS™ Mode

To start the SIOP in SCSI SCRIPTS™ Mode, the first step is to load the **DSP** Register with the address location that contains the first SCSI SCRIPTS™ instruction. The SIOP will fetch the first instruction from the address pointed to by the **DSP** Register. Once the instruction has been received, then the **DSP** Register will be incremented by 8 so as to point to the next SCSI SCRIPTS™ address. It will continue to fetch and execute the instructions from system memory until either an interrupt condition occurs or the Interrupt instruction is executed. Once an interrupt is generated, the SIOP will halt all operations until the interrupt is serviced. Once the SIOP has halted, the **DSP** Register needs to be written with the address of the next instruction in order to restart the automatic fetch and execution of the instructions.

6.2 Loopback Mode

The Loopback Mode of the SIOP allows the user to test both initiator and target operation. When the Loopback Enable bit is 1 in the **CTEST4** Register, the SIOP will allow the user to control *all* SCSI signals whether the SIOP is operating in initiator or target mode. The steps needed to implement the loopback function are described below.

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) To respond 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) Assert the desired phase by writing the MSG/, C/D, and I/O bits to the desired phase in the **SOCL** Register.
9) Assert REQ/ by keeping the phase bits the same and writing the REQ/ bit to 1 in the **SOCL** Register. By asserting REQ/ after asserting the phase signals, the 400 ns Bus Settle Delay before asserting REQ/ is accommodated.
10) The initiator role can be implemented by SCSI SCRIPTS™. The next step is to issue a Block Move instruction for the appropriate phase.

The above description describes the method of performing selection in the SCSI Loopback Mode. The sequences that can be tested here are too numerous to list. The basic flow that needs to be followed here is for the SIOP to execute initiator instructions, and the user to implement the target role by asserting the appropriate SCSI signals and polling for the appropriate SCSI signals.

6.3 Parity Options

The SIOP implements a flexible parity scheme that allows the user to control 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 the SCSI ATN/ signal when it detects a parity error when the SIOP is operating as an initiator.
2) Enable Parity Generation - Bit 2 in the **SCNTL0** Register
   This bit controls whether the SIOP will generate parity sent to the SCSI bus or allow 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 controls whether the SIOP will check for parity errors. It will check for odd or even SCSI 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 controls whether the SIOP will check for and assert even or odd parity on the SCSI bus.
5) Disable Halt on ATN/ or a Parity Error - Target Mode Only - Bit in **SXFER** Register
   This bit controls whether 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 control whether 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/). It is 1 when SDP/ is active.
9) Latched SCSI Parity Signal - Bit 3 in the **SSTAT2** Register
This status bit represents the parity signal (SDP/) when the First Byte Received is latched in the chip for a particular phase. It is 1 when SDP/ is active.
10) DMA FIFO Parity bit - Bit 3 in the **CTEST2** Register
This status bit is represents the parity bit in the DMA FIFO when 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 when 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 when data is read from the FIFO by reading the **CTEST3** Register.
13) Even Parity bit - Bit 2 in the **CTEST7** register
This bit inverts SCSI parity for the host bus, allowing the SIOP to transmit, receive and verify odd SCSI parity and transmit/receive even parity from the system bus.

Parity Control

EPG	EPC	ASEP	Description
0	0	0	Will not check for parity errors - Parity flows from DP3-DP0 thru chip to SCSI bus when sending SCSI data, Parity flows from SCSI bus to DP3-DP0 when receiving SCSI data - Asserts odd parity when sending SCSI data
0	0	1	Will not check for parity errors - Parity flows from DP3-DP0 thru chip to SCSI bus when sending SCSI data, Parity flows from SCSI bus to DP3-DP0 when receiving SCSI data - Asserts even parity when sending SCSI data
0	1	0	Checks for odd parity on both SCSI data received and system data when sending - Parity flows from DP3-DP0 thru chip to SCSI bus when sending SCSI data, Parity flows from SCSI bus to DP3-DP0 when receiving SCSI data - Asserts odd parity when sending SCSI data
0	1	1	Checks for even parity on both SCSI data received and system data when sending - Parity flows from DP3-DP0 thru chip to SCSI bus when sending SCSI data, Parity flows from SCSI bus to DP3-DP0 when receiving SCSI data - Asserts even parity when sending SCSI data
1	0	0	Will not check for parity errors - Parity on DP3-DP0 is ignored, parity is generated when sending SCSI data, Parity flows from SCSI bus to chip but is not asserted on DP3-DP0 when receiving SCSI data - Asserts odd parity when sending SCSI data
1	0	1	Will not check for parity errors - Parity on DP3-DP0 is ignored, parity is generated when sending SCSI data, Parity flows from SCSI bus to chip but is not asserted on DP3-DP0 when receiving SCSI data - Asserts even parity when sending SCSI data
1	1	0	Checks for odd parity on both SCSI data received and system data when sending - Parity on DP3-DP0 is ignored, parity is generated when sending SCSI data, Parity flows from SCSI bus to chip but is not asserted on DP3-DP0 when receiving SCSI data - Asserts odd parity when sending SCSI data
1	1	1	Checks for even parity on both SCSI data received and system data when sending - Parity on DP3-DP0 is ignored, parity is generated when sending SCSI data, Parity flows from SCSI bus to chip but is not asserted on DP3-DP0 when receiving SCSI data - Asserts even parity when sending SCSI data

EPG = Enable Parity Generation EPC = Enable Parity Checking ASEP = Assert SCSI Even Parity

Parity Errors & Interrupts

This table describes the options available when an parity error does occur. This table only applies to the case where the Enable Parity Checking bit is 1.

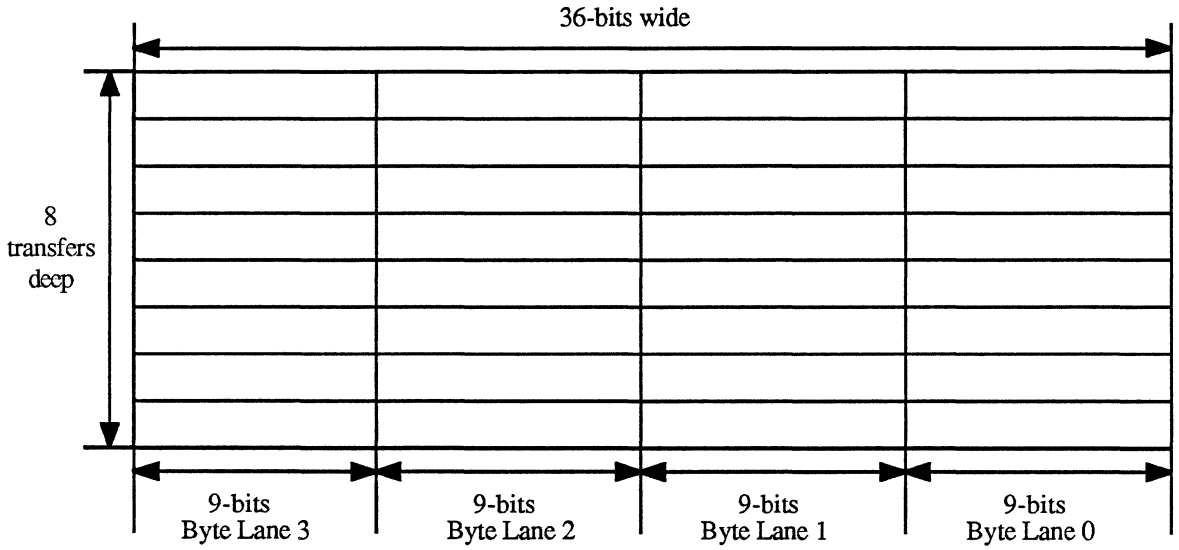
DHP	EPI	Description
0	0	Will not halt when a parity error occurs in target or initiator mode
0	1	Will Interrupt when a parity error occurs in target or initiator mode
1	0	Will halt when a parity error occurs in target mode, will not generate an interrupt
1	1	Will halt when a parity error occurs in target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

DHP = Disable Halt on ATN/ or a Parity Error EPI = Enable Parity Interrupt

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 up into 4 sections each being 9-bits wide and 8 transfers deep. Each of these four sections are labeled "byte lanes." Each can be individually tested by writing known data into the FIFO and reading that same data back out of the FIFO.



In order to write data into the DMA FIFO, data must be loaded 9 bits per instruction. When writing data to the FIFO, the data is written to the top of the FIFO. When reading data from the FIFO, the data is read from the bottom of the FIFO. Three control bits in the CTEST4 Register allow the user to access any one of the four "byte lanes." Parity is written to the FIFO through bit 3 of the CTEST7 Register. This bit should be set to the desired value before each write operation to the FIFO.

In order to choose the appropriate "byte lane", the following three bits should be written according to the table shown below.

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

Steps to follow to completely load the DMA FIFO with known data and be able to read back the data parity.

1) Setup access to Byte Lane 0 in the DMA FIFO by writing an 04h to the CTEST4 Register

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

2) Write the desired data pattern (incrementing data pattern) to the DMA FIFO by writing the parity bit to bit 3 of the CTEST7 Register and the 8-bit data value to the CTEST6 Register.

```
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) To read data back out of the FIFO read the CTEST6 Register, then read the parity bit in the CTEST2 Register.

```
for (i=0; i<8; ++i)
        {
        byte_lane0 [i] = inportb (CTEST6);          /* read back in data out of the FIFO */
        parity0 [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

The SCSI Synchronous data FIFO can be loaded with data by any microprocessor through the use of the SCSI FIFO Write Enable bit in the CTEST4 Register. Writing this bit to 1 allows the user the capability of loading the SCSI FIFO with a known data pattern. The data is loaded into the SCSI FIFO by writing the SODL Register. To read data back out of the SCSI FIFO, the microprocessor should read the CTEST3 Register. To check the parity when reading data out of the FIFO, read the SCSI FIFO parity bit, bit 4 in the CTEST2 register after reading CTEST3. The parity bit is stored in the CTEST2 Register during a CTEST3 Register read.

Parity can be written to the FIFO in one of three ways. Parity can flow into the SIOP on the parity signals if the Enable Parity Generation bit in the SCNTL0 Register is equal to 0. The parity signal should be driven by the microprocessor on the appropriate parity signal for the corresponding 8-bit data signals. For example, if the FIFO is being written to the SIOP on "Byte Lane 2" (D23-D16), then the parity information should be driven on DP2.

If the Parity Generation bit is equal to 1, then the SIOP will force the parity bit to reflect even or odd parity. In order to load the SCSI FIFO with odd parity, the Assert Even SCSI parity bit in the SCNTL1 Register should be equal to 0. If this bit is equal to 1, then the SCSI FIFO will be loaded with even parity.

Steps to follow 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

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) Enable the SCSI FIFO to accept data by writing the SCSI FIFO Write Enable bit to 1 in the CTEST4 Register. This bit must be rewritten to 0, when the FIFO test is complete.

```
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   */
        }
```

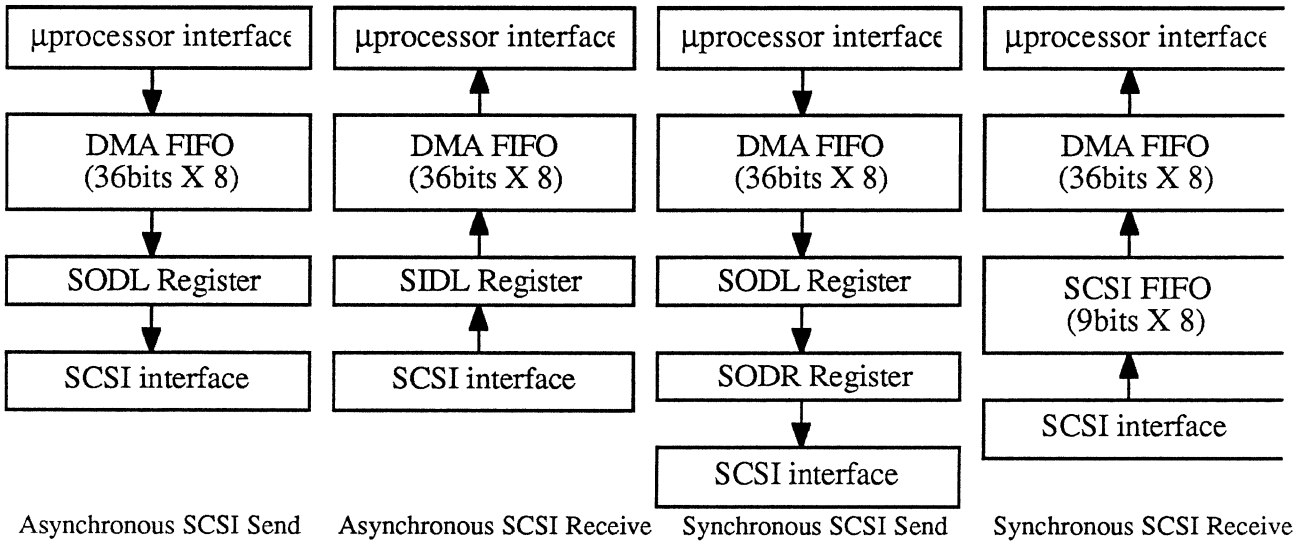
5) Reset the SCSI FIFO Write Enable bit.

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

6.5 80386 Interface

Data Paths of the 53C700 SIOP

The data path through the SIOP are dependent on whether data is being moved in or out of the chip, and whether SCSI data is being sent asynchronously or synchronously. The diagrams below show how data is moved to/from the SCSI bus in each of the different modes.



Asynchronous SCSI Send | Asynchronous SCSI Receive | Synchronous SCSI Send | Synchronous SCSI Receive

In order to determine if any bytes remain in the data path when the chip halts any operation, the following steps should be taken.

Asynchronous SCSI Send - Initiator and Target operation

1) To determine if any bytes are remaining in the DMA FIFO, use the algorithm described in the DFIFO Register description.
2) To determine if any bytes are remaining in the SODL Register, read the SSTAT1 Register and examine bit 6. If bit 6 is equal to 1, then there is a byte in the SODL Register.

Synchronous SCSI Send - Initiator and Target operation

1) To determine if any bytes are remaining in the DMA FIFO, use the algorithm described in the DFIFO Register description.
2) To determine if any bytes are remaining in the SODL Register, read the SSTAT1 Register and examine bit 6. If bit 6 is equal to 1, then there is a byte in the SODL Register.
3) To determine if any bytes are remaining in the SODR Register, read the SSTAT1 Register and examine bit 5. If bit 5 is equal to 1, then there is a byte in the SODR Register.

Asynchronous SCSI Receive - Initiator and Target operation

1) To determine if any bytes are remaining in the DMA FIFO, use the algorithm described in the DFIFO Register description.
2) To determine if any bytes are remaining in the SIDL Register, read the SSTAT1 Register and examine bit 7. If bit 7 is equal to 1, then there is a byte in the SIDL Register.

Synchronous SCSI Receive - Initiator and Target operation

1) To determine if any bytes are remaining in the DMA FIFO, use the algorithm described in the DFIFO Register description.
2) To determine if any bytes are remaining in the SCSI FIFO, read the SSTAT2 Register and examine bits 7 - 4 which is a binary representation of the number of valid data bytes residing in the SCSI FIFO.

Starting Block Move Operations at an "Odd-Byte Boundary"

The SIOP can transfer data 16-bits per transfer or 32-bit per transfer. The two bits that control 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.

BW16	286M	Description
0	0	32-Bit Data Transfers, SIOP asserts and expects 80386 signals
X	1	16-Bit Data Transfers, SIOP asserts and expects 80286 signals
1	0	16-Bit Data Transfers, SIOP asserts and expects 80386 signals

BW16 = Bus Width 16 bit in the DMODE Register X = Don't Care
286M = 80286 Mode bit in the DMODE Register

The SIOP has optimized the procedure of moving data to an odd-byte boundary address when operating as a bus master. An odd-byte boundary is defined to be an address with A0 = 1 in 16-bit operation, and an address with either A1 = 1 or A0 = 1 in 32-bit operation.

16-Bit Data Transfers - 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 bit 0 = 1, then the very first Block Move operation involves a 1 byte

transfer on D15 - D8, DP1. Each succesive transfer will occur on D15 - D0, DP1 - DP0. 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.

80286 Mode

	A0	BHE/
Even-byte start address	0	Asserted for the first transfer
Odd-byte start address	1	Not asserted for the first transfer
Even-byte finish address	0	Asserted for the last transfer
Odd-byte finish address	1	Not asserted for the last transfer

80386 Mode

	A0	BE1/	BE0/		
Even-byte start address	0	0	0	0 = asserted	1 = deasserted for BE1/ - BE0/
Odd-byte start address	1	0	1	1 = asserted	0 = deasserted for A0
Even-byte finish address	0	0	0		
Odd-byte finish address	1	0	1		

32-Bit Data Transfers

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 bit 0 = 1, and the Byte Counter Value stored in the DBC Register is greater than three, then the very first Block Move operation involves a 3 byte transfer on D31 - D8, DP3 - DP1 with BE3/, BE2/, and BE1/ all driven active. Each succesive transfer will occur on D31 - D0, DP3 - DP0 with BE3/ - BE0/ all 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 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 three bytes to transfer in order to complete a Block Move instruction, then those bytes will be transferred on D23 - D0, DP2 - DP0 with BE2/, BE1/, and BE0/ driven active.

80386 Mode

	A1	A0	BE3/	BE2/	BE1/	BE0/
Even-byte start address	0	0	0	0	0	0
Odd-byte start address	0	1	0	0	0	1
Odd-byte start address	1	0	0	0	1	1
Odd-byte start address	1	1	0	1	1	1
Even-byte finish address	0	0	0	0	0	0
Odd-byte finish address	0	1	0	0	0	1
Odd-byte finish address	1	0	0	0	1	1
Odd-byte finish address	1	1	0	1	1	1

0 = asserted 1 = deasserted for BE3/ - BE0/
1 = asserted 0 = deasserted for A1 - A0

Instruction Fetch Operation

To start the SIOP instruction fetch process, the address that contains the first SCSI SCRIPT™ should be written to the DSP Register. Once the DSP Register has been written with the first SCSI SCRIPT™ address, the SIOP will continue to fetch and execute its instructions by reading them from system memory. These SCSI SCRIPTS™ do not have to reside in system memory, and by decoding a certain address space, they could reside in a PROM. SCSI SCRIPTS™ must be stored 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. SCSI SCRIPTS™ instructions can be loaded in one of two ways: by fetching two 32-bit words, or by fetching four 16-bit words. The Scripts Loaded in 16-bit Mode bit in the DCNTL Register should be written 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. If this bit is 1, then the DC/ signal will be high 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 cache or only in memory. Usually only control (DC/ low) will reside in the cache, and data information (DC/ high) does not.

53C700 SIOP Bus Master Data Transfers

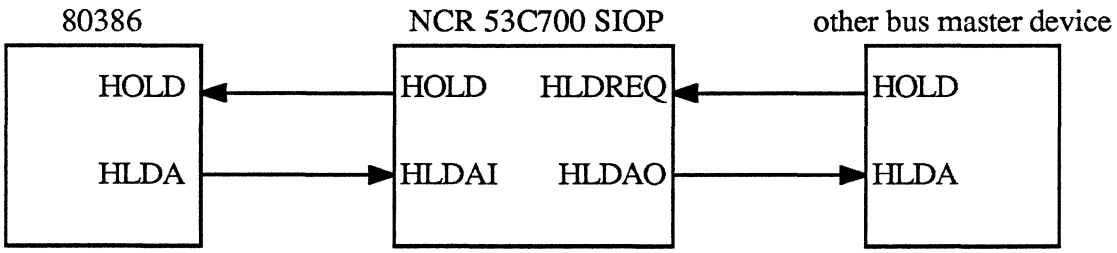
When the SIOP becomes bus master, it takes over control of the system bus and can transfer data in a variety of ways. The SIOP can transfer data to I/O addresses, memory addresses, or a fixed address. The following three bits determine the width and type of data transfer that will occur once the SIOP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

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

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

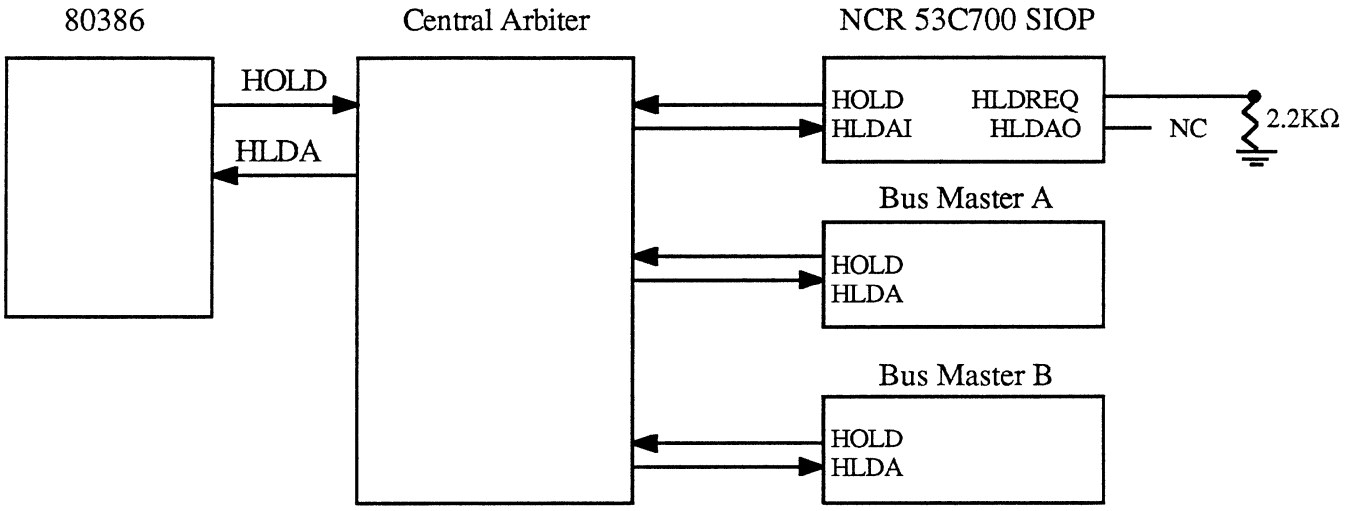
The SIOP will gain control of the system bus when it needs to do one of two operations: an instruction fetch, or a data transfer specified by a Block Move instruction. The SIOP gains requests control of the 80386 bus by asserting the HOLD output signal. When the HLDAI signal becomes driven active by 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 that the control of the bus is being granted to the device that asserted the HLDREQ signal. The diagram below shows an example of the daisy-chaining technique.



HOLD - HLDA Daisy-chaining scheme

If the SIOP and the other bus master device both request the bus at the same time, 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.

Another method of allowing bus master devices to gain access to the system bus is by having a central arbiter. This arbiter receives bus requests from various bus master devices and prioritizes the access according to some priority scheme. In this scheme, the HLDREQ and HLDAO signals are unused, and the SIOP only uses HOLD and HLDAI. If unused, HLDREQ should be tied to VSS directly or through a pull-down resistor. The diagram below shows an example of the central arbiter technique.



Central Arbiter scheme

6.6 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 does contain the open-drain output drivers that can be connected directly to the single-ended SCSI bus. Each output has been isolated from the VDD power supply to ensure that the SIOP will have 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 will provide the necessary biasing needed to "pull" the inactive signal to an appropriate inactive voltage level ~ 3.0 V. The terminators do not need to be present on every SCSI board, but the terminators do need to be installed at each end of the SCSI cable. Since the terminator location is completely a function of how the equipment is being set-up by the end user, most SCSI boards should provide a means of accomodating terminators. They should be socketed, so that if not needed, they may be removed. No system should ever have more than 2 sets of terminators installed and activated. If more than 2 sets of terminators active, then the impedance and inactive SCSI voltage levels may not be correct.

Being Selected or Reselected when trying to perform Selection

In multitasking or multithreaded SCSI I/O operations, it is common to become selected or reselected when trying to perform selection. This situation arises when a SCSI controller is operating in initiator mode and trying to select a target. The way that the SIOP handles this condition is very unique. If the SIOP is executing a SELECT instruction and it becomes selected as a target or reselected as an initiator, the SIOP will automatically configure itself to the appropriate mode. If the SIOP becomes reselected, then it will remain operating as an initiator, and it will fetch the next instruction from the address pointed by the DNAD Register which is the second 32-bit word of the SELECT instruction. If the SIOP becomes selected, then it will automatically change to the target role, and it will fetch the next instruction from the address pointed by the DNAD Register

Being Selected or Reselected when trying to perform Reselection

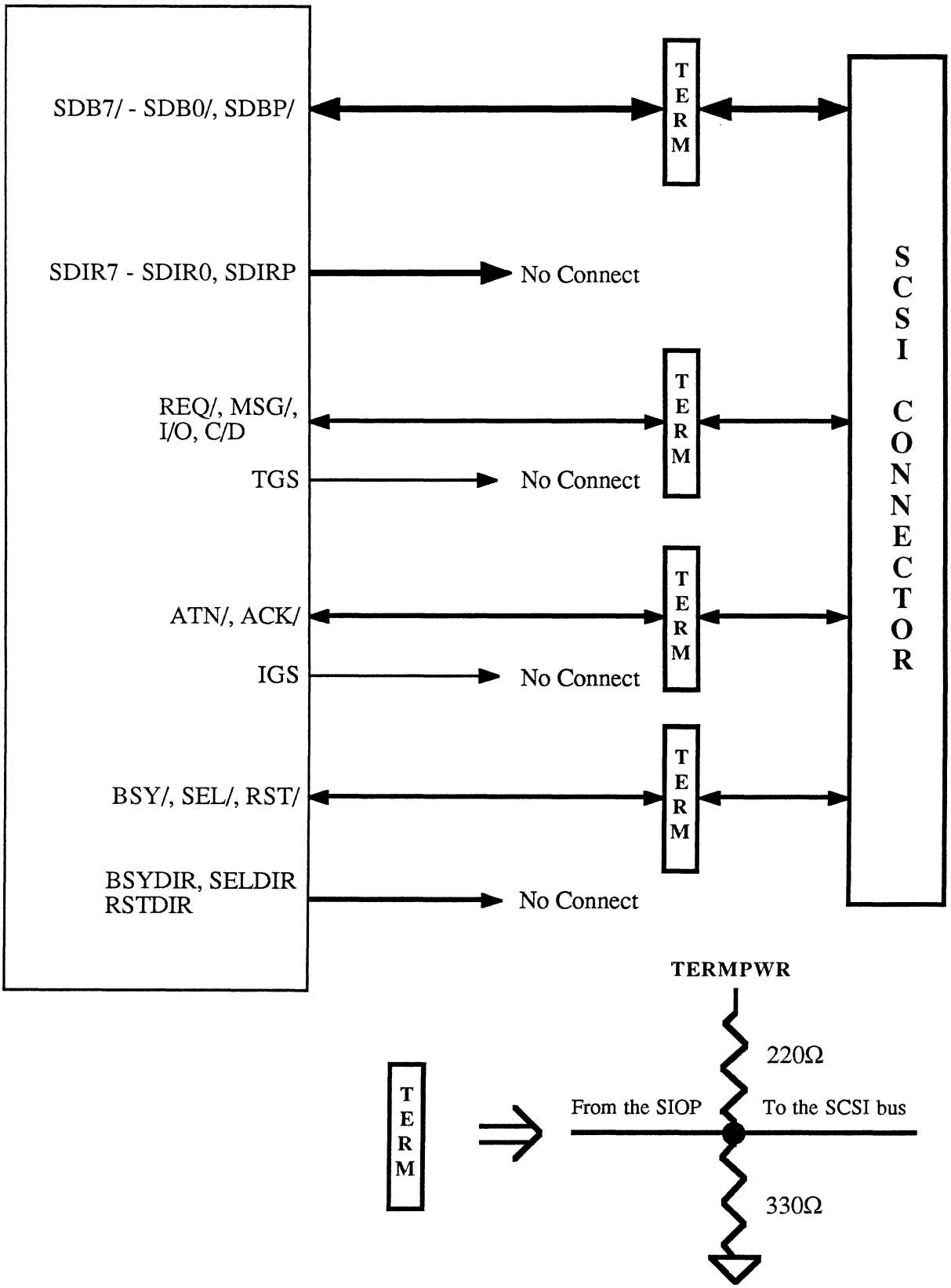
In multitasking or multithreaded SCSI I/O operations, it is also common to become selected or reselected when trying to perform a reselection. This situation arises when a SCSI controller is operating in target mode and trying to reselect an initiator. If the SIOP is executing a RESELECT instruction and it becomes selected as a target or reselected as an initiator, the SIOP will automatically configure itself to the appropriate mode. If the SIOP becomes reselected, then it will automatically change to the initiator role, and it will fetch the next instruction from the address pointed by the DNAD Register which is the second 32-bit word of the SELECT instruction. If the SIOP becomes selected, then it will remain in the target role, and it will fetch the next instruction from the address pointed by the DNAD Register

Synchronous Operation

The SIOP can transfer synchronous SCSI data in both initiator and target modes. The synchronous offset can be controlled by writing to bits 3-0 of the SXFER Register. The synchronous transfer period can be controlled by writing bits 6-4 of the SXFER Register. If bits 3-0 of the SXFER Register are equal to 0, then the SIOP will transfer SCSI data asynchronously. If bits 3-0 are not equal to 0, the SIOP will transfer data synchronously during data transfer phases. The SIOP can always receive synchronous data at a period of 200 ns. Therefore, when negotiating for synchronous SCSI data transfers, the transfer period suggested is 200 ns. The SIOP can send synchronous SCSI data at a period of 200 ns, depending on the input clock frequency. Even if the SIOP cannot send synchronous data at a period of 200 ns, it can still receive synchronous data at a period of 200 ns.

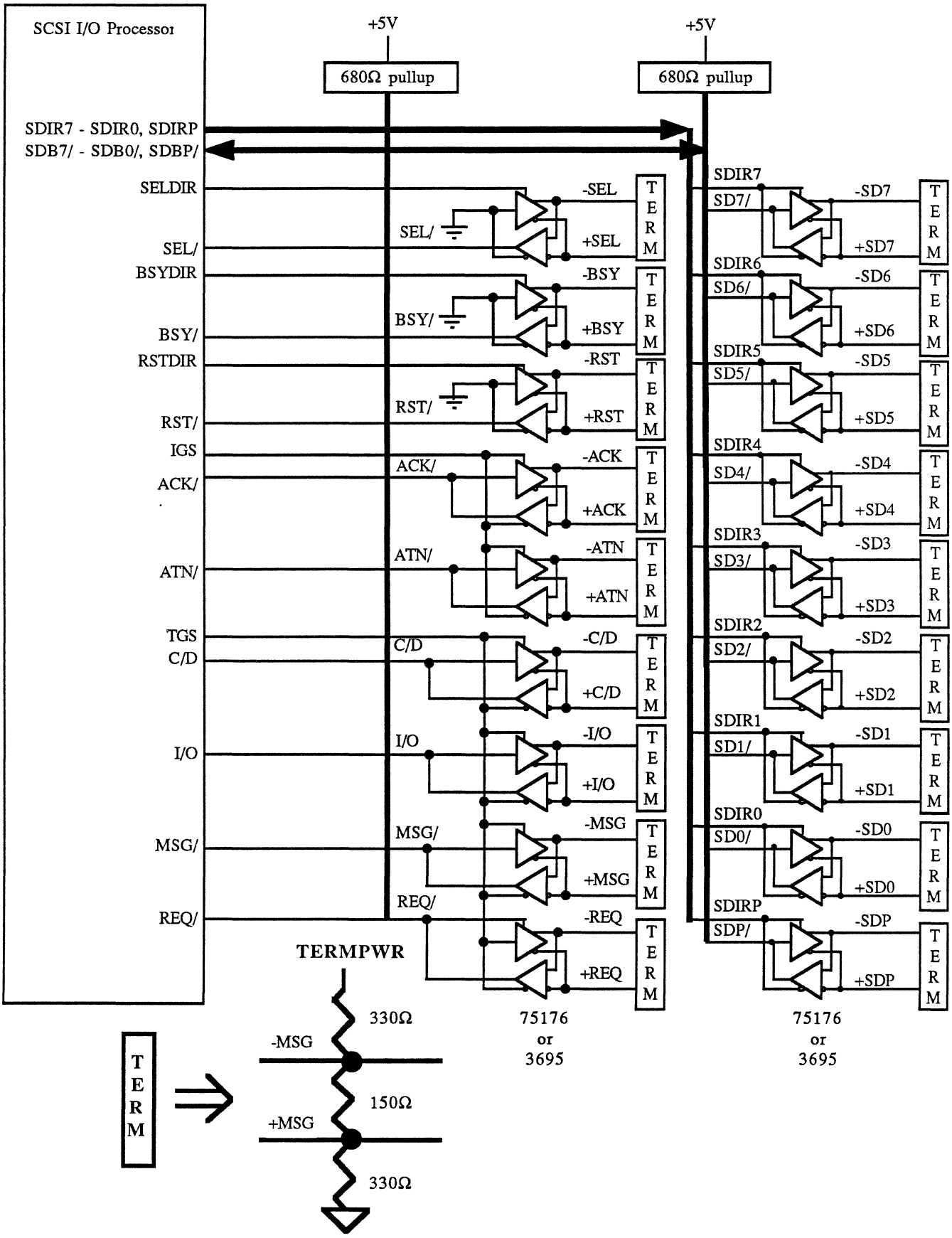
SCSI I/O Processor

Single - Ended Interface Diagram



SCSI I/O Processor

Differential Interface Diagram



D.C. Characteristics

Absolute Maximum Stress Ratings

SYMBOL	PARAMETER	MIN	MAX	UNIT
Tstg	Storage Temperature	-55	150	℃
VDD	Supply Voltage	-0.5	7.0	V
VIN	Input Voltage	VSS - 0.5	VDD + 0.5	V
ESD*	Electrostatic Discharge Sensitivity		10K	V

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

Operating Conditions

SYMBOL	PARAMETER	MIN	MAX	UNIT
VDD	Supply Voltage	4.75	5.25	V
IDD	Supply Current		TBD	mA
Ta	Operating Free-Air	0	70	℃

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

SYMBOL	PARAMETER	MIN	MAX	UNIT	CONDITIONS
VIH	Input High Voltage	2.0	VDD + 0.5	V	
VIL	Input Low Voltage	VSS - 0.5	0.8	V	
VOL	Output Low Voltage	VSS	0.5	V	IOL = 48 mA
VHYS	Hysteresis	200		mV	
IIN	Input Leakage Current	-10	10	µA	
IOZ	Output Leakage Current	-10	10	µA	

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

SYMBOL	PARAMETER	MIN	MAX	UNIT	CONDITIONS
VOH	Output High Voltage	2.4	VDD	V	IOH = -4 mA
VOL	Output Low Voltage	VSS	0.4	V	IOL = 4 mA
IOH	Output High Current	-2.0		mA	VOH = VDD - 0.5 V
IOL	Output Low Current	4.0		mA	VOL = 0.4 V
IOZ	Output Leakage Current	-10	10	µA	

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

SYMBOL	PARAMETER	MIN	MAX	UNIT	CONDITIONS
VIH	Input High Voltage	2.0	VDD + 0.5	V	
VIL	Input Low Voltage	VSS - 0.5	0.8	V	
IIN	Input Leakage Current	-10	10	µA	

CLK Input Signal

SYMBOL	PARAMETER	MIN	MAX	UNIT	CONDITIONS
VIH	Input High Voltage	3.85	VDD + 0.5	V	
VIL	Input Low Voltage	VSS - 0.5	1.65	V	
IIN	Input Leakage Current	-10	10	µA	

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

SYMBOL	PARAMETER	MIN	MAX	UNIT	CONDITIONS
VOH	Output High Voltage	2.4	VDD	V	IOH = -8 mA
VOL	Output Low Voltage	VSS	0.4	V	IOL = 8 mA
IOH	Output High Current	-4.0		mA	VOH = VDD - 0.5 V
IOL	Output Low Current	8.0		mA	VOL = 0.4 V
IOZ	Output Leakage Current	-10	10	µA	

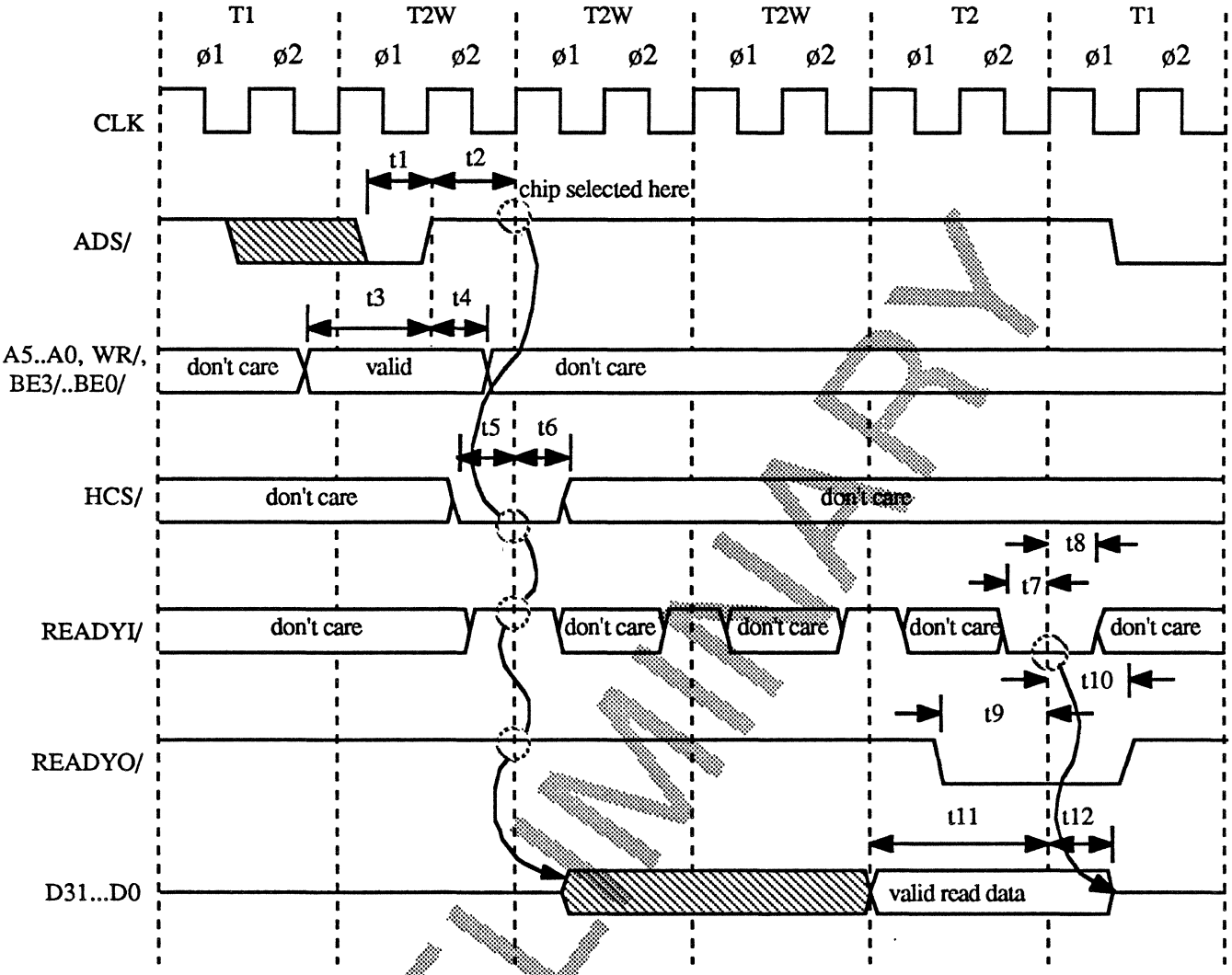
Output Signal - IRQ/

SYMBOL	PARAMETER	MIN	MAX	UNIT	CONDITIONS
VOL	Output Low Voltage	VSS	0.4	V	IOL = 8 mA
IOL	Output Low Current	8.0		mA	VOL = 0.4 V
IOZ	Output Leakage Current	-400	10	µA	

Bidirectional Signals - WR/, ADS, A31-A2, D31-D0, DP3_ABRT/, DP2-DP0, BE0/_A0, BE1/_A1, BE2/_BHE/, BE3/

SYMBOL	PARAMETER	MIN	MAX	UNIT	CONDITIONS
VIH	Input High Voltage	2.0	VDD + 0.5	V	
VIL	Input Low Voltage	VSS - 0.5	0.8	V	
VOH	Output High Voltage	2.4	VDD	V	IOH = -8 mA
VOL	Output Low Voltage	VSS	0.5	V	IOL = 8 mA
IOH	Output High Current	-4.0		mA	VOH = VDD - 0.5 V
IOL	Output Low Current	8.0		mA	VOL = 0.4 V
IIN	Input Leakage Current	-10	10	µA	
IOZ	Output Leakage Current	-10	10	µA	

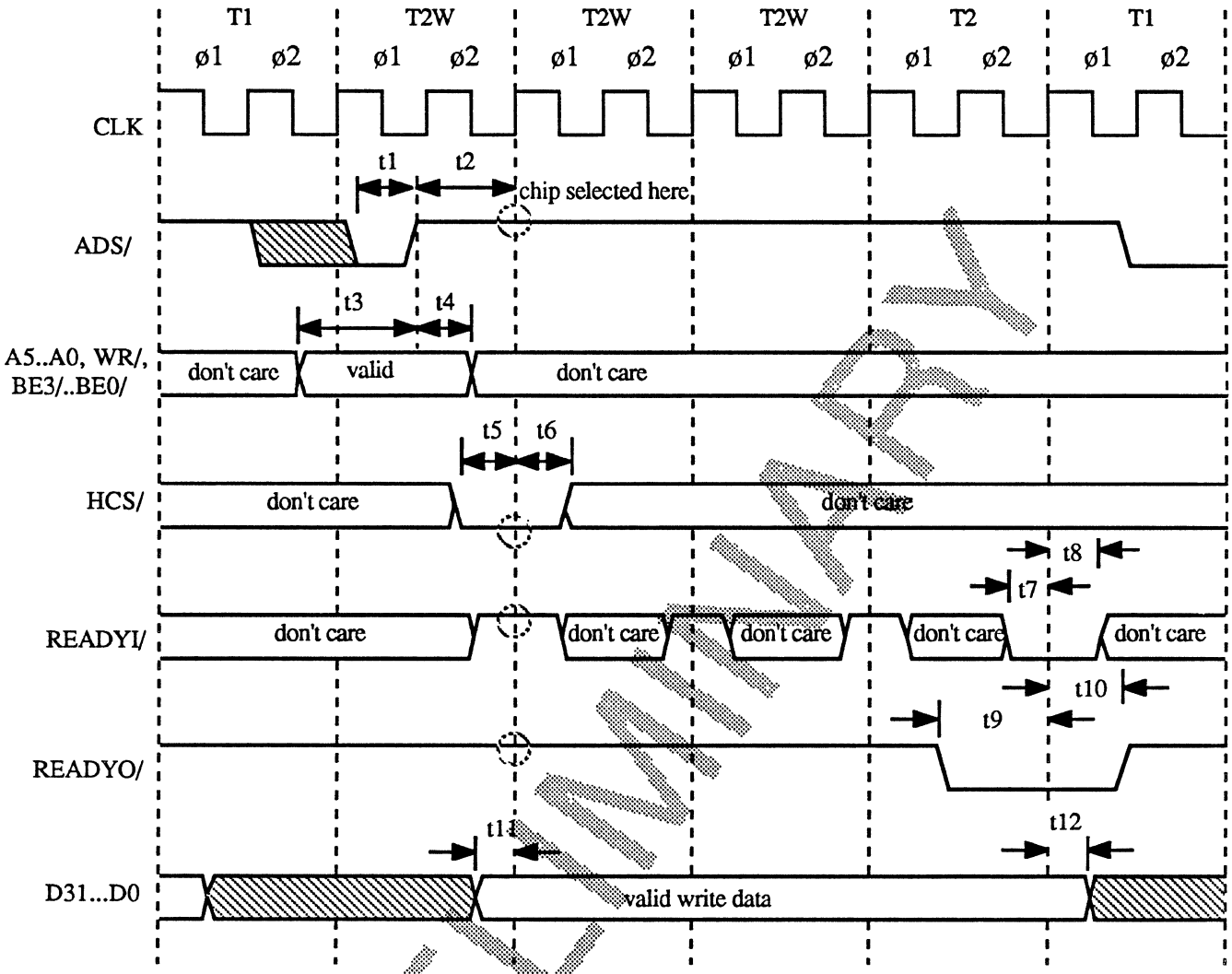
Slave Mode Read Cycle



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.

Slave Mode Read Cycle				
Symbol	Description	Min.	Max.	Units
t1	ADS/ pulse width	10	-	ns
t2	ADS/ deasserted setup to CLK rising edge	7	35	ns
t3	Control and address setup to ADS/ deasserted	10	-	ns
t4	Control and address hold from ADS/ deasserted	10	-	ns
t5	HCS/ setup to CLK rising edge	7	-	ns
t6	HCS/ hold from CLK rising edge	5	-	ns
t7	READYI/ setup to CLK rising edge	9	-	ns
t8	READYI/ hold from CLK rising edge	4	-	ns
t9	READYO/ setup to CLK rising edge	23	38	ns
t10	READYO/ hold from CLK rising edge	3	19	ns
t11	Data setup to CLK rising edge	2	-	ns
t12	Data hold from CLK rising edge	5	25	ns

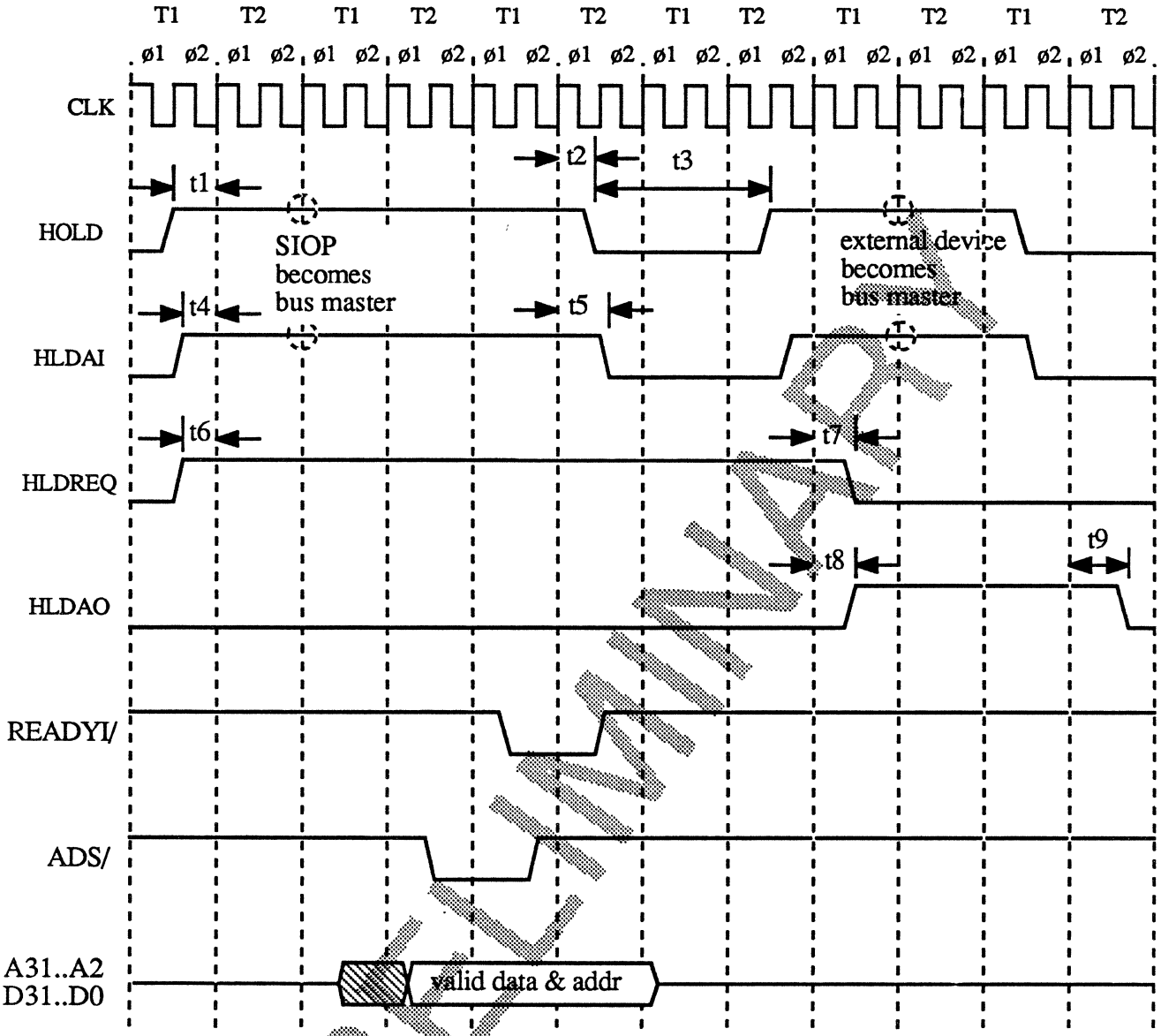
Slave Mode Write Cycle



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.

Slave Mode Write Cycle				
Symbol	Description	Min.	Max.	Units
t1	ADS/ pulse width	10	-	ns
t2	ADS/ deasserted setup to CLK rising edge	7	35	ns
t3	Control and address setup to ADS/ deasserted	10	-	ns
t4	Control and address hold from ADS/ deasserted	10	-	ns
t5	HCS/ setup to CLK rising edge	7	-	ns
t6	HCS/ hold from CLK rising edge	5	-	ns
t7	READYI/ setup to CLK rising edge	9	-	ns
t8	READYI/ hold from CLK rising edge	4	-	ns
t9	READYO/ setup to CLK rising edge	23	38	ns
t10	READYO/ hold from CLK rising edge	3	19	ns
t11	Data setup to CLK rising edge	0	-	ns
t12	Data hold from CLK rising edge	20	-	ns

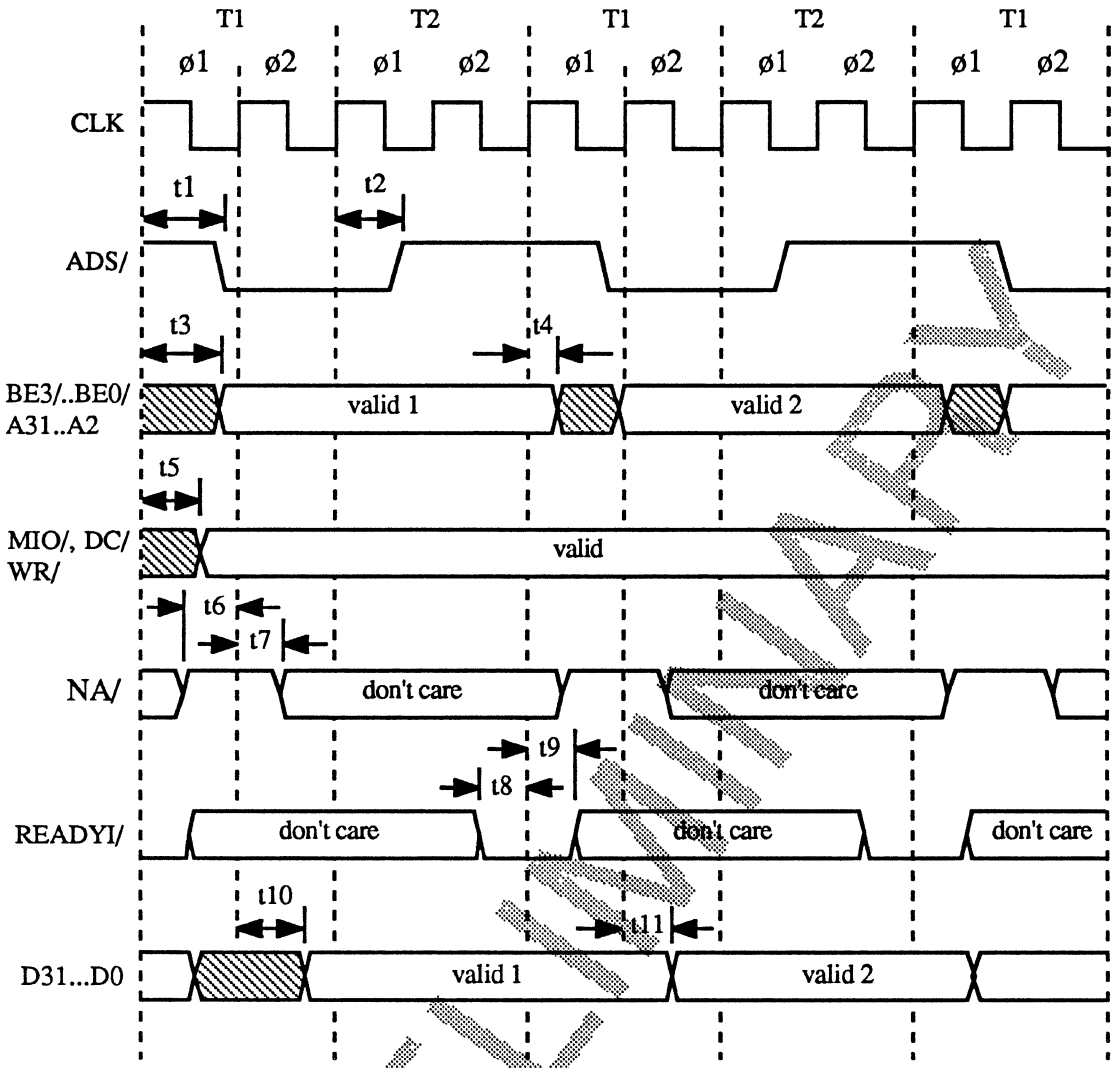
Bus Arbitration



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.

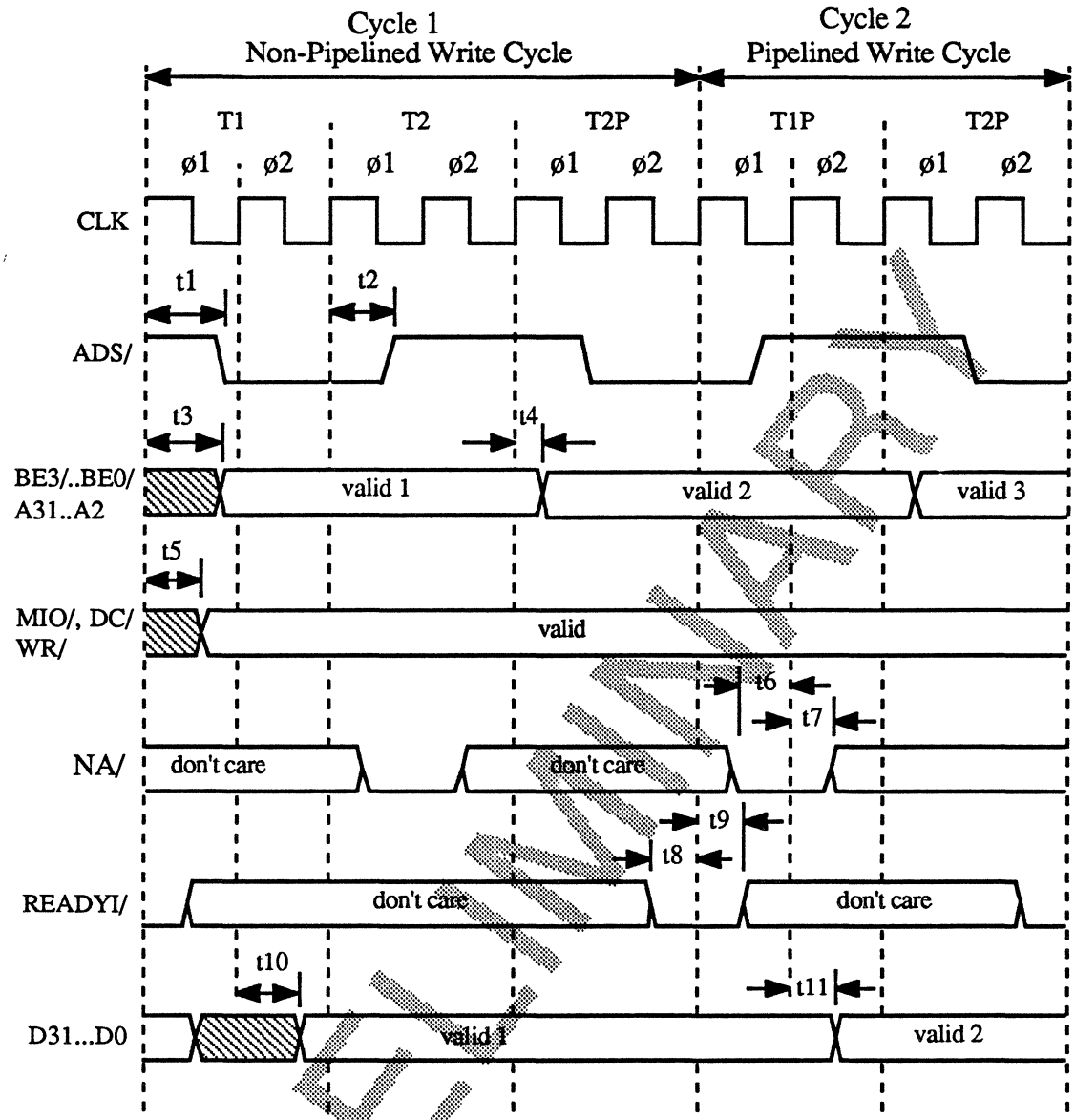
Bus Arbitration				
Symbol	Description	Min.	Max.	Units
t1	HOLD setup to CLK rising edge	15	-	ns
t2	HOLD hold from CLK rising edge	4	-	ns
t3	HOLD inactive to HOLD active	2	-	CLK
t4	HLDAI setup to CLK rising edge	3	-	ns
t5	HLDAI hold from CLK rising edge	3	-	ns
t6	HLDREQ setup to CLK rising edge	3	-	ns
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 Write Cycle - Non-Pipelined Next Address



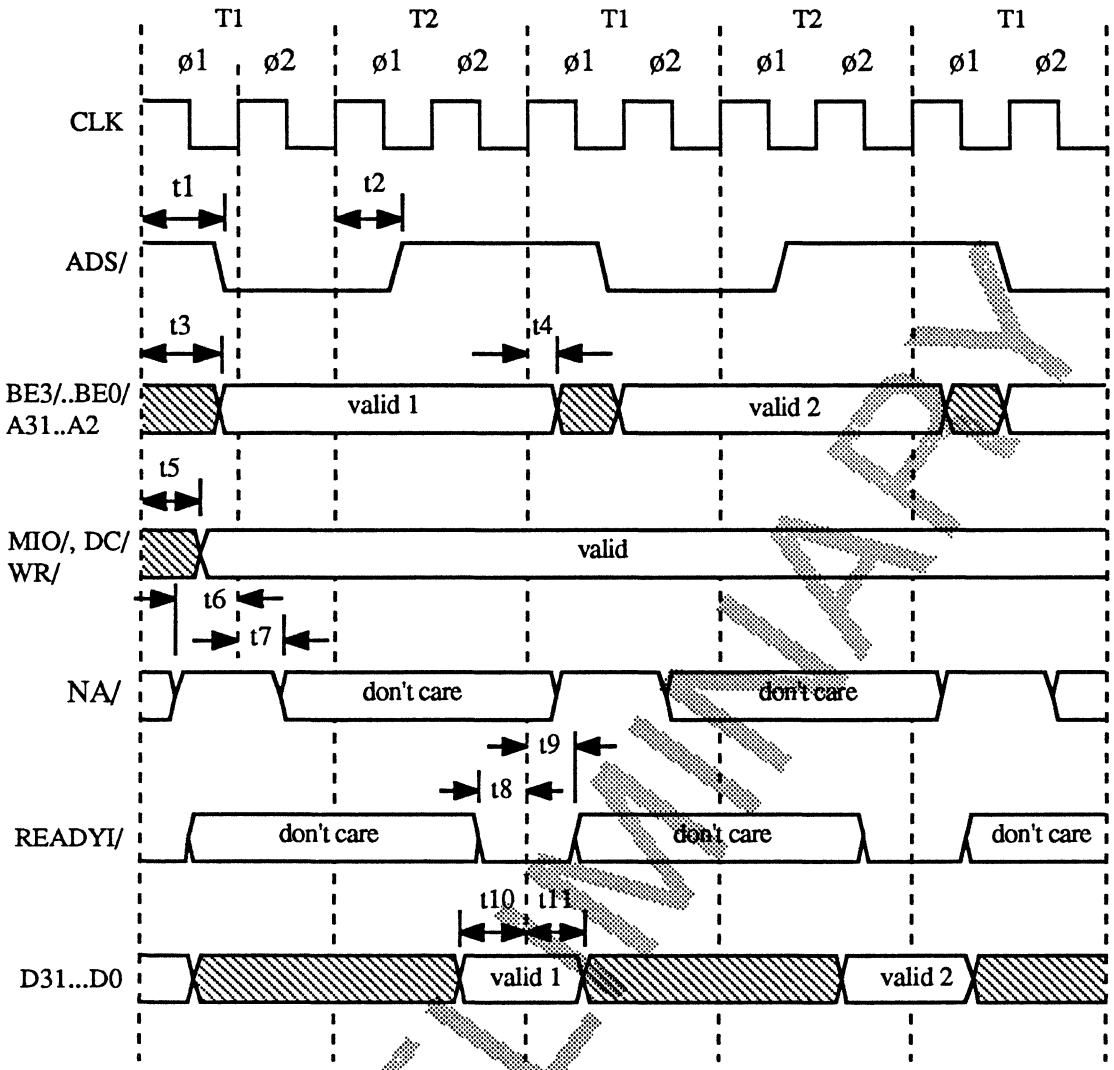
Bus Master Memory Write Cycle - Non-Pipelined Next Address				
Symbol	Description	Min.	Max.	Units
t1	ADS/ valid from CLK rising edge	4	21	ns
t2	ADS/ hold from CLK rising edge	4	30	ns
t3	BE3/..BE0/ and address valid from CLK rising edge	4	24	ns
t4	BE3/..BE0/ and address hold from CLK rising edge	4	30	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	4	-	ns
t10	Data valid from CLK rising edge	5	27	ns
t11	Data hold from CLK rising edge	2	-	ns

Bus Master Memory Write Cycle - Pipelined Next Address



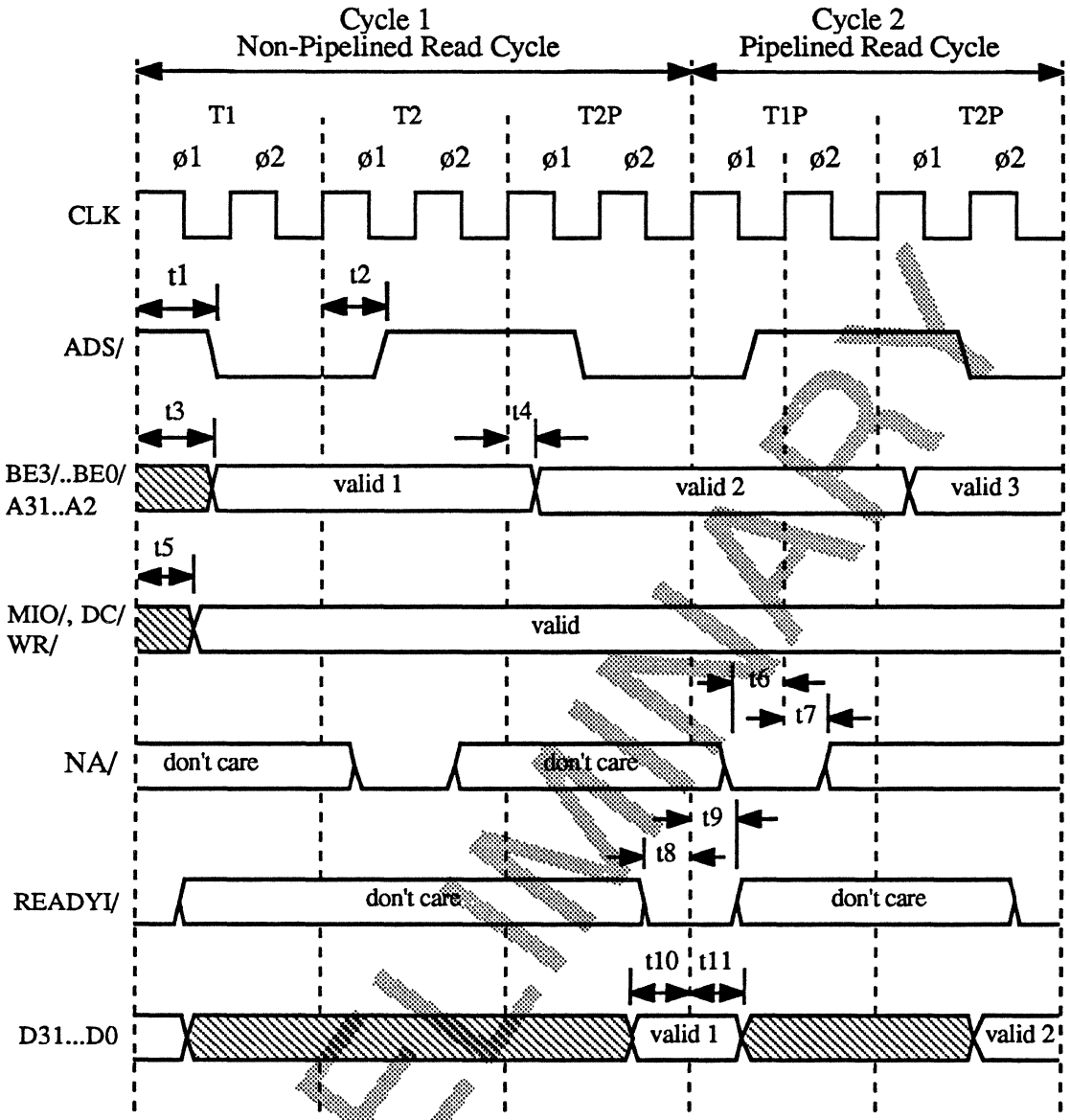
Bus Master Memory Write Cycle - Pipelined Next Address				
Symbol	Description	Min.	Max.	Units
t1	ADS/ valid from CLK rising edge	4	21	ns
t2	ADS/ hold from CLK rising edge	4	30	ns
t3	BE3/..BE0/ and address valid from CLK rising edge	4	24	ns
t4	BE3/..BE0/ and address hold from CLK rising edge	4	30	ns
t5	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	4	-	ns
t10	Data valid from CLK rising edge	5	27	ns
t11	Data hold from CLK rising edge	2	-	ns

Bus Master Memory Read Cycle - Non-Pipelined Next Address



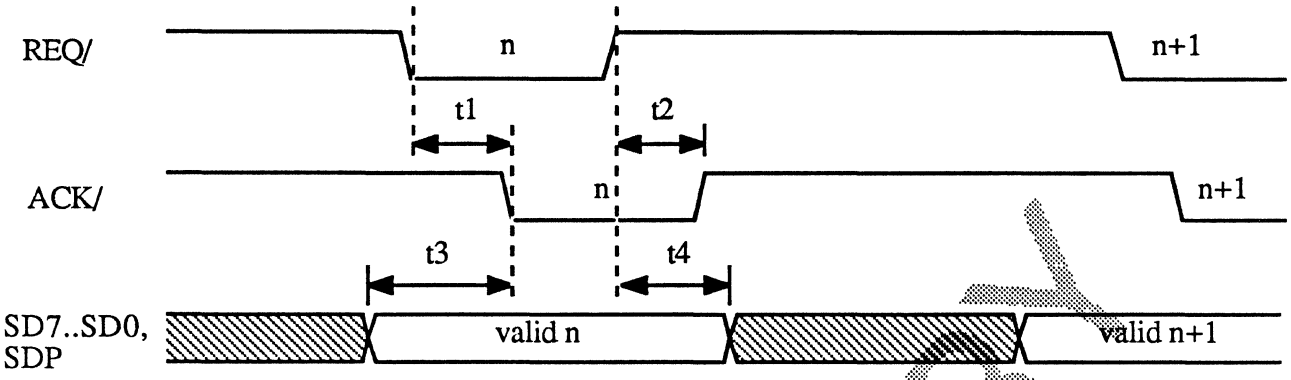
Bus Master Memory Read Cycle - Non-Pipelined Next Address				
Symbol	Description	Min.	Max.	Units
t1	ADS/ valid from CLK rising edge	4	21	ns
t2	ADS/ hold from CLK rising edge	4	30	ns
t3	BE3/..BE0/ and address valid from CLK rising edge	4	24	ns
t4	BE3/..BE0/ and address hold from CLK rising edge	4	30	ns
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	4	-	ns
t10	Data setup to CLK rising edge	7	-	ns
t11	Data hold from CLK rising edge	5	-	ns

Bus Master Memory Read Cycle - Pipelined Next Address



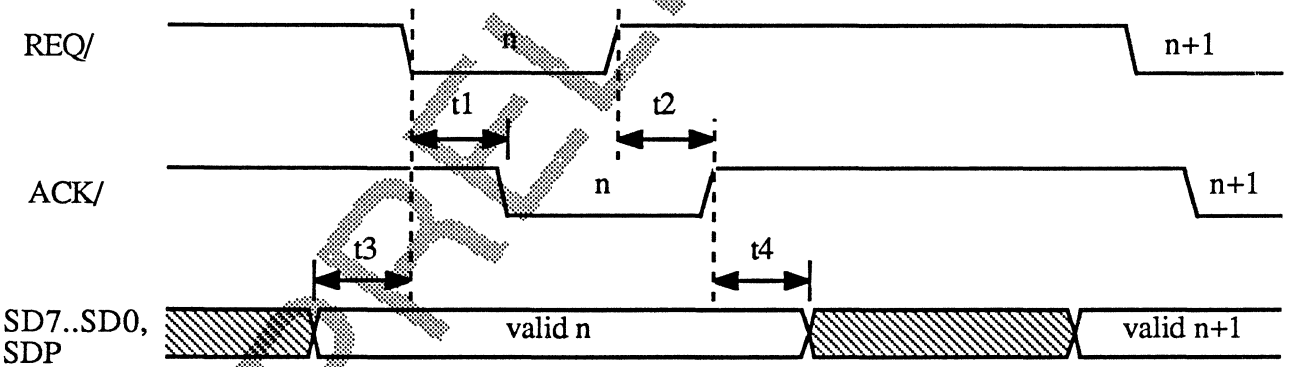
Bus Master Memory Read Cycle - Pipelined Next Address				
Symbol	Description	Min.	Max.	Units
t1	ADS/ valid from CLK rising edge	4	21	ns
t2	ADS/ hold from CLK rising edge	4	30	ns
t3	BE3/..BE0/ and address valid from CLK rising edge	4	24	ns
t4	BE3/..BE0/ and address hold from CLK rising edge	4	30	ns
t5	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	4	-	ns
t10	Data setup to CLK rising edge	7	-	ns
t11	Data hold from CLK rising edge	5	-	ns

Initiator Asynchronous Send



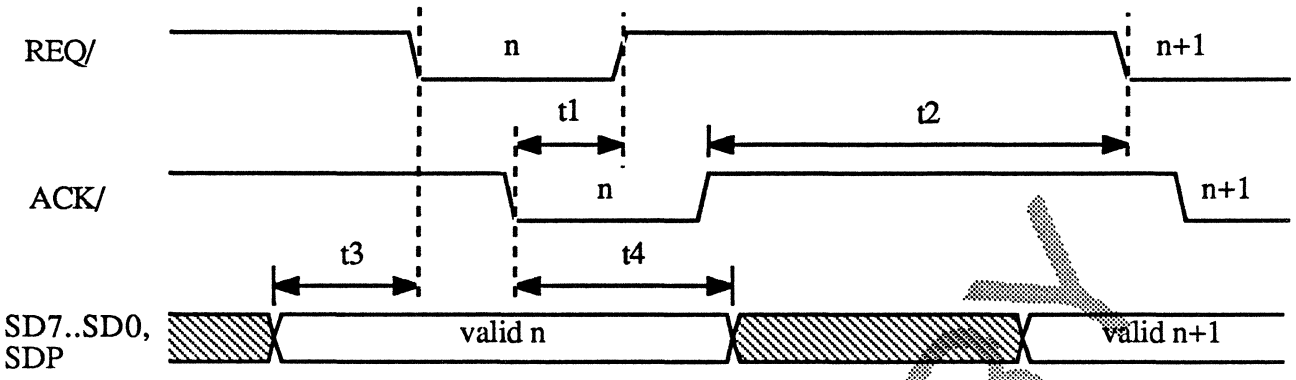
Initiator Asynchronous Send				
Symbol	Description	Min.	Max.	Units
t1	ACK/ asserted from REQ/ asserted	10	-	ns
t2	ACK/ deasserted from REQ/ deasserted	10	-	ns
t3	Data setup to ACK/ asserted	55	-	ns
t4	Data hold from REQ/ deasserted	20	-	ns

Initiator Asynchronous Receive



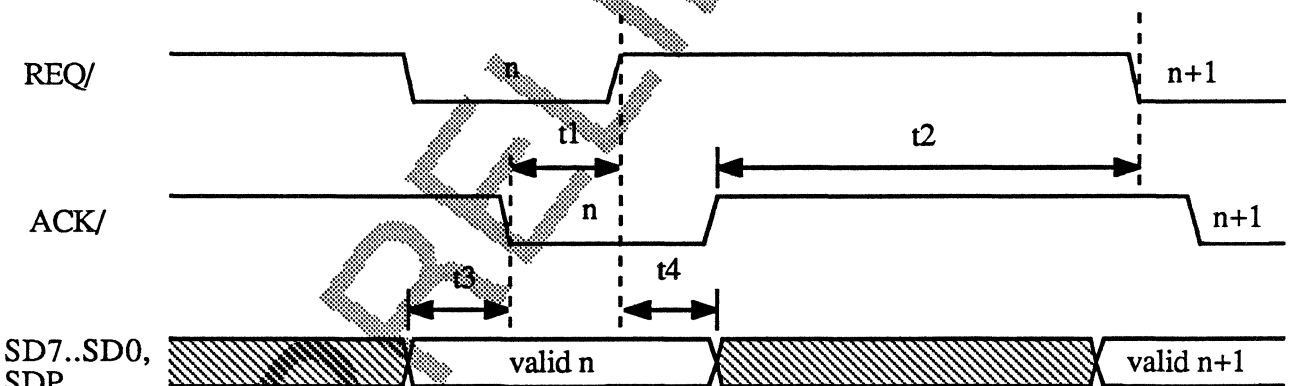
Initiator Asynchronous Receive				
Symbol	Description	Min.	Max.	Units
t1	ACK/ asserted from REQ/ asserted	10	-	ns
t2	ACK/ deasserted from REQ/ deasserted	10	-	ns
t3	Data setup to REQ/ asserted	0	-	ns
t4	Data hold from ACK/ deasserted	0	-	ns

Target Asynchronous Send



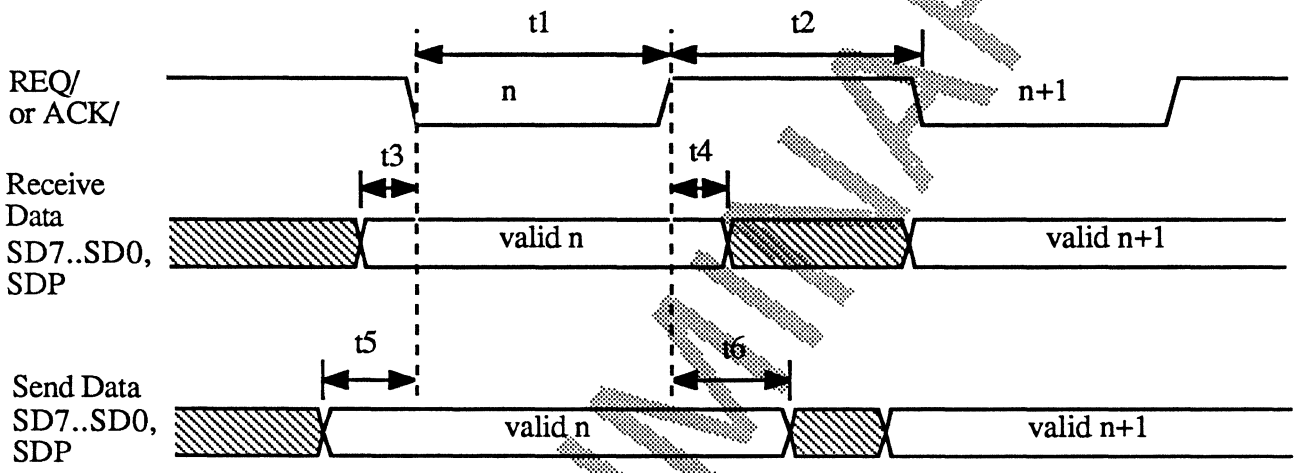
Target Asynchronous Send				
Symbol	Description	Min.	Max.	Units
t1	REQ/ deasserted from ACK/ asserted	10	-	ns
t2	REQ/ asserted from ACK/ deasserted	10	-	ns
t3	Data setup to REQ/ asserted	55	-	ns
t4	Data hold from ACK/ asserted	20	-	ns

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

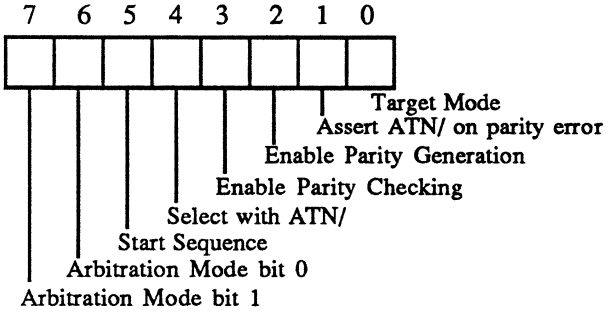


Initiator and Target Synchronous Transfers				
Symbol	Description	Min.	Max.	Units
t1	REQ/ or ACK/ assertion pulse width	90	-	ns
t2	REQ/ or ACK/ deassertion pulse width	90	-	ns
t3	Receive data setup to REQ/ or ACK/ asserted	0	-	ns
t4	Receive data hold from REQ/ or ACK/ asserted	45	-	ns
t5	Send data setup to REQ/ or ACK/ asserted	55	-	ns
t6	Send data hold from REQ/ or ACK/ asserted	100	-	ns

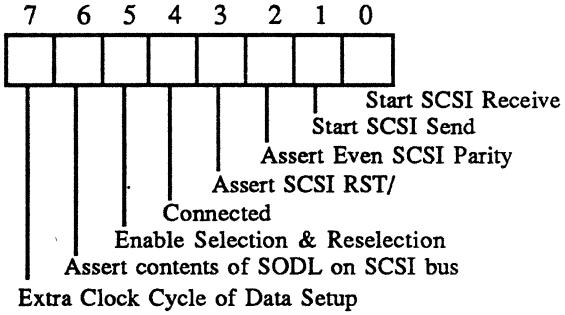
NCR 53C700 SIOP386

Register Summary

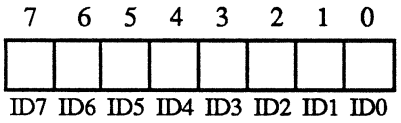
SCNTL0 Register R/W (00h)



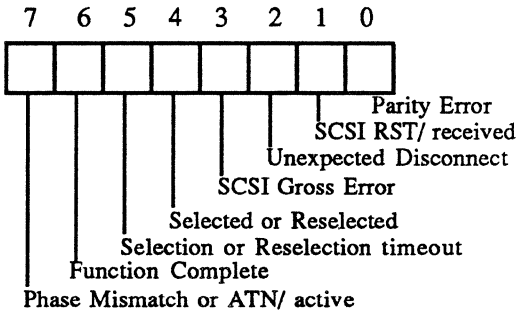
SCNTL1 Register R/W (01h)



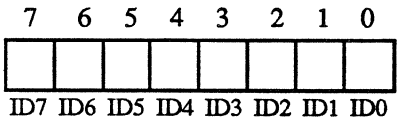
SDID Register R/W (02h)



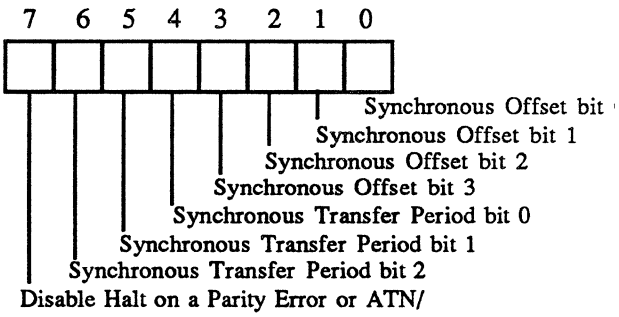
SIEN Register R/W (03h)



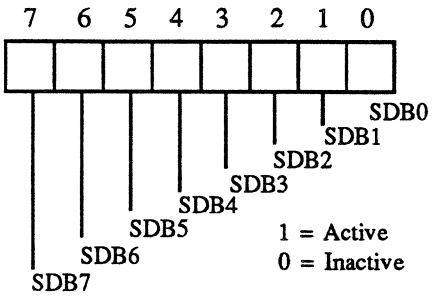
SCID Register R/W (04h)



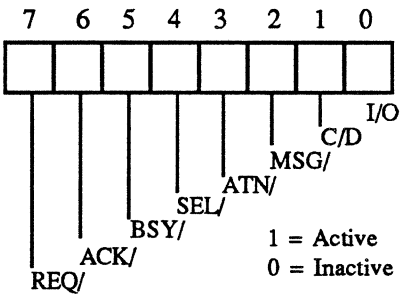
SXFER Register R/W (05h)



SODL Register R/W (06h)



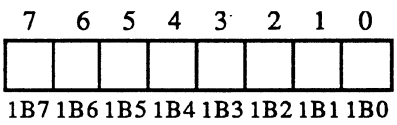
SOCL Register R/W (07h)



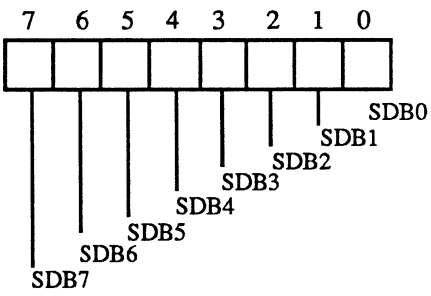
NCR 53C700 SIOP386

Register Summary

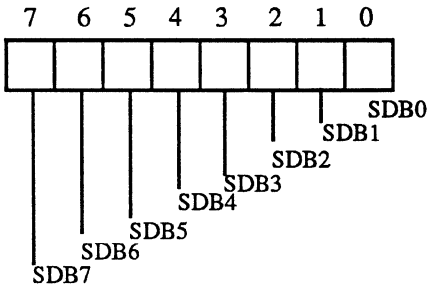
SFBR Register R (08h)



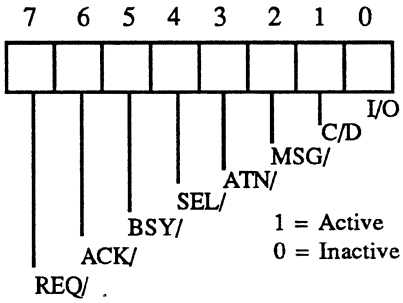
SIDL Register R (09h)



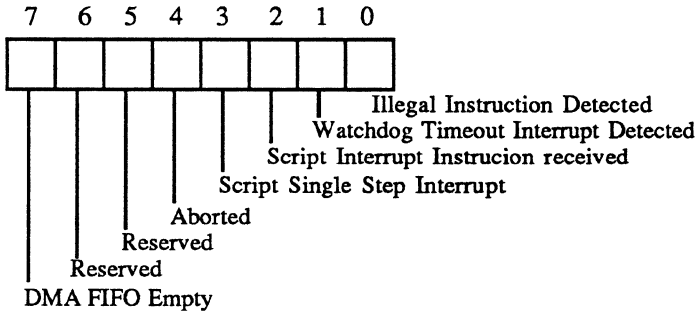
SBDL Register R (0Ah)



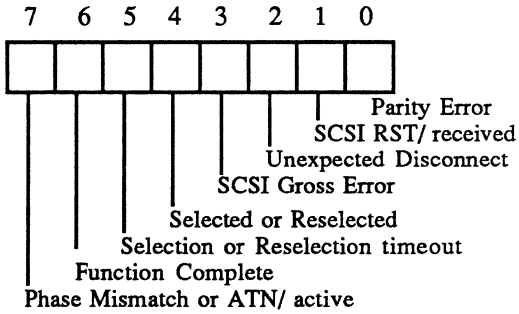
SBCL Register R (0Bh)



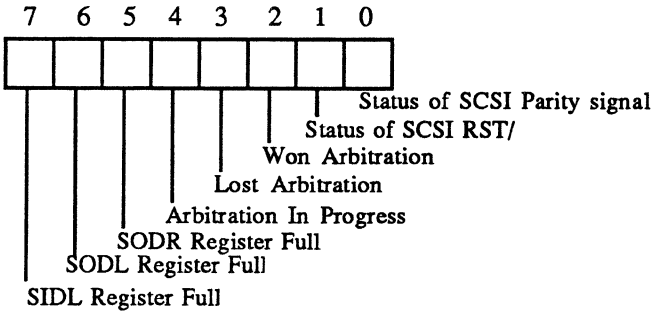
DSTAT Register R (0Ch)



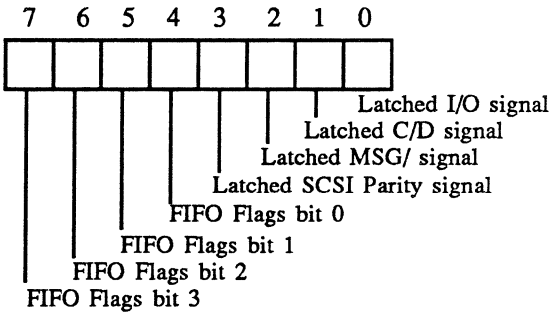
SSTAT0 Register R (0Dh)



SSTAT1 Register R (0Eh)



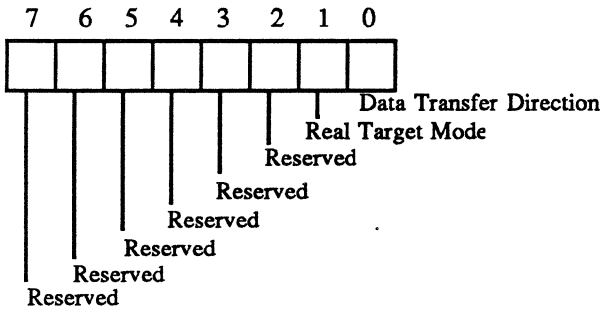
SSTAT2 Register R (0Fh)



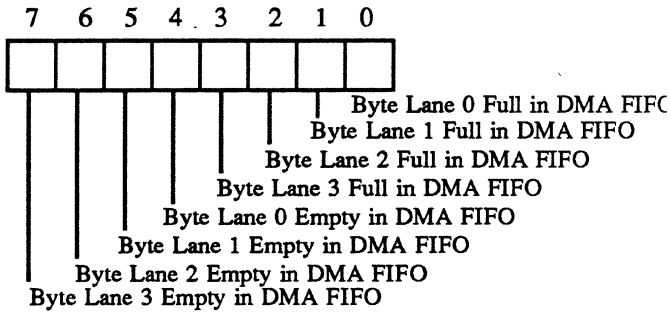
NCR 53C700 SIOP386

Register Summary

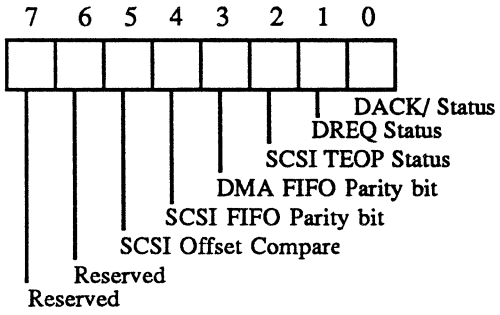
CTEST0 Register R (14h)



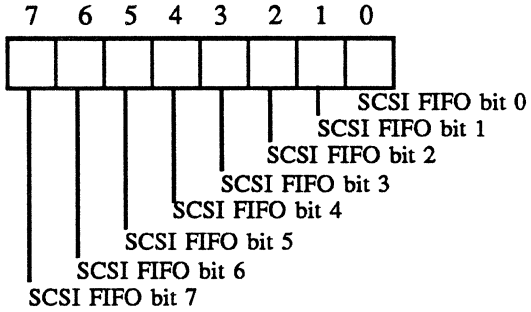
CTEST1 Register R (15h)



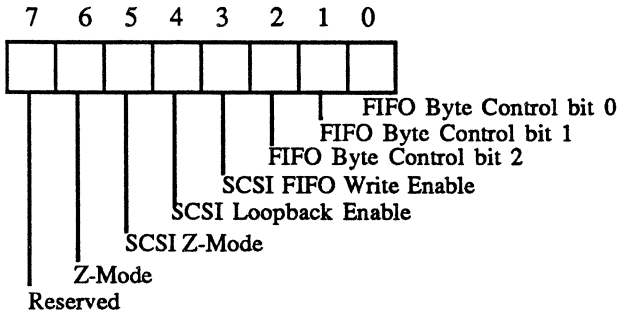
CTEST2 Register R (16h)



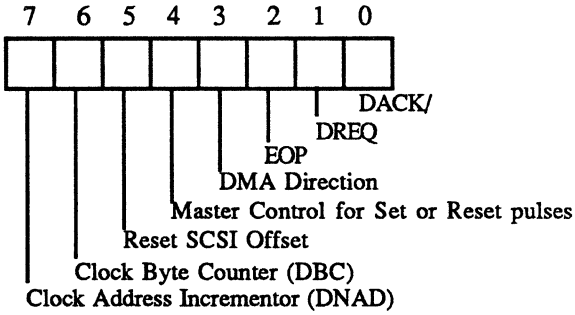
CTEST3 Register R (17h)



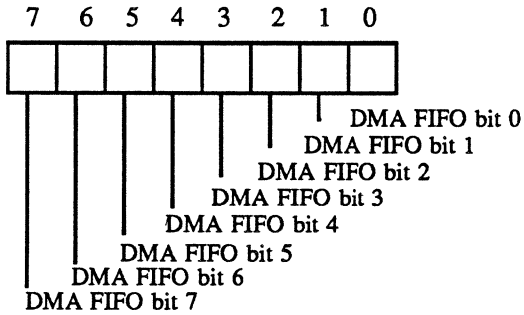
CTEST4 Register R/W (18h)



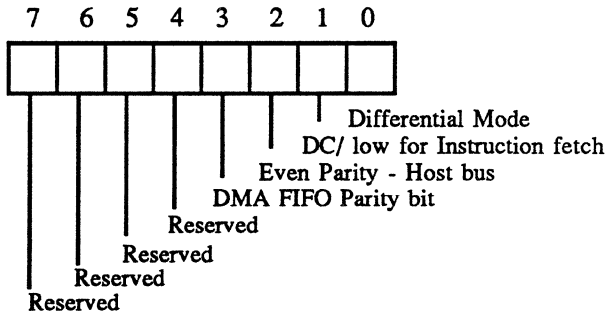
CTEST5 Register R/W (19h)



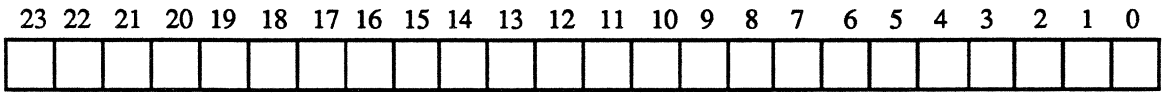
CTEST6 Register R/W (1Ah)



CTEST7 Register R/W (1Bh)



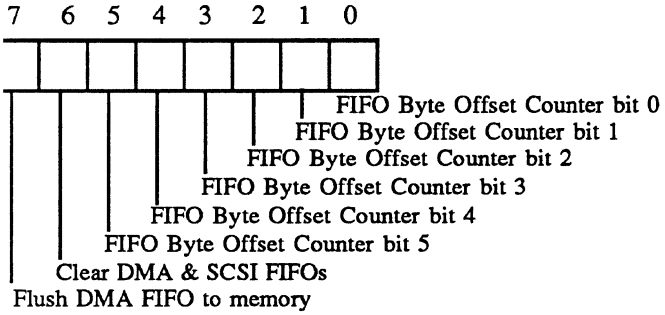
TEMP Register R/W (1C-1Fh)



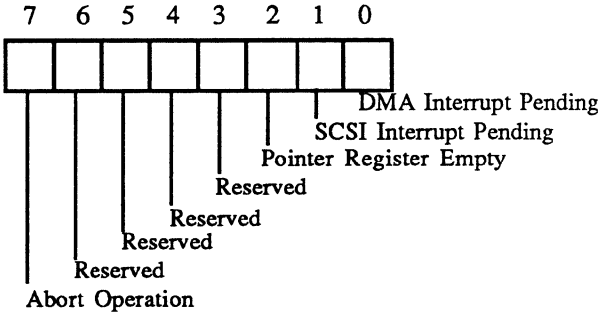
NCR 53C700 SIOP386

Register Summary

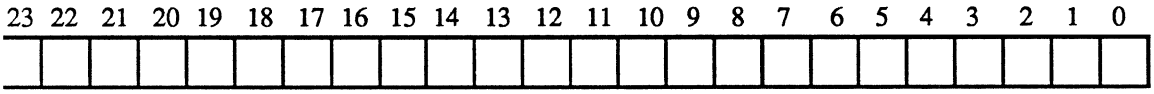
)FIFO Register R/W (20h)



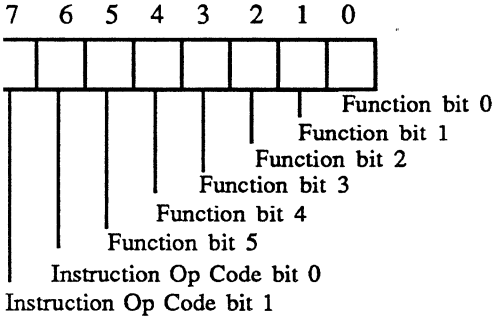
ISTAT Register R/W (21h)



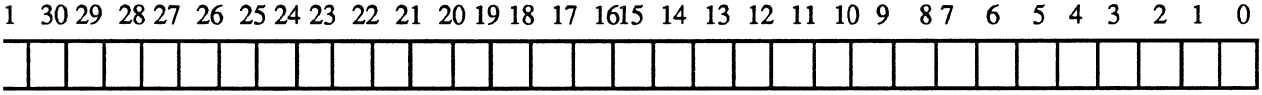
)BC Register R/W (24-26h)



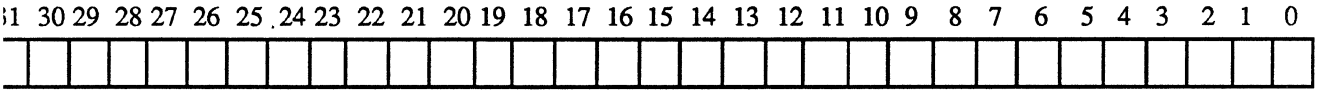
)CMD Register R/W (27h)



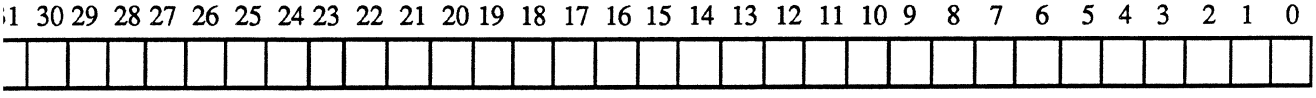
NAD Register R/W (28-2Bh)



)SP Register R/W (2C-2Fh)



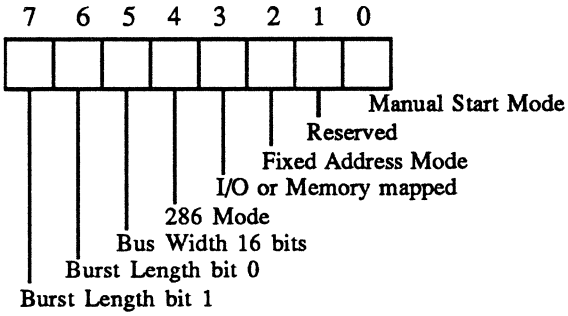
)SPS Register R/W (30-33h)



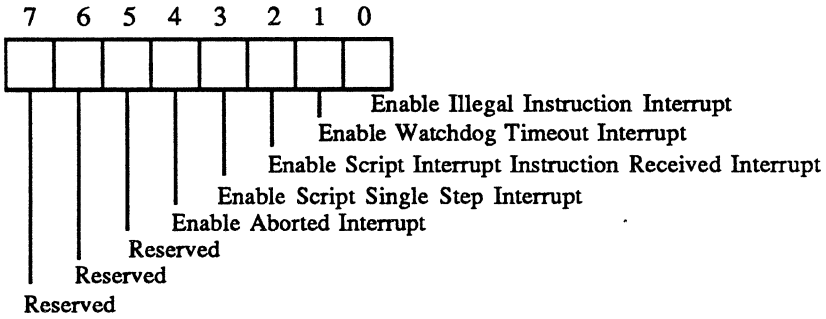
NCR 53C700 SIOP386

Register Summary

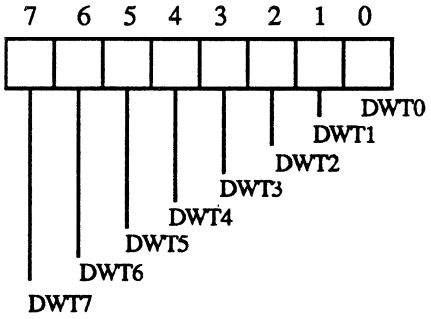
DMODE Register R/W (34h)



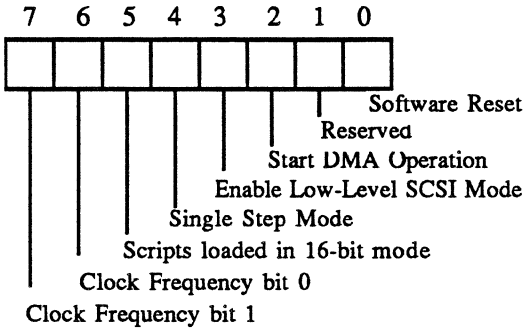
DIEN Register R/W (39h)



DWT Register R/W (3Ah)



DCNTL Register R/W (3Bh)



NCR MICROELECTRONICS
1635 Aeroplaza Drive
Colorado Springs, CO 80916
(719) 596-5612, (800) 525-2252

NORTH WESTERN SALES OFFICE
Suite 209
3130 De La Cruz Boulevard
Santa Clara, CA 95054-2410
(408) 727-6575

NORTH CENTRAL SALES OFFICE
Suite 4050
South Barrington Office Center
33 West Higgins Road
South Barrington, IL 60010
(312) 426-4600

NORTH EASTERN SALES OFFICE
Suite 4000
500 West Cummings Park
Woburn, MA 01801-6336
(617) 933-0778

SOUTH WESTERN SALES OFFICE
1940 Century Park East
Los Angeles, CA 90067
(213) 556-5231, (213) 556-5395

Suite 255
3300 Irvine Avenue
Newport Beach, CA 92660
(714) 474-7264

SOUTH CENTRAL SALES OFFICE
Suite 100
400 Chisholm Place
Plano, TX 75075
(214) 578-9113

SOUTH EASTERN SALES OFFICE
Suite 250
700 Old Roswell Lake Parkway
Roswell, GA 30076
(404) 587-3136

EUROPEAN SALES OFFICE
NCR GMBH
Gustav-Heinemann-Ring 133
8000 Munchen 83
West Germany
(49) 89-632-202

ASIA/PACIFIC
2501 Vicwood Plaza
199 Des Voeux Road
Central
Hong Kong
852-5-859-6888