# Symbios® SYM53C1010-33 PCI to Dual Channel Ultra3 SCSI Multifunction Controller

# Technical Manual

**May 2000**

*Version 3.0*

LSI LOGIC®

Order Number S14025.B

Document DB14-000132-02, Third Edition (May 2000)
This document describes the LSI Logic Corporation's Symbios®
SYM53C1010-33 PCI to Dual Channel Ultra3 SCSI Multifunction Controller and will remain the official reference source for all revisions/releases of this product until rescinded by an update.

**To receive product literature, visit us at http://www.lsilogic.com.**



Ultra SCSI is the term used by the SCSI Trade Association (STA) to describe Fast-20 SCSI, as documented in the SCSI-3 Fast-20 Parallel Interface standard, X3.277-199X.

Ultra2 SCSI is the term used by the SCSI Trade Association (STA) to describe Fast-40 SCSI, as documented in the SCSI Parallel Interface–2 standard, (SPI–2) X3T10/1142D.



TRADEMARK ACKNOWLEDGMENT
The LSI Logic logo design, TolerANT, Symbios, SCRIPTS, LVD Link, SDMS, and SURElink are registered trademarks or trademarks of LSI Logic Corporation. All other brand and product names may be trademarks of their respective companies.

# Preface

This book is the primary reference and technical manual for the LSI Logic Symbios® SYM53C1010-33 PCI to Dual Channel Ultra3 SCSI Multifunction Controller. It contains a complete functional description for the product and includes complete physical and electrical specifications.

## Audience

This document was prepared for system designers and programmers who are using this device to design an Ultra3 SCSI port for PCI-based personal computers, workstations, servers or embedded applications.

## Organization

This document has the following chapters and appendixes:

- Chapter 1, **Introduction,** describes the general information about the SYM53C1010-33.
- Chapter 2, **Functional Description**, describes the main functional areas of the chip in more detail, including the interfaces to the SCSI bus and external memory.
- Chapter 3, **Signal Descriptions**, contains the pin diagram and signal descriptions.
- Chapter 4, **Registers**, describes each bit in the operating registers, and is organized by register address.
- Chapter 5, **SCSI SCRIPTS Instruction Set**, defines all of the SCSI SCRIPTS instructions that are supported by the SYM53C1010-33.
- Chapter 6, **Specifications**, contains the electrical characteristics and AC timing diagrams.
- Appendix A, **Register Summary**, is a register summary.

- Appendix B, **External Memory Interface Diagram Examples**, contains several example interface drawings for connecting the SYM53C1010-33 to external ROMs.

## Related Publications

For background please contact:

**ANSI**
11 West 42nd Street
New York, NY 10036
(212) 642-4900
Ask for document number X3.131-199X (SCSI-2)

**Global Engineering Documents**
15 Inverness Way East
Englewood, CO 80112
(800) 854-7179 or (303) 397-7956 (outside U.S.) FAX (303) 397-2740
Ask for document number X3.131-1994 (SCSI-2) or X3.253
*(SCSI-3 Parallel Interface)*

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

**Prentice Hall**
113 Sylvan Avenue
Englewood Cliffs, NJ 07632
(800) 947-7700
Ask for document number ISBN 0-13-796855-8, *SCSI: Understanding the Small Computer System Interface*

**LSI Logic (Storage Components) Electronic Bulletin Board**
(719) 533-7235
Ask for document *Symbios® PCI to SCSI I/O Processors Programming Guide,* Order Number J25972I

**SCSI Electronic Bulletin Board**
(719) 533-7950

**LSI Logic World Wide Web Home Page**
www.lsil.com

**LSI Logic Internet Anonymous FTP Site**
ftp.lsil.com (204.131.200.1)
Directory: /pub/symchips/scsi

**PCI Special Interest Group**
2575 N. E. Katherine
Hillsboro, OR 97214
(800) 433-5177; (503) 693-6232 (International); FAX (503) 693-8344

## Conventions Used in This Manual

The word *assert* means to drive a signal true or active. The word *deassert* means to drive a signal false or inactive.

Hexadecimal numbers are indicated by the prefix “0x” —for example, 0x32CF. Binary numbers are indicated by the prefix “0b” —for example, 0b0011.0010.1100.1111.

## Revision Record

| Revision | Date | Remarks |
|---|---|---|
| 0.5 | 4/99 | Advanced Information Data - contains Signal Descriptions, Registers, and Mechanical Drawings. |
| 1.0 | 11/15/99 | Preliminary Technical Manual. |
| 1.1 | 12/26/99 | Numerous Rewordings. |
| 2.0 | 3/00 | Final Release. |
| 3.0 | 5/00 | Miscellaneous edits, update to Figure 6.42. |

# Contents

**Figures**

**Tables**

# Chapter 1
# Introduction

This chapter provides a general overview on the SYM53C1010-33 PCI to Dual Channel Ultra3 SCSI Multifunction Controller. This chapter contains the following sections:



## 1.1 General Description

The SYM53C1010-33 PCI to Dual Channel Ultra3 SCSI Multifunction Controller brings Ultra3 SCSI performance to host adapter, workstation, and server designs, making it easy to add a high-performance SCSI bus to any PCI system.

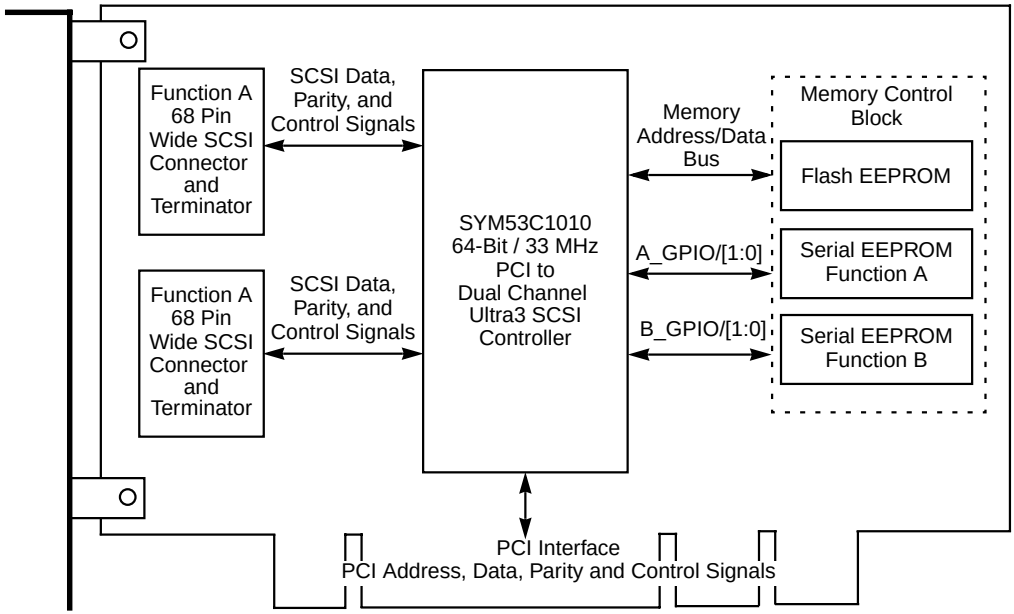
The SYM53C1010 is pin and software compatible with the SYM53C896 PCI to Dual Channel Ultra2 SCSI Multifunction Controller, thus making migration easy (pin compatible means that the SYM53C896 can operate in an SYM53C1010 socket but the SYM53C1010 cannot operate in an SYM53C896 socket). The SYM53C1010 supports a 64-bit or 32-bit, 33 MHz PCI bus. The Ultra3 SCSI features in the SYM53C1010 are: Double Transition (DT) clocking, Cyclic Redundancy Check (CRC), and Domain Validation. These features comply with the Ultra160 SCSI industry initiative.

DT clocking permits the SYM53C1010 to transfer data up to 160 megabytes per second (Mbytes/s) on each channel for a total of 320

Mbytes/s. CRC improves the integrity of the SCSI data transmission through enhanced detection of communication errors. Asynchronous Information Protection (AIP) augments CRC to protect all nondata phases, providing complete end-to-end protection of the SCSI I/O. SURElink™ Domain Validation detects the SCSI bus configuration and automatically tests and adjusts the SCSI transfer rate to optimize interoperability. Three levels of Domain Validation are provided, assuring robust system operation.

The SYM53C1010 has a local memory bus. This allows local storage of the device's BIOS ROM in flash memory or standard EPROMs. The SYM53C1010 supports programming of local flash memory for BIOS updates. The chip is packaged in a 329 Ball Grid Array (BGA). Figure 1.1 shows a typical SYM53C1010 board application connected to external ROM or flash memory.

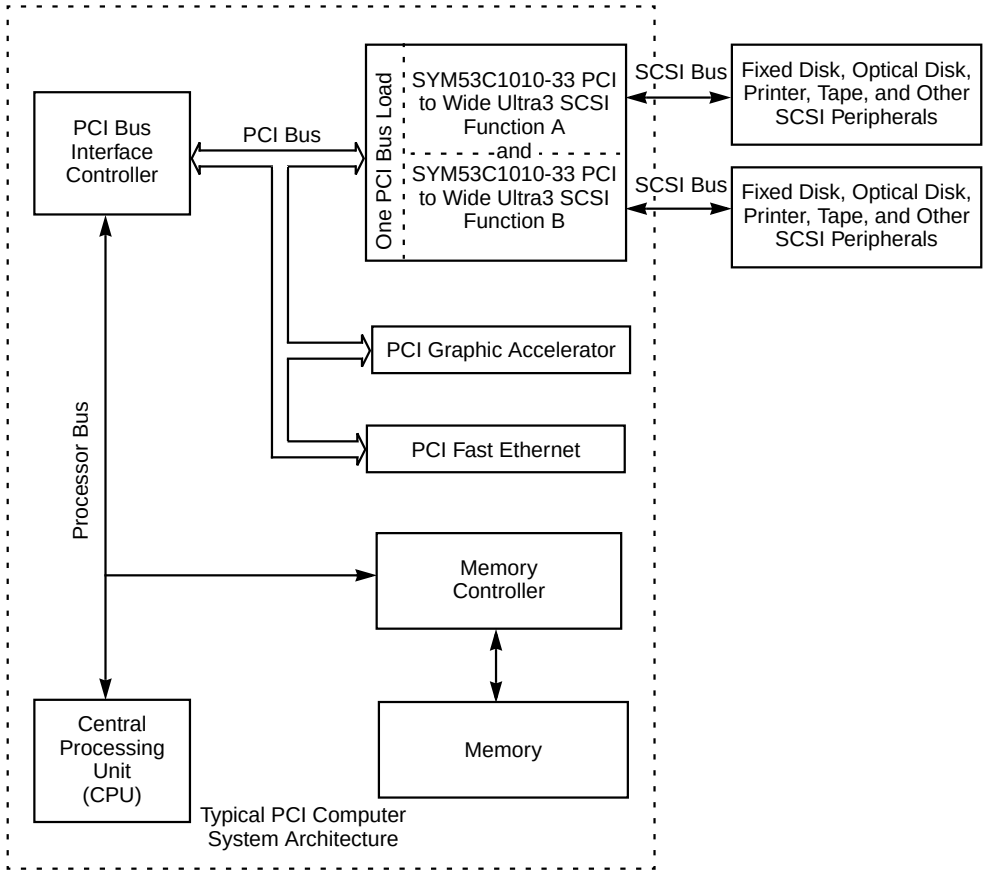
**Figure 1.1 Typical SYM53C1010-33 Board Application**




LVD Link™ technology is the LSI Logic implementation of Low Voltage Differential (LVD). LVD Link transceivers allow the SYM53C1010 to perform either Single-Ended (SE) or LVD transfers. The SYM53C1010 integrates two high-performance SCSI cores, a 64-bit/33 MHz PCI bus master DMA core, and the LSI Logic SCSI SCRIPTS™ processor to

meet the flexibility requirements of Ultra3 SCSI standards. It implements multithreaded I/O algorithms with minimum processor intervention, solving the protocol overhead problems of previous intelligent and nonintelligent adapter designs. Figure 1.2 illustrates a typical SYM53C1010 system application.

**Figure 1.2 Typical SYM53C1010-33 System Application**



### 1.1.1 New Features in the SYM53C1010-33

The SYM53C1010 is functionally similar to the SYM53C896 PCI to Dual Channel Ultra2 SCSI Multifunction Controller, with additional features and benefits. Following is a list of new SYM53C1010 features from the SYM53C896:

- Complies with PCI Rev. 2.2 specification
- Supports Ultra3 DT clocking for data transfers up to 160 Mbytes/s per channel
- Supports enhanced protection on nondata asynchronous phases through AIP
- Supports CRC checking and generation in DT phases
- Supports Domain Validation
  - Basic (level 1) with Inquiry Command (Inquiry Check)
  - Enhanced (level 2) with Read/Write Buffer
  - Margined (level 3) with margining of LVD drivers and programmable skew test
- All cycles to SCRIPTS RAM stay internal to the device, not generating PCI cycles
- SCRIPTS engine with improved instruction fetch performance

## 1.2 Benefits of Ultra3 SCSI

Ultra3 SCSI delivers data up to two times faster than Ultra2 SCSI. Ultra3 SCSI is an extension of the SPI-3 draft standard that allows faster synchronous SCSI data transfer rates than Ultra2 SCSI. When enabled, Ultra3 SCSI performs 80 megatransfers per second (megatransfers/s) resulting in approximately double the synchronous data transfer rates of Ultra2 SCSI. The SYM53C1010 performs 16-bit, Ultra3 SCSI synchronous data transfers as fast as 160 Mbytes/s on each channel providing a total bandwidth of 320 Mbytes/s. This advantage is most noticeable in heavily loaded systems, or large block size applications such as video on-demand and image processing. The Ultra3 data transfer speed is accomplished using DT clocking. DT clocking refers to transferring data on both polarity edges of the request or acknowledge

signals. Data is clocked on both rising and falling edges of the request and acknowledge signals. Double-edge clocking doubles data transfer speeds without increasing the clock rate.

Ultra3 SCSI also includes CRC which offers higher levels of data reliability by ensuring complete integrity of transferred data. CRC is a 32-bit scheme, referred to as CRC-32. CRC is guaranteed to detect all single bit errors, any two bits in error, or any combination of errors within a single 32-bit range.

AIP is also supported by the SYM53C1010, protecting all nondata phases, including command, status, and messages. CRC, along with AIP, provides end-to-end protection of the SCSI I/O.

An advantage of Ultra3 SCSI is that it significantly improves SCSI bandwidth while preserving existing hardware and software investments. The primary software changes required are to enable the chip to perform synchronous negotiations for Ultra3 SCSI rates and to enable the clock quadrupler. Ultra3 SCSI uses the same connectors as Ultra SCSI and Ultra2 SCSI. Chapter 2, “Functional Description” contains more information on migrating an Ultra SCSI or Ultra2 SCSI design to an Ultra3 SCSI design.

## 1.3 Benefits of SURElink (Ultra3 SCSI Domain Validation)

SURElink represents the very latest SCSI interconnect management solution. It ensures robust and low risk Ultra3 SCSI implementations by extending the Domain Validation guidelines documented in the ANSI T10 SPI-3 specifications. Domain Validation verifies that the system is capable of transferring data at Ultra3 speeds, allowing it to renegotiate to lower speed and bus width if necessary. SURElink is the software control for the manageability enhancements in the SYM53C1010 PCI to Dual Channel Ultra3 SCSI Multifunction Controller. Fully integrated in the SDMS™ software solution, SURElink provides Domain Validation at boot time as well as throughout system operation. SURElink extends to the DMI (Desktop Management Interface) based System Management components of SDMS, providing the network administrator remote management capability.

SURElink Domain Validation provides 3 levels of integrity checking: Basic (level 1), Enhanced (level 2) and Margined (level 3). The basic check

consists of an inquiry command to detect gross problems. The enhanced check sends a known data pattern using the Read and Write Buffer commands to detect additional problems. Margined check verifies that the physical parameters have some degree of margin. By varying LVD drive strength and REQ/ACK timing characteristics level 3 verifies that no errors occur on the transfers. These altered signals are only used during the diagnostic check and not during normal system operation. Should errors occur with any of these checks, the system can drop back to a lower transmission speed on a per target basis to ensure robust system operation.

## 1.4 Benefits of LVD Link

The SYM53C1010 supports LVD through LVD Link. This signaling technology increases the reliability of SCSI data transfers over longer distances than are supported by SE SCSI. The low current output of LVD allows the I/O transceivers to be integrated directly onto the chip. LVD provides the reliability of High Voltage Differential (HVD) SCSI without the added cost of external differential transceivers. Ultra3 SCSI with LVD allows a longer SCSI cable and more devices on the bus, with the same cables defined in the SCSI-3 Parallel Interface standard for Ultra SCSI. LVD provides a long-term migration path to even faster SCSI transfer rates without compromising signal integrity, cable length, or connectivity.

For backward compatibility to existing SE devices, the SYM53C1010 features universal LVD Link transceivers that support LVD SCSI and SE SCSI. This allows use of the SYM53C1010 in both legacy and Ultra3 SCSI applications.

## 1.5 Benefits of TolerANT® Technology

The SYM53C1010 features TolerANT technology, which includes active negation on the SCSI drivers and input signal filtering on the SCSI receivers. Active negation causes the SCSI Request, Acknowledge, Data, and Parity signals to be actively driven high rather than passively pulled up by terminators.

TolerANT receiver technology improves data integrity in unreliable cabling environments where other devices would be subject to data

corruption. TolerANT receivers filter the SCSI bus signals to eliminate unwanted transitions, without the long signal delay associated with RC-type input filters. This improved driver and receiver technology helps eliminate double clocking of data which is the single biggest reliability issue with SCSI operations. TolerANT input signal filtering is a built-in feature of the SYM53C1010 and all LSI Logic fast SCSI, Ultra SCSI, Ultra2 SCSI, and Ultra3 SCSI devices.

The benefits of TolerANT technology include increased immunity to noise when the signal is going HIGH, better performance due to balanced duty cycles, and improved fast SCSI transfer rates. In addition, TolerANT SCSI devices do not cause glitches on the SCSI bus at power-up or power-down. This protects other devices on the bus from data corruption. When it is used with the LVD Link transceivers, TolerANT technology provides excellent signal quality and data reliability in real world cabling environments. TolerANT technology is compatible with both the Alternative One and Alternative Two termination schemes proposed by the American National Standards Institute.

## 1.6 Summary of SYM53C1010-33 Benefits

This section provides a summary of the SYM53C1010 features and benefits. It contains information on SCSI Performance, PCI Performance, Integration, Ease of Use, Flexibility, Reliability, and Testability.

### 1.6.1 SCSI Performance

The SYM53C1010-33:

- Performs wide, Ultra3 SCSI synchronous data transfers as fast as 160 Mbytes/s on each SCSI channel for a total of 320 Mbytes/s using DT clocking.
- Supports CRC checking and generation in DT phases.
- Protects nondata phases with AIP.
- Supports Domain Validation:
  - Basic (level 1)
  - Enhanced (level 2)
  - Margined (level 3)

- Includes integrated LVD Link universal transceivers:
  - Supports SE and LVD signals.
  - Allows greater device connectivity and longer cable length.
  - LVD Link transceivers save the cost of external differential transceivers.
  - Supports a long-term performance migration path.
- Bursts of up to 512 bytes across the PCI bus with an independent 896–920 byte FIFO on each SCSI channel.
- Includes two separate SCSI channels on one chip.
- Handles phase mismatches in SCRIPTS without interrupting the system processor.
- Includes an on-chip SCSI clock quadrupler that allows the chip to achieve Ultra3 SCSI transfer rates with an input frequency of 40 MHz.
- Includes 8 Kbytes of internal RAM for SCRIPTS instruction storage for each SCSI channel.
- Supports 31 levels of SCSI synchronous offset in the Single Transition (ST) mode and 62 levels in the DT mode.
- Supports variable block size and scatter/gather data transfers.
- Performs sustained Memory-to-Memory DMA transfers to approximately 100 Mbytes/s.
- Minimizes the SCSI I/O start latency.
- Performs complex bus sequences without interrupts, including restoring data pointers.
- Reduces ISR overhead through a unique interrupt status reporting method.
- Includes RAID ready SCSI on the motherboard with separate interrupts for routing to a RAID adapter.
- Supports Load/Store SCRIPTS instructions to increase the performance of data transfers to and from the chip registers without using PCI cycles.
- Includes SCRIPTS support of 64-bit addressing.

- Supports target disconnect and later reconnect with no interrupt to the system processor.
- Supports multithreaded I/O algorithms in SCSI SCRIPTS with fast I/O context switching.
- Supports expanded register Move instructions to support additional arithmetic capability.

### 1.6.2 PCI Performance

The SYM53C1010:

- Complies with PCI 2.2 specification.
- Supports a 64-bit/33 MHz PCI interface for 264 Mbytes/s bandwidth that:
  - Can function in a 32-bit PCI slot
  - Operates at 33 MHz
  - Supports dual address cycle generation for all SCRIPTS
  - Presents a single electrical load to the PCI Bus (True PCI Multifunction Device)
- Bursts 2/4, 4/8, 8/16, 16/32, 32/64, or 64/128 Qword/Dword transfers across the PCI bus.
- Supports 32-bit or 64-bit word data bursts with variable burst lengths.
- Prefetches up to 8 Dwords of SCRIPTS instructions.
- Bursts SCRIPTS opcode fetches across the PCI bus.
- Performs zero wait-state bus master data bursts up to 264 Mbytes/s (@ 33 MHz).
- Supports PCI Cache Line Size (CLS) register.
- Supports PCI Write and Invalidate, Read Line, and Read Multiple commands.
- Complies with PCI Bus Power Management Specification Rev 1.1.
- Complies with PC99.

### 1.6.3 Integration

The following features ease integration of the SYM53C1010 into a system:

- Dual channel Ultra3 SCSI PCI multifunction controller.
- Integrated LVD transceivers.
- Full 32-bit or 64-bit PCI DMA bus master.
- Memory-to-Memory Move instructions allow use as a third-party PCI bus DMA controller.
- Integrated SCRIPTS processor.

### 1.6.4 Ease of Use

The following features of the SYM53C1010 make the device user friendly:

- The SYM53C1010 is pin and software compatible with the SYM53C896 PCI to Dual Channel Ultra2 SCSI Multifunction Controller, thus making migration easy (pin compatible means that the SYM53C896 can operate in an SYM53C1010 socket but the SYM53C1010 cannot operate in an SYM53C896 socket).
- Up to one Mbyte of add-in memory support for BIOS and SCRIPTS storage.
- Reduced SCSI development effort.
- Compiler-compatible with existing SYM53C7XX and SYM53C8XX family SCRIPTS.
- Direct connection to PCI and SCSI SE and LVD.
- Development tools and sample SCSI SCRIPTS available.
- Maskable and pollable interrupts.
- Wide SCSI, A or P cable, and up to 15 devices per SCSI channel supported.
- Three programmable SCSI timers: Select/Reselect, Handshake-to-Handshake, and General Purpose.
- Software for PC-based operating system support.

- Support for relative jumps.
- SCSI Selected As ID bits for responding with multiple IDs.

### 1.6.5 Flexibility

The following features increase the flexibility of the SYM53C1010:

- Universal LVD transceivers are backward compatible with SE devices.
- High level programming interface (SCSI SCRIPTS).
- Programs local and bus flash memory.
- Tailored SCSI sequences execute from main system RAM or internal SCRIPTS RAM.
- Flexible programming interface to tune I/O performance or to adapt to unique SCSI devices.
- Support for changes in the logical I/O interface definition.
- Low level access to all registers and all SCSI bus signals.
- Fetch, Master, and Memory Access control pins.
- Separate SCSI and system clocks.
- SCSI clock quadrupler bits enable Ultra3 SCSI transfer rates with a 40 MHz SCSI clock input.
- Selectable IRQ pin disable bit.
- Compatible with 3.3 V and 5 V PCI.

### 1.6.6 Reliability

The following features enhance the reliability of the SYM53C1010:

- CRC and AIP provide end-to-end SCSI I/O protection.
- 2 kV ESD protection on SCSI signals.
- 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 150 mA.

- Voltage feed-through protection (minimum leakage current through SCSI pads).
- A high proportion of pins are power and ground.
- Power and ground isolation of I/O pads and internal chip logic.
- TolerANT technology provides:
  - Active negation of SCSI Data, Parity, Request, and Acknowledge signals for improved fast SCSI transfer rates.
  - Input signal filtering on SCSI receivers improves data integrity, even in noisy cabling environments.

### 1.6.7 Testability

The following features enhance the testability of the SYM53C1010:

- All SCSI signals accessible through programmed I/O.
- SCSI bus signal continuity checking.
- Support for single-step mode operation.
- JTAG boundary scan.

# Chapter 2
# Functional Description

This chapter provides a functional description of the SYM53C1010-33 PCI to Dual Channel Ultra3 SCSI Multifunction Controller. This chapter is divided into the following sections:

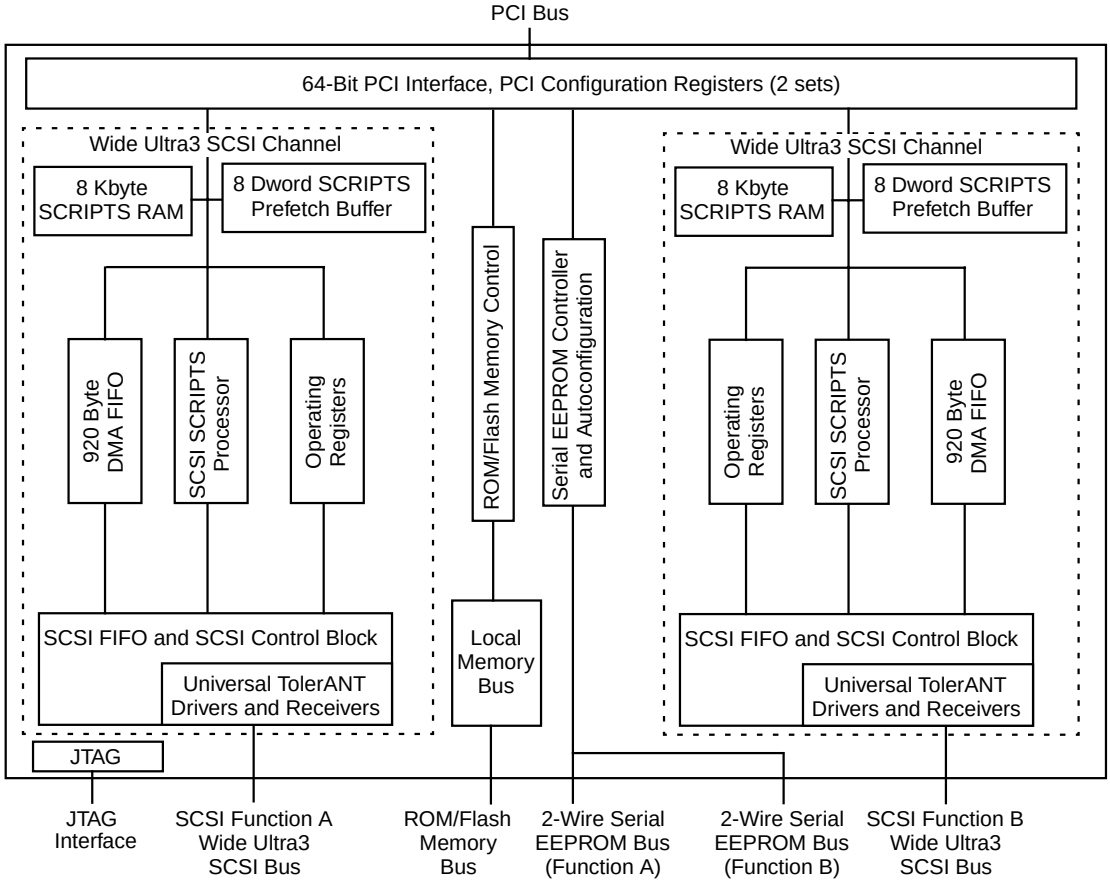


The SYM53C1010-33 PCI to Dual Channel Ultra3 SCSI Multifunction Controller is composed of the following modules:

- 64-bit PCI Interface
- Two independent PCI to Wide Ultra3 SCSI Controllers
- ROM/Flash Memory Controller
- Serial EEPROM Controller

Figure 2.1 illustrates the relationship between these modules.

**Figure 2.1 SYM53C1010-33 Block Diagram**




The SYM53C1010 has two wide Ultra3 SCSI channels in a single package. Each SCSI channel (A and B) incorporates an independent DMA FIFO and a separate internal 8 Kbyte SCRIPTS RAM.

## 2.1 PCI Functional Description

The SYM53C1010 implements two PCI to Wide Ultra3 SCSI controllers in a single package. This configuration presents only one load to the PCI bus and uses only one REQ/ - GNT/ pair for PCI bus arbitration. Separate interrupt signals are generated for SCSI Function A and SCSI Function B.

### 2.1.1 PCI Addressing

There are three physical address spaces defined in the PCI specification:

- PCI Configuration space
- I/O space for operating registers
- Memory space for operating registers

#### 2.1.1.1 Configuration Space

The host processor uses this configuration space to initialize the SYM53C1010. Two independent sets of configuration space registers are defined, one set for each SCSI function. Each SCSI function contains the same register set with identical default values except for the Interrupt Pin. The configuration registers are initialized by the system BIOS using PCI configuration cycles. Each configuration space is a contiguous 256 x 8-bit set of addresses. Decoding C_BE[3:0]/ determines if a PCI cycle is intended to access the configuration register space. The IDSEL bus signal is a “chip select” that allows access to the configuration register space only. A configuration read/write cycle without IDSEL is ignored. The host processor uses the eight lower order address bits (AD[7:0]) to select a specific 8-bit register. Since the SYM53C1010 is a PCI multifunction device, bits AD[10:8] decode either SCSI Function A Configuration register (AD[10:8] = 0b000) or SCSI Function B Configuration register (AD[10:8] = 0b001). Figure 4.1 is an illustration of the PCI Configuration Register Map.

At initialization time, each PCI device is assigned a base address for memory and I/O accesses. In the SYM53C1010, the upper 24 bits of the address are selected. On every access, the SYM53C1010 compares its assigned base addresses with the value on the Address/Data bus during the PCI address phase. If the upper 24 bits match, the access is designated for the SYM53C1010. The low order eight bits define the register to be accessed. A decode of C_BE[3:0]/ determines which register and what type of access is performed.

#### 2.1.1.2 I/O Space

The PCI specification defines I/O space as a contiguous 32-bit I/O address that is shared by all system resources, including the

SYM53C1010. Base Address Register Zero (BAR0) (I/O) determines which 256-byte I/O area this device occupies.

#### 2.1.1.3 Memory Space

The PCI specification defines memory space as a contiguous 64-bit memory address that is shared by all system resources. Base Address Register One (BAR1) (MEMORY) determines which 1 Kbyte memory area this device occupies. Each SCSI function uses an 8 Kbyte SCRIPTS RAM memory space. Base Address Register Two (BAR2) (MEMORY) determines the 8 Kbyte memory area the SCRIPTS RAM occupies.

### 2.1.2 PCI Bus Commands and Functions Supported

Bus commands indicate to the target the type of transaction the master is requesting. Bus commands are encoded on the C_BE[3:0]/ lines during the address phase. PCI bus commands and encoding types appear in Table 2.1.

**Table 2.1 PCI Bus Commands and Encoding Types**

| C_BE[3:0]/ | Command Type | Supported as Master | Supported as Slave |
|---|---|---|---|
| 0000 | Interrupt Acknowledge | No | No |
| 0001 | Special Cycle | No | No |
| 0010 | I/O Read | Yes | Yes |
| 0011 | I/O Write | Yes | Yes |
| 0100 | Reserved | N/A | N/A |
| 0101 | Reserved | N/A | N/A |
| 0110 | Memory Read | Yes | Yes |
| 0111 | Memory Write | Yes | Yes |
| 1000 | Reserved | N/A | N/A |
| 1001 | Reserved | N/A | N/A |
| 1010 | Configuration Read | No | Yes |
| 1011 | Configuration Write | No | Yes |
| 1100 | Memory Read Multiple | Yes[1] | Yes (defaults to 0110) |
| 1101 | Dual Address Cycle (DAC) | Yes | Yes |
| 1110 | Memory Read Line | Yes[1] | Yes (defaults to 0110) |
| 1111 | Memory Write and Invalidate | Yes[2] | Yes (defaults to 0111) |

1. See the DMA Mode (DMODE) register.
2. See the Chip Test Three (CTEST3) register.

#### 2.1.2.1 Interrupt Acknowledge Command

The SYM53C1010 does not respond to this command as a slave and it never generates this command as a master.

#### 2.1.2.2 Special Cycle Command

The SYM53C1010 does not respond to this command as a slave and it never generates this command as a master.

#### 2.1.2.3 I/O Read Command

The SYM53C1010 uses the I/O Read command to read data from an agent mapped in the I/O address space. All 64 address bits are decoded.

#### 2.1.2.4 I/O Write Command

The SYM53C1010 uses the I/O Write command to write data to an agent mapped in the I/O address space. All 64 address bits are decoded.

#### 2.1.2.5 Reserved Command

The given bus encoding is reserved.

#### 2.1.2.6 Memory Read Command

The SYM53C1010 uses the Memory Read command to read data from an agent mapped in the Memory Address Space. The target may perform an anticipatory read if such a read produces no side effects.

#### 2.1.2.7 Memory Write Command

The SYM53C1010 uses the Memory Write command to write data to an agent mapped in the Memory Address Space. When the target returns “ready”, it assumes responsibility for data coherency, which includes ordering.

#### 2.1.2.8 Configuration Read Command

The SYM53C1010 uses the Configuration Read command to read the configuration space of each agent. An agent is selected during a configuration access when its IDSEL signal is asserted and AD[1:0] are 0b00. During the address phase of a configuration cycle, AD[7:2] address one of the 64 Dword registers in the configuration space of each device. Byte enables address the individual bytes within each Dword. AD[10:8] indicate which device of a multifunction agent is being addressed. AD[63:11] are logical don’t cares to the selected agent.

#### 2.1.2.9 Configuration Write Command

The SYM53C1010 uses the Configuration Write command to transfer data to the configuration space of each agent. An agent is selected when

its IDSEL signal is asserted and AD[1:0] are 0b00. During the address phase of a configuration cycle, the AD[7:2] lines address the 64 Dword registers in the configuration space of each device. Individual byte enable commands address the bytes within each Dword. AD[63:11] are logical don't cares to the selected agent. AD[10:8] indicate which device of a multifunction agent is addressed.

#### 2.1.2.10 Memory Read Multiple Command

This command is identical to the Memory Read command, except it additionally indicates that the master intends to fetch multiple cache lines before disconnecting. The SYM53C1010 supports PCI Memory Read Multiple functionality and issues Memory Read Multiple commands on the PCI bus when the Read Multiple mode is enabled. This mode is enabled by setting bit 2 (ERMP) of the DMA Mode (DMODE) register. If the cache mode is enabled, a Memory Read Multiple command is issued on all read cycles, except opcode fetches, when the following conditions are met:

- The CLSE bit (Cache Line Size Enable, bit 7, DMA Control (DCNTL) register) is set.
- The ERMP bit (Enable Read Multiple, bit 2, DMA Mode (DMODE) register) is set.
- The Cache Line Size (CLS) register for each function contains a legal burst size value (4, 8, 16, 32, 64, or 128 Dwords) that is less than or equal to the DMODE burst size.
- The transfer crosses a cache line boundary.

When these conditions are met, the chip issues a Memory Read Multiple command instead of a Memory Read during all PCI read cycles.

**Burst Size Selection –** The Read Multiple command reads in multiple cache lines of data during a single bus ownership. Revision 2.2 of the PCI specification specifies the number of cache lines to read as a multiple of the cache line size. The logic selects the largest multiple of the cache line size based on the amount of data to transfer. The maximum allowable burst size is determined from the DMA Mode (DMODE) burst size bits and the Chip Test Five (CTEST5) register, bit 2.

#### 2.1.2.11 Dual Address Cycle (DAC) Command

When 64-bit addressing is required, the SYM53C1010 performs DACs, per the PCI 2.2 specification. If any of the selector registers contain a nonzero value, a DAC is generated.

#### 2.1.2.12 Memory Read Line Command

This command is identical to the Memory Read command, except it additionally indicates that the master intends to fetch a complete cache line. This command is intended for use with bulk sequential data transfers where the memory system and the requesting master might gain some performance advantage by reading to a cache line boundary rather than a single memory cycle. The Read Line function in the SYM53C1010 takes advantage of the PCI 2.2 specification regarding issuance of this command.

If the cache mode is disabled, no Read Line commands are issued.

If the cache mode is enabled, a Read Line command is issued on all read cycles, except nonprefetch opcode fetches, when the following conditions are met:

- The CLSE bit (Cache Line Size Enable, bit 7, of the DMA Control (DCNTL) register) is set.
- The ERL bit (Enable Read Line, bit 3, of the DMA Mode (DMODE) register) is set.
- The Cache Line Size (CLS) register for each function must contain a legal burst size value (4, 8, 16, 32, 64, or 128 Dwords) that is less than or equal to the DMODE burst size.
- The transfer crosses a Dword boundary but not a cache line boundary.

When these conditions are met, the chip issues a Read Line command instead of a Memory Read during all PCI read cycles. Otherwise, it issues a normal Memory Read command.

**Read Multiple with Read Line Enabled –** When both the Read Multiple and Read Line modes are enabled, the Read Line command is not issued if the above conditions are met. Instead, a Read Multiple command is issued.

If the Read Multiple mode is enabled, Read Multiple commands are issued if the Read Multiple conditions are met.

### 2.1.2.13 Memory Write and Invalidate Command

The Memory Write and Invalidate command is identical to the Memory Write command, except it additionally guarantees a minimum transfer of one complete cache line. That is, the master intends to write all bytes within the addressed cache line in a single PCI transaction unless interrupted by the target. This command requires implementation of the PCI Cache Line Size (CLS) register. The SYM53C1010 enables Memory Write and Invalidate cycles when bit 0 (WRIE), in the Chip Test Three (CTEST3) register, and bit 4 (WIE), in the PCI Command register, are set.

When the following conditions are met, Memory Write and Invalidate commands are issued:

- The following bits are set:
  - the CLSE bit (Cache Line Size Enable, bit 7, of the DMA Control (DCNTL) register),
  - the WRIE bit (Write and Invalidate Enable, bit 0, of the Chip Test Three (CTEST3) register),
  - bit 4 of the PCI configuration Command register.
- The Cache Line Size (CLS) register for each function contains a legal burst size value (4, 8, 16, 32, 64, or 128 Dwords) that is less than or equal to the DMA Mode (DMODE) burst size.
- The chip has enough bytes in the DMA FIFO to complete at least one full cache line burst.
- The chip is aligned to a cache line boundary.

When these conditions are met, the SYM53C1010 issues a Write and Invalidate command instead of a Memory Write command during all PCI write cycles.

**Multiple Cache Line Transfers –** The Memory Write and Invalidate command can write multiple cache lines of data in a single bus ownership. The chip issues a burst transfer as soon as it reaches a cache line boundary. The transfer size is not automatically the cache line size, but rather a multiple of the cache line size specified in Revision 2.2

of the PCI specification. The logic selects the largest multiple of the cache line size based on the transfer size. The maximum allowable burst size is determined from the DMA Mode (DMODE) burst size bits, and bit 2 of the Chip Test Five (CTEST5) register. If multiple cache line size transfers are not desired, set the DMODE burst size to exactly the cache line size and the chip will only issue single cache line transfers.

After each data transfer, the chip re-evaluates the burst size based on the amount of remaining data to transfer. It again selects the highest possible multiple of the cache line size, and no larger than the DMA Mode (DMODE) burst size. Usually, the chip selects the DMODE burst size after alignment and issues bursts of this size. The burst size is, in effect, throttled down toward the end of a long Memory Move or Block Move transfer until only the cache line size left is burst size. The chip finishes the transfer with this burst size.

**Latency –** In accordance with the PCI specification, the latency timer is ignored when issuing a Memory Write and Invalidate command. Therefore, when a latency time-out occurs, the SYM53C1010 continues to transfer up to a cache line boundary. At that point, the chip relinquishes the bus, and finishes the transfer at a later time using another bus ownership. If the chip is transferring multiple cache lines it continues to transfer until the next cache boundary is reached.

**PCI Target Retry –** A retry is defined as a STOP with no TRDY/, indicating that no data was transferred. If the target issues a retry during a Memory Write and Invalidate transfer, the chip relinquishes the bus and immediately tries to finish the transfer on another bus ownership. The chip issues another Memory Write and Invalidate command on the next ownership, in accordance with the PCI specification.

**PCI Target Disconnect –** If the target device issues a disconnect during a Memory Write and Invalidate transfer, the SYM53C1010 relinquishes the bus and immediately tries to finish the transfer on another bus ownership. The chip does not issue another Memory Write and Invalidate command on the next ownership unless the address is aligned.

### 2.1.3 Internal Arbiter

The PCI to SCSI controller uses a single REQ/ - GNT/ signal pair to arbitrate for access to the PCI bus. An internal arbiter circuit allows the different bus mastering functions resident in the chip to arbitrate among

themselves for the privilege of arbitrating for PCI bus access. There are two independent bus mastering functions inside the SYM53C1010, one for each of the SCSI functions.

The internal arbiter uses a round robin arbitration scheme to decide which internal bus mastering function may arbitrate for access to the PCI bus. This ensures that no function is starved for access to the PCI bus.

## 2.1.4 PCI Cache Mode

The SYM53C1010 supports the PCI specification for an 8-bit Cache Line Size (CLS) register located in the PCI configuration space. The Cache Line Size (CLS) register provides the ability to sense and react to nonaligned addresses corresponding to cache line boundaries. In conjunction with the Cache Line Size (CLS) register, the PCI commands Memory Read Line (MRL), Memory Read Multiple (MRM), and Memory Write and Invalidate (MWI) are individually software enabled or disabled. Information on PCI cache mode alignment is provided in Table 2.2.

### 2.1.4.1 Enabling Cache Mode

To enable the cache logic to issue PCI cache commands (Memory Read Line, Memory Read Multiple, and Memory Write and Invalidate) on any PCI master operation, the following conditions must be met:

- The Cache Line Size Enable bit in the DMA Control (DCNTL) register must be set.
- The PCI Cache Line Size (CLS) register must contain a valid binary cache size, i.e., 4, 8, 16, 32, 64, or 128 Dwords. These values are the only valid cache sizes.
- The programmed burst size (in Dwords) must be equal to or greater than the cache line size register. The DMA Mode (DMODE) register, bits [7:6], and the Chip Test Five (CTEST5) register, bit 2, denote the burst length.
- The device must be performing a PCI Master transfer. The following PCI Master transactions do not utilize the PCI cache logic so no PCI cache commands are issued during these types of cycles: a nonprefetch SCRIPTS fetch, a Load/Store data transfer, and a data flush operation. All other types of PCI Master transactions utilize the PCI cache logic.

Not only must the above four conditions be met in order for the cache logic to control the type of PCI cache command that is issued, proper alignment is also necessary during write operations. If these conditions are not met for any given PCI Master transaction, a Memory Read or Memory Write is issued and no cache write alignment is done.

#### 2.1.4.2 Issuing Cache Commands

In order to issue each type of PCI cache command, the corresponding enable bit(s) must be set.

- To issue Memory Read Line commands, the Enable Read Line (ERL) bit in the DMA Mode (DMODE) register must be set.
- To issue Memory Read Multiples, the Enable Read Multiple (ERMP) bit in the DMA Mode (DMODE) register must be set.
- To issue Memory Write and Invalidates, both the Write and Invalidate Enable (WRIE) bit in the Chip Test Three (CTEST3) register and the Write and Invalidate Enable (WIE) bit in the PCI configuration Command register must be set.

If the corresponding cache command is not enabled, the cache logic falls back to the next command enabled. For example, if the Memory Read Multiple command is not enabled and the Memory Read Line command is, Memory Read Line command is issued in place of Memory Read Multiple command. If no cache commands are enabled, cache write alignment still occurs but no cache commands are issued; only Memory Reads and Memory Writes are issued.

#### 2.1.4.3 Memory Read Caching

The type of Memory Read command issued depends on the starting location of the transfer and the number of bytes to be transferred. During reads, no cache alignment is done, as it is neither required nor optional according to PCI 2.2 specification. Reads are a programmed burst length in size, as set in the DMA Mode (DMODE) and Chip Test Five (CTEST5) registers. In the case of a transfer that is smaller than the burst length, all bytes for that transfer are read in one PCI burst transaction. If the transfer crosses a Dword boundary (A[1:0] = 0b00) a Memory Read Line command is issued. If the transfer crosses a cache boundary, as specified by the cache line size programmed into the PCI configuration register, a Memory Read Multiple command is issued. If a transfer does

not cross a Dword or cache boundary or if cache mode is not enabled a Memory Read command is issued.

#### 2.1.4.4 Memory Write Caching

Memory Writes are aligned in a single burst transfer to reach a cache boundary. At that point, Memory Write and Invalidate commands are issued and continue at the burst length programmed into the DMA Mode (DMODE) register. Memory Write and Invalidate commands continue to be issued as long as the remaining byte count is greater than the Memory Write and Invalidate threshold. When the remaining byte count drops below this threshold, a single Memory Write burst is issued to complete the transfer.

In summary, the general pattern for PCI writes is:

- A single Memory Write to align to a cache boundary
- Multiple Memory Write and Invalidates
- A single data residual Memory Write to complete the transfer

**Table 2.2 PCI Cache Mode Alignment**

| Host Memory | | | | |
|---|---|---|---|---|
| | | A | | 0x00 |
| | B | | | 0x04 |
| | | | | 0x08 |
| | | C | | 0x0C |
| | | D | | 0x10 |
| | | | | 0x14 |
| | | | | 0x18 |
| | | | | 0x1C |
| | | E | | 0x20 |
| | | | | 0x24 |
| | | | | 0x28 |
| | | | | 0x2C |
| | | | F | 0x30 |
| | | | | 0x34 |
| | | | | 0x38 |
| | | | | 0x3C |
| | G | | | 0x40 |
| | | | | 0x44 |
| | | | | 0x48 |
| | | | | 0x4C |
| | | H | | 0x50 |
| | | | | 0x54 |
| | | | | 0x58 |
| | | | | 0x5C |
| | | | | 0x60 |

### 2.1.4.5 Examples

MR = Memory Read, MRL = Memory Read Line, MRM = Memory Read, Multiple, MW = Memory Write, MWI = Memory Write and Invalidate.

**Read Example 1 –** Burst = 4 Dwords, Cache Line Size = 4 Dwords:

**A to B:** MRL (6 bytes)

**A to C:** MRL (13 bytes)

**A to D:** MRM (16 bytes)
MR (1 byte)

**C to D:** MRM (5 bytes)

**C to E:** MRM (16 bytes)
MRM (5 bytes)

**D to F:** MRM (16 bytes)
MRM (16 bytes)

**A to H:** MRM (16 bytes)
MRM (16 bytes)
MRM (16 bytes)
MRM (16 bytes)
MRM (16 bytes)
MR (1 byte)

**A to G:** MRM (16 bytes)
MRM (16 bytes)
MRM (16 bytes)
MRM (16 bytes)
MR (2 bytes)

**Read Example 2 –** Burst = 8 Dwords, Cache Line Size = 4 Dwords:

**A to B:** MRL (6 bytes)

**A to C:** MRL (13 bytes)

**A to D:** MRM (17 bytes)

**C to D:** MRM (5 bytes)

**C to E:** MRM (21 bytes)

**D to F:** MRM (32 bytes)

**A to H:** MRM (32 bytes)
MRM (32 bytes)
MRM (17 bytes)

**A to G:** MRM (32 bytes)
MRM (32 bytes)
MR (2 bytes)

**Read Example 3 –** Burst = 16 Dwords, Cache Line Size = 8 Dwords:

**A to B:** MRL (6 bytes)

**A to C:** MRL (13 bytes)

**A to D:** MRL (17 bytes)

**C to D:** MRL (5 bytes)

**C to E:** MRM (21 bytes)

**D to F:** MRM (32 bytes)

**A to H:** MRM (64 bytes)
MRL (17 bytes)

**A to G:** MRM (64 bytes)
MR (2 bytes)

**Write Example 1 –** Burst = 4 Dwords, Cache Line Size = 4 Dwords:

**A to B:** MW (6 bytes)

**A to C:** MW (13 bytes)

**A to D:** MW (17 bytes)

**C to D:** MW (5 bytes)

**C to E:** MW (3 bytes)
MWI (16 bytes)
MW (2 bytes)

**D to F:** MW (15 bytes)
MWI (16 bytes)
MW (1 byte)

**A to H:** MW (15 bytes)
MWI (16 bytes)
MWI (16 bytes)
MWI (16 bytes)
MWI (16 bytes)
MW (2 bytes)

**A to G:** MW (15 bytes)
MWI (16 bytes)
MWI (16 bytes)
MWI (16 bytes)
MW (3 bytes)

**Write Example 2 –** Burst = 8 Dwords, Cache Line Size = 4 Dwords:

**A to B:** MW (6 bytes)

**A to C:** MW (13 bytes)

**A to D:** MW (17 bytes)

**C to D:** MW (5 bytes)

**C to E:** MW (3 bytes)
MWI (16 bytes)
MW (2 bytes)

**D to F:** MW (15 bytes)
MWI (16 bytes)
MW (1 byte)

**A to H:** MW (15 bytes)
MWI (32 bytes)
MWI (32 bytes)
MW (2 bytes)

**A to G:** MW (15 bytes)
MWI (32 bytes)
MWI (16 bytes)
MW (3 bytes)

**Write Example 3 –** Burst = 16 Dwords, Cache Line Size = 8 Dwords:

**A to B:** MW (6 bytes)

**A to C:** MW (13 bytes)

**A to D:** MW (17 bytes)

**C to D:** MW (5 bytes)

**C to E:** MW (21 bytes)

**D to F:** MW (32 bytes)

**A to H:** MW (31 bytes)
MWI (32 bytes)
MW (18 bytes)

**A to G:** MW (31 bytes)
MWI (32 bytes)
MW (3 bytes)

#### 2.1.4.6 Memory-to-Memory Moves

Memory-to-Memory moves also support PCI cache commands, as described above, with one limitation: Memory Write and Invalidate on Memory-to-Memory move writes are only supported if the source and destination address are quad word aligned. If the source and destination are not quad word aligned, i.e., Source Address[2:0] == Destination Address[2:0], write alignment is not performed and Memory Write and Invalidates are not issued.

The SYM53C1010 is little endian. This mode assigns the least significant byte to bits [7:0].

## 2.2 SCSI Functional Description

Both Ultra3 SCSI controllers on the SYM53C1010 provide a SCSI function that supports either an 8-bit or 16-bit bus. Each controller supports Wide Ultra3 SCSI synchronous transfer rates up to 160 Mbytes/s on a LVD SCSI bus. SCSI functions can be programmed with SCSI SCRIPTS, making it easy to “fine tune” the system for specific mass storage devices or Ultra3 SCSI requirements. Figure 2.1 illustrates the relationship between the SYM53C1010 modules.

The SYM53C1010 offers low level register access or a high level control interface. Like first generation SCSI devices, the SYM53C1010 is accessed as a register-oriented device. The ability to sample and/or assert any signal on the SCSI bus is used in error recovery and diagnostic procedures.

The SYM53C1010 is controlled by the integrated SCRIPTS processor through a high level logical interface. Commands controlling the SCSI functions are fetched out of the main host memory or local memory. These commands instruct the SCSI functions to Select, Reselect, Disconnect, Wait for a Disconnect, Transfer Information, Change Bus Phases, and implement all other aspects of the SCSI protocol. The SCRIPTS processor is a special high speed processor optimized for SCSI protocol.

### 2.2.1 SCRIPTS Processor

The SCSI SCRIPTS processor allows fetches of both DMA and SCSI commands from host memory or internal SCRIPTS RAM. Algorithms written in SCSI SCRIPTS control the actions of the SCSI and DMA cores. The SCRIPTS processor, running off of the PCI clock, executes complex SCSI bus sequences independently of the host CPU.

Algorithms can be designed to tune SCSI bus performance to adjust to new bus device types, such as scanners, communication gateways, etc. They can also incorporate changes in the SCSI logical bus definitions without sacrificing I/O performance. SCSI SCRIPTS are hardware independent, so they can be used interchangeably on any host or CPU system bus. SCSI SCRIPTS handle conditions such as Phase Mismatch.

#### 2.2.1.1 Phase Mismatch Handling in SCRIPTS

The SYM53C1010 can handle phase mismatches due to drive disconnects without needing to interrupt the processor. The primary goal of this logic is to completely eliminate the need for CPU intervention during an I/O disconnect/reselect sequence.

SCRIPTS control the storage of appropriate information needed to restart the I/O state, eliminating the need for processor intervention during an I/O disconnect/reselect sequence. Calculations are performed such that the appropriate information is available to SCRIPTS so that an I/O state can be properly stored for restart later.

The Phase Mismatch Jump logic is disabled at power-up. It must be enabled by setting the Phase Mismatch Jump Enable bit (ENPMJ, bit 7 in the Chip Control Zero (CCNTL0) register). Utilizing the information supplied in the Phase Mismatch Jump Address registers allows all overhead involved in a disconnect/reselect sequence to be handled with a modest amount of SCRIPTS instructions. These registers are described in detail in Chapter 4, “Registers”.

### 2.2.2 Internal SCRIPTS RAM

The SYM53C1010 has 8 Kbytes (2048 x 32 bits) of internal, general purpose RAM for each SCSI function. The RAM is designed for SCRIPTS program storage, but is not limited to this type of information. When the chip fetches SCRIPTS instructions or Table Indirect

information from the internal RAM, these fetches remain internal to the chip and do not use the PCI bus. In addition, any SCRIPTS instruction that contains a source or destination address residing in SCRIPTS RAM memory space remains internal to the chip and does not generate PCI cycles. SCRIPTS instructions able to access SCRIPTS RAM memory space in this manner include Memory-to-Memory moves, Load/Stores, and Block Moves. While an internal cycle is occurring, any external PCI slave cycle is retried on the PCI bus. Setting the DISRC (Disable Internal SCRIPTS RAM Cycles) bit in the Chip Control Zero (CCNTL0) register disables this feature.

SCRIPTS RAM should be initialized before it is read. Reading SCRIPTS RAM before initialization will result in the SCRIPTS RAM parity bit, bit 7, being set in the Shadowed SCSI SGE Status 0 register.

PCI system BIOS can relocate the RAM anywhere in the 64-bit address space. Base Address Register Three (BAR3) (SCRIPTS RAM) and Base Address Register Four (BAR4) (SCRIPTS RAM), in the PCI configuration space, contain the base address of the internal RAM. To simplify SCRIPTS instruction loading, the base address of the RAM appears in the Scratch Register B (SCRATCHB) register when bit 3 of the Chip Test Two (CTEST2) register is set. The upper 32 bits of a 64-bit base address are in the SCRIPT Fetch Selector (SFS) register. The RAM is byte accessible from the PCI bus and is visible to any bus mastering device on the bus. External, CPU accesses to the RAM follow the same timing sequence as a standard slave register access, except that the required target wait-states drop from 5 to 3. SCRIPTS RAM must first be written before being read in order to initialize SCRIPTS RAM parity. If a SCRIPTS RAM parity error is encountered a SCSI Gross Error interrupt will be signaled.

A complete set of development tools is available for writing custom drivers with SCSI SCRIPTS. For more information on the SCSI SCRIPTS instructions supported by the SYM53C1010, see Chapter 5, “SCSI SCRIPTS Instruction Set”.

### 2.2.3 64-Bit Addressing in SCRIPTS

The PCI interface for the SYM53C1010 provides 64-bit address and data capability in the initiator mode. The chip can also respond to 64-bit addressing in the target mode.

DACs can be generated for all SCRIPTS operations. There are six selector registers which hold the upper Dword of a 64-bit address. All but one of these is static and requires manual loading using a CPU access, a Load and Store instruction, or a Memory Move instruction. One of the selector registers is dynamic and is used during 64-bit direct block moves only. All selectors will default to zero, meaning the SYM53C1010 will power-up in a state where only Single Address Cycles (SACs) are generated. When any of the selector registers are written to a nonzero value, DACs are generated.

Direct, Table Indirect and Indirect Block Moves, Memory-to-Memory Moves, Load/Stores, and Jumps are all instructions with 64-bit address capability.

Note: Crossing the 4 Gbyte boundary on any one SCRIPTS operation is not permitted. Therefore, software must handle all such transactions.

### 2.2.4 Hardware Control of SCSI Activity LED

The SYM53C1010 controls an LED through the GPIO_0 pin to indicate that it is connected to the SCSI bus. This function was previously handled by a software driver.

Bit 3 (CON), in the Interrupt Status Zero (ISTAT0) register, is presented at the GPIO_0 pin when the following occurs:

- Bit 5 (LED_CNTL) in the General Purpose Pin Control (GPCNTL) register is set,
- Bit 6 (Fetch Enable) in the General Purpose Pin Control (GPCNTL) register is cleared,
- the SYM53C1010 is not performing an EEPROM autodownload.

The CON (Connected) bit in Interrupt Status Zero (ISTAT0) register is set anytime the SYM53C1010 is connected to the SCSI bus either as an initiator or a target. This happens after the SYM53C1010 has successfully completed a selection or when it has successfully responded to a selection or reselection. It will also be set when the SYM53C1010 wins arbitration in low level mode.

## 2.2.5 Designing an Ultra3 SCSI System

Software modifications are needed to take advantage of the Ultra3 speed in the SYM53C1010. Since Ultra3 SCSI is based on existing SCSI standards, it can use existing drivers if they are able to negotiate for Ultra3 synchronous transfer rates. Also, the target device must be able to communicate at Ultra3 speed. The SYM53C1010 uses Domain Validation to determine whether or not the system is capable of Ultra3 SCSI before activating DT clocking. Refer to Section 2.2.5.1, "Ultra160 SCSI Features," for more information on DT clocking.

LVD SCSI fulfills the hardware requirements for Ultra3 SCSI transfer rates, increased cable lengths, and additional devices on the bus. All devices on the bus must have LVD SCSI capabilities to guarantee Ultra3 SCSI transfer rates. For additional information on Ultra3 SCSI, refer to the SCSI Parallel Interface-3 (SPI-3) working document that is available on the world wide web at the T10 Home Page, http://www.t10.org. Also, check the SCSI Trade Association web site at http://www.scsita.org/. Chapter 6, "Specifications" contains Ultra3 SCSI timing information. In addition to the guidelines in the draft standard, make the following software adjustments to accommodate Ultra3 SCSI transfers.

### 2.2.5.1 Ultra160 SCSI Features

**Domain Validation –** A procedure where a host queries a device to determine its ability to communicate at the negotiated Ultra3 data rate. In software, the following steps are performed to ensure the selected device can successfully transfer data at the negotiated speed.

Step 1. Select a device.

Step 2. Issue Inquiry command.

Step 3. Issue Parallel Protocol Request (PPR) message.

Step 4. Issue Write Buffer command.

Step 5. Issue Read Buffer command.

Step 6. Examine the data pattern to ensure validity.

If the commands complete successfully with no CRC errors, bus hangs, or data pattern errors, then the negotiated speed is valid.

**CRC –** The error detecting code used in Ultra3 SCSI. Four bytes are transferred with data to increase the reliability of data transfers. CRC is used in the DT Data-In and DT Data-Out phases only. Because CRC is implied with DT mode and only works with DT mode, the DT setting can be used for CRC.

**DT Clocking –** Ultra3 SCSI implements DT clocking to provide speeds up to 80 megatransfers/s. DT clocking means that the data is sampled on both the asserting and deasserting edge of REQ/ACK. DT clocking is only valid using a LVD SCSI bus.

In order to support DT clocking, there are two new phases for the SCSI bus. The old Data-In and Data-Out phases are now called Single Transition (ST) Data-In and ST Data-Out. The new phases are DT Data-In and DT Data-Out. The use of DT and ST phases implies that the SCRIPTS engine may use a different jump point for DT or ST. Table 2.3 illustrates SCSI signal configuration for these phases.

**Table 2.3 New Phases on SCSI Bus**

| Phase | MSG | C/D | I/O | Description |
|---|---|---|---|---|
| ST Data-Out | 0 | 0 | 0 | – |
| ST Data-In | 0 | 0 | 1 | – |
| DT Data-Out | 1 | 0 | 0 | Previously reserved |
| DT Data-In | 1 | 0 | 1 | Previously reserved |

To indicate DT or ST mode, a bit is set in the current “selection” data reserved byte. BMOVE instructions identify the current BMOVE as either DT or ST through the phase bits.

### 2.2.5.2 Parallel Protocol Request (PPR)

CRC, Sync/Wide, DT, QAS, and “information units” are negotiated with a new SCSI extended message:

| | | |
|---|---|---|
| Byte 0 | 0x01 | Extended message |
| Byte 1 | 0x06 | Length |
| Byte 2 | 0x04 | PPR |
| Byte 3 | 0xXX | Transfer Period Factor |
| Byte 4 | 0x00 | Reserved |
| Byte 5 | 0xXX | REQ/ACK Offset |
| Byte 6 | 0xXX | Transfer Width Exponent |
| Byte 7 | 0x0X | Protocol options |

**Transfer Period Factor (Byte 3) –** Transfer Period Factor is the old Synchronous Period value. These are the same with one addition for 80 megatransfers/s rate:

| | | |
|---|---|---|
| 0x09 | = | 12.5 ns (Ultra3 SCSI) only valid when using DT |
| 0x0A | = | 25 ns (Ultra2 SCSI) |
| 0x0B | = | 30.3 ns |
| 0x0C | = | 50 ns (Ultra SCSI) |
| 0x0D–0xFF | = | *value* x 4 = *Period* in ns |

The transfer period is related to the data transfer speed, NOT the clock period. So, in DT mode, 0x09 means 12.5 ns between clock edges which really means a 25 ns clock period. In DT mode, 0x0A would mean a clock period of 50 ns but a data rate of 40 megatransfers/s (25 ns). In ST mode, 0x0A would mean a clock period of 25 ns and a data rate of 40 megatransfers/s.

**REQ/ACK Offset (Byte 5) –** REQ/ACK Offset is the maximum SCSI offset.

**Transfer Width Exponent (Byte 6) –** Transfer Width Exponent is the old width value. It is set to 0 (8-bit SCSI width) or 1 (16-bit SCSI width).

Note: For DT mode or when the Protocol Options field is nonzero, the Transfer Width Exponent must be one indicating a SCSI width of 16-bits.

The Table Indirect data (used during selection/reselection) must be updated to enable certain control bits in the SCSI Control Four (SCNTL4) register. Specific bits to look at include: bit 7, U3EN (Ultra3 Transfer Enable); bit 6, AIPEN (Asynchronous Information Protection Enable); and bits [3:0] (extra clock setup/hold).

The Protocol Options are described in Table 2.4.

**Table 2.4 Protocol Options (Byte 7)**

| OAS_REQ | DT_REQ | IU_REQ | Description |
|---|---|---|---|
| 0 | 0 | 0 | Use ST Data-In and ST Data-Out phase to transfer data |
| 0 | 1 | 0 | Use DT Data-In and DT Data-Out phase to transfer data with CRC |
| 0 | 1 | 1 | Use DT Data-In and DT Data-Out phase to transfer data with information units |
| 1 | 1 | 0 | Use DT Data-In and DT Data-Out phase to transfer data with CRC and use the QAS method for arbitration |
| 1 | 1 | 1 | Use DT Data-In and DT Data-Out phase to transfer data with information units and use the QAS method for arbitration |

A bus or device reset, power cycle, or change between LVD/SE modes invalidates these settings. A renegotiation resets the Protocol Options.

### 2.2.5.3 Asynchronous Information Protection (AIP)

The AIP feature provides error checking for asynchronous, nondata phases through BCH encoding. During the command, status, message in/out phases, the BCH code is transferred on the upper SCSI data bus. For details on the BCH code, see T10 119r3 document *Protection for the Asynchronous Phases*.

The AIP error status and the live AIP code values are captured in the AIP Control One (AIPCNTL1) register for debug purposes. AIP checking

and generation are enabled by setting bit 6 in the SCSI Control Four (SCNTL4) register.

The sequence ID is reset on any phase change, chip reset, bus free, or synchronous phase. It is also reset by writing the RSQAIP bit in the AIP Control Zero (AIPCNTL0) register. The AIP sequence value can be read using this register (the SEQAIP bits).

All AIP errors are treated in the same fashion as parity errors. Bit 0 of the SCSI Interrupt Status Zero (SIST0) register indicates if SCSI parity, CRC, or AIP errors are present. The LAIPERR bit in the AIP Control One (AIPCNTL1) register indicates if the error is an AIP error.

#### 2.2.5.4 Register Considerations

The following is a summary of the registers and bits required to enable Ultra3 SCSI on the SYM53C1010 device.

- The PCI Device ID register value must be 0x20.
- The PCI Max_Lat (ML) register contains a value of 0x12, indicating it requires the bus every 4.5 µs.
- The SCSI Control Zero (SCNTL0) register:
  - Bit 3, EPC (Enable Parity/CRC/AIP Checking) is set to enable the CRC feature.
  - Bit 1, AAP (Assert SATN/ on Parity/CRC/AIP Error), is set in the initiator mode to automatically assert SATN/ on the detection of an error.
- The SCSI Control One (SCNTL1) register:
  - Bit 5, DHP (Disable Halt on Parity/CRC/AIP Error or ATN) (Target Only), is set in accordance with user requirements. When bit 5 is cleared, a SCSI transfer halts if an error occurs. When bit 5 is set, a SCSI transfer continues if an error occurs.
- The SCSI Control Three (SCNTL3) register:
  - Bit 7 is now reserved. It was previously the Ultra Enable bit.
  - Bits [6:4], SCF[2:0] (Synchronous Clock Conversion Factor), select the divisor of the SCLK frequency. The SCLK is divided before its presentation to the synchronous SCSI control logic.

  - Bit 3, EWS (Enable Wide SCSI), is set to enable wide SCSI. Ultra3 requires wide SCSI. Therefore, this bit must be set during these transfers.
  - Bits [2:0] are reserved.
- The SCSI Transfer (SXFER) register:
  - Bits [7:6] are reserved.
  - Bits [5:0], MO[5:0] (Max SCSI synchronous offset), are set for the maximum offset.
- The SCSI Status Two (SSTAT2) register:
  - Bit 2 is reserved. HVD SCSI is not supported.
- The SCSI Interrupt Enable Zero (SIEN0) register:
  - Bit 0, PAR (SCSI Parity/CRC/AIP Error), is set to detect a Parity/CRC/AIP error while receiving or sending SCSI data. For more information, see SCSI Control One (SCNTL1), bit 5.
- The Chip Control Three (CCNTL3) register:
  - Bit 4, ENDSKEW (Enable REQ/ACK to Data skew control) is set to enable control of the relative skew between the SCSI REQ/ACK signal and the data signals.
  - Bits [3:2], DSKEW[1:0] (REQ/ACK – Data skew control), control the amount of skew between the SCSI REQ/ACK signal and the SCSI data signals. These bits are used for Ultra3 SCSI Domain Validation only and control the skew only if bit 4 is set.
  - Bits [1:0], LVDDL[1:0] (LVD Drive strength select), control the drive level of the LVD pad drivers. This feature is intended for use in Ultra3 SCSI Domain Validation testing environments only. These bits should be set to 0b00 during normal operation.
- The SCSI Control Four (SCNTL4) register:
  - Bit 7, U3EN (Ultra3 Transfer Enable) is set to enable Ultra3 transfers.
  - Bit 6, AIPEN (Asynchronous Information Protection Enable), is set to enable checking and generation of the upper byte lane of protection information during Command, Status and Message Phases.
  - Bit [5:4] are reserved.

- Bit 3, XCLKH_DT (Extra Clock of Data Hold on DT Transfer Edge) is set to add a clock of data hold to synchronous DT SCSI transfers on the DT edge.
- Bit 2, XCLKH_ST (Extra Clock of Data Hold on ST Transfer Edge) is set to add a clock of data hold to synchronous DT or ST SCSI transfers on the ST edge. This bit impacts both ST and DT transfers as it affects data hold to the ST edge.
- Bit 1, XCLKS_DT (Extra Clock of Data Setup on DT Transfer Edge) is set to add a clock of data setup to synchronous DT SCSI transfers on the DT edge. This bit only impacts DT transfers as it affects data setup to the DT edge.
- Bit 0, XCLKS_ST (Extra Clock of Data Setup on ST Transfer Edge) is set to add a clock of data setup to synchronous DT or ST SCSI transfers on the ST edge. This bit impacts both ST and DT transfers as it affects data setup to the ST edge.

Note: The XCLKH_DT, XCLKH_ST, XCLKS_DT, and XCLKS_ST bits do not affect CRC timings.

- The AIP Control Zero (AIPCNTL0) register:
  - Bit 7, FBAIP (Force Bad AIP Parity value), forces bad AIP values to be sent over the SCSI bus.
  - Bit 6, RSQAIP (Reset AIP Sequence value), resets the sequence value used in the protection code calculation.
  - Bits [5:4], SEQAIP (AIP Sequence value), contain the current AIP sequence value.
  - Bits [3:0] are reserved.
- The AIP Control One (AIPCNTL1) register:
  - Bit 7, AIPERR (AIP Error Status), indicates the live value of the AIP checking logic error status.
  - Bit 6, LAIPERR (Latched AIP Error Status), represents the latched version of the AIP checking logic.
  - Bits [5:0], AIPV (AIP Value), indicate the current, calculated protection code value.
- The CRC Pad Byte Value (CRCPAD) register:
  - Bits [15:0], the CRC Pad byte value, contain the value placed onto the bus for the CRC pad bytes.

- The CRC Control Zero (CRCCNTL0) register:
  - Bit 7, DISCRCCK (Disable CRC Checking), is set to cause the internal logic to not check or report CRC errors during Ultra3 transfers. The device continues to calculate and send CRC's as requested by the target per SPI-3 specification.
  - Bit 6, DISCRC (Disable CRC protocol checking) causes the SYM53C1010 to not check for a CRC request prior to a phase change on the SCSI bus. This condition creates a SCSI error condition and makes the device noncompliant with the SPI-3 specification. This bit should not be set under normal operating conditions.
  - Bit 5, RSTCRCINT (Reset CRC Interval Counter) resets the internal CRC interval counter to zero.
  - Bit 4 is reserved.
  - Bits [3:0], CRCINT[3:0] (CRC Request Interval (Target Mode only)), determine when a CRC request is sent by the device when operating in target mode and transferring data in DT Data-In or DT Data-Out Phases.
- The CRC Control One (CRCCNTL1) register:
  - Bit 7, CRCERR (CRC Error), indicates whether or not a CRC error has been detected during a DT Data-In SCSI transfer. This bit is independent of the DISCRCCHK bit setting. To clear this condition, either write this bit to a one or read the SCSI Interrupt Status Zero (SIST0) and SCSI Interrupt Status One (SIST1) registers. When CRC checking and the Parity/CRC/AIP Error interrupt are both enabled, CRCERR is mirrored in the SIST0 register, bit 0, as a Parity/CRC/AIP error.
  - Bit 6 is reserved.
  - Bit 5, ENAS (Enable CRC Auto Seed), is set to cause the CRC logic to automatically reseed itself after every CRC check performed during DT Data-In SCSI transfers. When this bit is cleared, the SCSI control logic controls when the CRC logic is reseeded.
  - Bit 4, TSTSD (Test CRC Seed), is set to cause the CRC logic to immediately reseed itself. This bit should never be set during normal operation as it may cause corrupt CRCs to be generated.

- Bit 3, TSTCHK (Test CRC Check), is set to cause the CRC logic to initiate a CRC check. This bit should never be set during normal operation as it creates spurious CRC errors.
- Bit 2, CRCADD (CRC Accumulate), is set to cause the CRC block to include the value present in the input register in the current CRC calculation. A new output CRC value results. This bit should not be set during normal operation as corrupt CRC values result.
- Bits [1:0], CRCDSEL[1:0] (CRC Data Register Selector), control the data visible in the CRC Data register (located at 0xE0).

• The CRC Data (CRCD) register:

- Bits [31:0] CRCDATA (CRC Data). The value in this register is dependent upon the setting of the CRCDSEL bits.

### 2.2.5.5 Using the SCSI Clock Quadrupler

The SYM53C1010 can quadruple the frequency of a 40 MHz SCSI clock, allowing the system to perform Ultra3 SCSI transfers. This option is user selectable with bit settings in the SCSI Test One (STEST1), SCSI Test Three (STEST3), and SCSI Control Three (SCNTL3) registers. At power-on or reset, the quadrupler is disabled and powered down. Follow these steps to use the clock quadrupler:

1. Set the SCLK Quadrupler Enable bit (SCSI Test One (STEST1) register, bit 3).
2. Do not poll bit 5 of the SCSI Test Four (STEST4) register. Bit 5 is reserved. Use a delay of 50 µs after the quadrupler enable bit is set in Step 1.
3. Halt the SCSI clock by setting the Halt SCSI Clock bit (SCSI Test Three (STEST3) register, bit 5).
4. Set the clock conversion factor using the SCF (Synchronous Clock Conversion Factor) field in the SCSI Control Three (SCNTL3) register.
5. Set the SCLK Quadrupler Select bit (SCSI Test One (STEST1), bit 2).
6. Clear the Halt SCSI Clock bit.

### 2.2.6 Prefetching SCRIPTS Instructions

The prefetch logic in the SYM53C1010 fetches 8 Dwords of instructions when enabled by setting the Prefetch Enable bit (bit 5) in the DMA Control (DCNTL) register. The maximum burst size that can be performed is automatically determined using the burst length values in the DMA Mode (DMODE) register. If the unit cannot perform bursts of at least four Dwords, it disables itself. While the chip is prefetching SCRIPTS instructions, it uses the PCI cache commands Memory Read Line and Memory Read Multiple, if PCI caching is enabled.

Note: This feature is only useful when fetching SCRIPTS instructions from main memory. Due to the short access time of SCRIPTS RAM, prefetching is not necessary when fetching instructions from SCRIPTS RAM.

To ensure the SYM53C1010 always operates from the current version of the SCRIPTS instruction, the contents of the prefetch unit may be flushed under certain conditions. The contents of the prefetch unit are automatically flushed under the following conditions:

- On every Memory Move instruction

  The Memory Move instruction is used to place modified code into memory. To assure the device executes recent modifications, the prefetch unit flushes its contents and reloads the code each time an instruction is issued. To avoid inadvertently flushing the prefetch unit contents, use the No Flush option for all Memory Move operations that do not modify code within the next 8 Dwords. For more information refer to Chapter 5, “SCSI SCRIPTS Instruction Set”.

- On every Store instruction

  The Store instruction may also be used to place modified code directly into memory. To avoid inadvertently flushing the prefetch unit contents use the No Flush option for all Store operations that do not modify code within the next 8 Dwords.

- On every write to the DMA SCRIPTS Pointer (DSP) register
- On all Transfer Control instructions, when the transfer conditions are met

This is necessary since the next instruction to be executed is not the sequential next instruction in the prefetch unit.

- When the Prefetch Flush bit (DMA Control (DCNTL) register, bit 6) is set

  The unit flushes whenever this bit is set. This bit is self-clearing.

### 2.2.7 Opcode Fetch Burst Capability

Setting the Burst Opcode Fetch Enable bit (bit 1) in the DMA Mode (DMODE) register (0x38) causes the SYM53C1010 to burst in the first two Dwords of all instruction fetches. If the instruction is a Memory-to-Memory Move, the third Dword is accessed in a separate ownership. If the instruction is an indirect type, the additional Dword is accessed in a subsequent bus ownership. If the instruction is a Table Indirect Block Move, the device uses two accesses, each a two Dword burst, to obtain the four Dwords required.

Note: This feature is only useful if Prefetching is disabled.

This feature is only useful if fetching SCRIPTS instructions from main memory. Due to the short access time of SCRIPTS RAM, burst opcode fetching is not necessary when fetching instructions from SCRIPTS RAM.

### 2.2.8 Load and Store Instructions

The SYM53C1010 supports the Load and Store instruction type, which simplifies data movement between memory and the internal registers. It also enables the chip to transfer bytes to addresses relative to the Data Structure Address (DSA) register. Load and Store data transfers to or from the SCRIPTS RAM remain internal to the chip and do not generate PCI bus cycles. While a Load/Store to or from SCRIPTS RAM is occurring, any external PCI slave cycles that occur are retried on the PCI bus. Setting the DISRC (Disable Internal SCRIPTS RAM Cycles) bit in the Chip Control Zero (CCNTL0) register disables this feature. For more information on the Load and Store instructions, refer to Chapter 5, “SCSI SCRIPTS Instruction Set”.

### 2.2.9 JTAG Boundary Scan Testing

With one exception, the SYM53C1010 includes support for JTAG boundary scan testing in accordance with the IEEE 1149.1 specification. The exception concerns the TST_RSTN pin. This pin must not be toggled as it will reset the JTAG TAP controller. For more information, refer to the BSDL (Boundary Scan Descriptor Language) file.

This device accepts all required boundary scan instructions including the optional CLAMP, HIGH-Z, and IDCODE instructions. The optional JTAG pin TRST is not implemented. Reset of the JTAG logic through the TAP controller occurs when TMS is held high for at least 5 TCK clock cycles.

The SYM53C1010 uses an 8-bit instruction register to support all boundary scan instructions. The data registers included in the device are the Boundary Data register, the IDCODE register, and the Bypass register. This device can handle a 20 MHz TCK frequency with all TAP pins having a 50% duty cycle.

### 2.2.10 Parity/CRC/AIP Options

The SYM53C1010 implements a flexible parity scheme that permits control of the parity sense, allows parity checking to be turned on or off, and can deliberately send a byte with bad parity over the SCSI bus. Table 2.5 defines the bits that are involved in parity control and observation. Table 2.6 describes the parity control function of the Enable Parity Checking and Assert SCSI Even Parity bits in the SCSI Control One (SCNTL1) register, bit 2.

SCRIPTS RAM must first be written before being read in order to initialize SCRIPTS RAM parity. If a SCRIPTS RAM parity error is encountered, a SCSI Gross Error interrupt will be signaled.

The SYM53C1010 supports CRC checking and generation in DT phases and CRC checking and generation during DT Data Transfers.

The new CRC registers are CRC Pad Byte Value (CRCPAD), CRC Control Zero (CRCCNTL0), CRC Control One (CRCCNTL1), CRC Data (CRCD), SCSI Control Zero (SCNTL0), bit 3; EPC (Enable Parity/CRC/AIP checking), bit 1; AAP (Assert SATN/ on Parity/CRC/AIP error); SCSI Control One (SCNTL1), bit 5; DHP (Disable Halt on

Parity/CRC/AIP Error or ATN), and SCSI Interrupt Enable Zero (SIEN0), bit 0, (SCSI Parity/CRC/AIP Error).

The new AIP registers are SCSI Control Zero (SCNTL0), AIP Control Zero (AIPCNTL0), and AIP Control One (AIPCNTL1).

**Table 2.5 Bits Used for Parity/CRC/AIP Control and Generation**

| Bit Name | Location | Description |
|---|---|---|
| AAP (Assert SATN/ on Parity/CRC/AIP Errors) | SCSI Control Zero (SCNTL0), Bit 1 | When this bit is set, the SYM53C1010 SCSI function automatically asserts the SATN/ signal upon detection of a parity, CRC, or AIP error. SATN/ is only asserted in initiator mode. |
| EPC (Enable Parity/CRC/AIP Checking) | SCSI Control Zero (SCNTL0), Bit 3 | When set, this bit enables parity checking on the SYM53C1010. The SYM53C1010 checks for odd parity. |
| Assert Even SCSI Parity | SCSI Control One (SCNTL1), Bit 2 | When set, this bit forces even SCSI parity on each byte sent to the SCSI bus from the SYM53C1010. |
| Disable Halt on SATN/ or Parity/CRC/AIP Error (Target Mode Only) | SCSI Control One (SCNTL1), Bit 5 | This bit determines if the SYM53C1010 should halt operations when a parity error is detected in target mode. |
| Enable Parity/CRC/AIP Error Interrupt | SCSI Interrupt Enable Zero (SIEN0), Bit 0 | This bit determines whether the SYM53C1010 generates an interrupt when it detects a SCSI Parity/CRC/AIP error. |
| Parity Error | SCSI Interrupt Status Zero (SIST0), Bit 0 | This status bit is set whenever the SYM53C1010 detects a Parity/CRC/AIP error on the SCSI bus. |
| Status of SCSI Parity Signal | SCSI Status Zero (SSTAT0), Bit 0 | This status bit represents the active HIGH current state of the SCSI SDP0 parity signal. |
| SCSI SDP1 Signal | SCSI Status Two (SSTAT2), Bit 0 | This bit represents the active HIGH current state of the SCSI SDP1 parity signal. |
| Latched SCSI Parity | SCSI Status Two (SSTAT2), Bit 3<br>SCSI Status One (SSTAT1), Bit 3 | These bits reflect the SCSI odd parity signal corresponding to the data latched into the SCSI Input Data Latch (SIDL) register. |
| Master Parity Error Enable | Chip Test Four (CTEST4), Bit 3 | This bit enables parity checking during PCI master data phases. |
| Master Data Parity Error | DMA Status (DSTAT), Bit 6 | This bit is set when the SYM53C1010, as a PCI master, detects a target device signaling a parity error during a data phase. |
| Master Data Parity Error Interrupt Enable | DMA Interrupt Enable (DIEN), Bit 6 | By clearing this bit, a Master Data Parity Error does not cause assertion of INTA/ (or INTB/) but the status bit is set in the DMA Status (DSTAT) register. |
| AIP Checking | SCSI Control Four (SCNTL4), Bit 6 | Setting this bit enables the AIP checking and generation of the upper byte lane of protection information during command, status, and message phases. |

**Table 2.5 Bits Used for Parity/CRC/AIP Control and Generation (Cont.)**

| Bit Name | Location | Description |
|---|---|---|
| CRC Request Pending | SCSI Control Zero (SCNTL0), Bit 2 | This bit indicates it is acceptable to force a CRC request. This bit will only be clear when a CRC request has been sent and no data has been transferred since the request. This bit may be used to prevent back-to-back CRC conditions. |
| Disable CRC Checking | CRC Control Zero (CRCCNTL0), Bit 7 | This bit is set to cause internal logic not to check or report CRC errors during Ultra3 transfers. |
| Disable CRC Protocol Checking | CRC Control Zero (CRCCNTL0), Bit 6 | This bit is set to cause the device to not check for a CRC request prior to a phase change on the SCSI bus. This condition normally causes a SCSI error condition. Note: Setting this bit makes the SYM53C1010 noncompliant to the SPI-3 specification. This bit should not be set under normal operating conditions. |
| CRC Reset Counter (Target Mode Only) | CRC Control Zero (CRCCNTL0), Bit 5 | When set, this bit resets the internal CRC interval counter to zero. |
| CRC Interval Counter (Target Mode Only) | CRC Control Zero (CRCCNTL0), Bits [3:0] | These bits determine when a CRC request is sent out by the device. The interval is only applicable when the device is operating in target mode and transferring data in DT Data-In or DT Data-Out phases. The intervals are provided, in bytes, as: 0x0 = disabled; 0x1 = 128; 0x2 = 256; 0x3 = 512; ... ; 0x9 = 32768; 0xA = 65536; 0xB–0xF = Reserved. |

**Table 2.6 SCSI Parity Errors and Interrupts**

| DHP[1] | PAR[2] | Description |
|---|---|---|
| 0 | 0 | Halts when a parity error occurs in the target or initiator mode and does NOT generate an interrupt. |
| 0 | 1 | Halts when a parity error occurs in the target mode and generates an interrupt in the target or initiator mode. |
| 1 | 0 | Does not halt in target mode when a parity error occurs until the end of the transfer. An interrupt is not generated. |
| 1 | 1 | Does not halt in target mode when a parity error occurs until the end of the transfer. An interrupt is generated. |

1. DHP = Disable Halt on SATN/ or Parity Error (bit 5, SCSI Control One (SCNTL1))
2. PAR = Parity Error (bit 0 SCSI Interrupt Enable One (SIEN1))

## 2.2.11 DMA FIFO

The DMA FIFO is 8 bytes wide by 112–115 transfers deep depending on the type and direction of data transfer. The DMA FIFO is illustrated in Figure 2.2. Small FIFO mode (112 bytes) is not supported by the SYM53C1010.

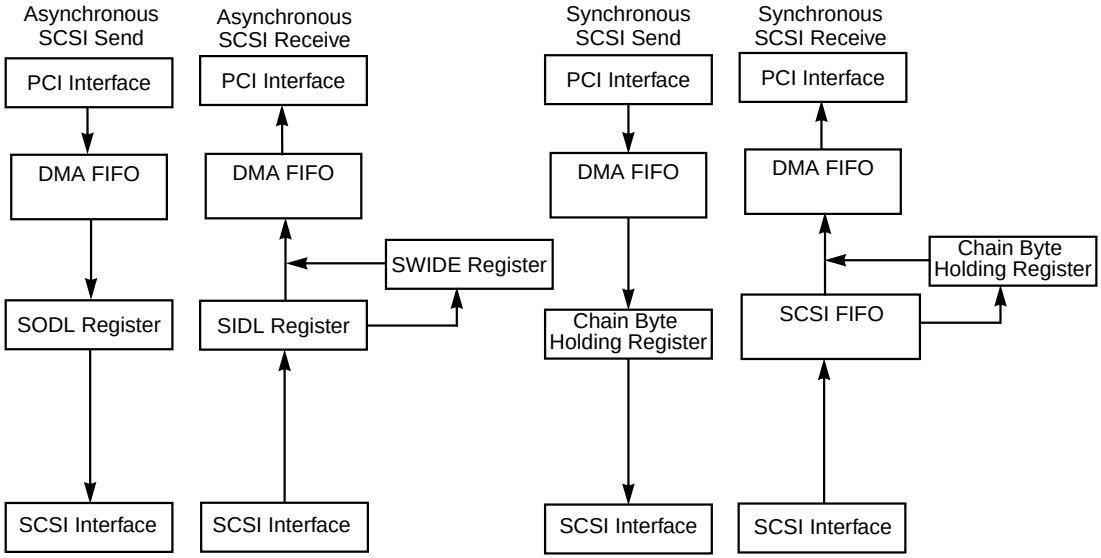
**Figure 2.2 DMA FIFO Sections**


The SYM53C1010 supports 64-bit memory and automatically supports misaligned DMA transfers. The FIFO allows the SYM53C1010 to support 4, 8, 16, 32, 64, or 128 Dword bursts across the PCI bus interface.

## 2.2.12 SCSI Data Paths

The data path through the SYM53C1010 is dependent on whether data is moved into or out of the chip and whether the SCSI data transfer is asynchronous or synchronous. Figure 2.3 illustrates how data is moved to and from the SCSI bus in each of the different modes. The following sections determine if any bytes remain in the data path when the device halts an operation.

**Figure 2.3 SYM53C1010 Host Interface SCSI Data Paths**




### 2.2.12.1 Asynchronous SCSI Send

To determine the number of bytes remaining in the DMA FIFO when a phase mismatch occurs, read the DMA FIFO Byte Count (DFBC) register. This 16-bit read only register contains the actual number of bytes remaining in the DMA FIFO. In addition, the SCSI Output Data Latch (SODL) register must be checked to determine if it contains any remaining bytes. If bit 5 (OLF) in the SCSI Status Zero (SSTAT0) register is set, then the least significant byte in the SODL register contains data. If bit 5 (OLF1) in the SCSI Status Two (SSTAT2) register is set, then the most significant byte in the SODL register contains data. Checking these bits also reveals bytes left in the SODL register from a Chained Move operation with an odd byte count. To recover from all other error conditions, the DMA FIFO should be cleared by setting bit 2 (CLF) in Chip Test Three (CTEST3) and the I/O should be retried.

If the Wide SCSI Send (WSS) bit in the SCSI Control Two (SCNTL2) register is set when a phase mismatch occurs, then adjustments must be made to the previous block move, not the current block move loaded into DCMD/DBC. To recover the byte of chain data in the SODL register the previous block move byte count should be set to 1 and the address set to the last data address for that block move.

#### 2.2.12.2 Synchronous SCSI Send

The DMA FIFO is the only location where data can reside when a phase mismatch occurs during a synchronous SCSI send transfer. To determine the number of bytes remaining in the DMA FIFO, read the DMA FIFO Byte Count (DFBC) register. This 16-bit, read only register contains the actual number of bytes remaining in the DMA FIFO. To recover from all other error conditions the DMA FIFO should be cleared by setting bit 2 (CLF) in Chip Test Three (CTEST3) and the I/O should be retried.

If the Wide SCSI Send (WSS) bit in the SCSI Control Two (SCNTL2) register is set when a phase mismatch occurs, then adjustments must be made to the previous block move, not the current block move loaded into DCMD/DBC. To recover the byte of chain data in the outbound chain byte holding register, the previous block move byte count should be set to one and the address set to the last data address for that block move.

#### 2.2.12.3 Asynchronous SCSI Receive

When a phase mismatch occurs during an asynchronous SCSI receive, the only data that may remain in the device is a potential wide residue byte in the SCSI Wide Residue (SWIDE) register. If bit 0 (WSR) in SCSI Control Two (SCNTL2) is set, then the SWIDE register contains a residual byte. This byte can be flushed by executing a block move instruction with a byte count of one. To recover from all other error conditions the DMA FIFO should be cleared by setting bit 2 (CLF) in Chip Test Three (CTEST3) and the I/O should be retried.

#### 2.2.12.4 Synchronous SCSI Receive

When a phase mismatch occurs during a synchronous SCSI receive transfer no data recovery operation is necessary. All data, including chain bytes from chained block moves, are flushed from the device prior to the phase mismatch occurring. To recover from all other error conditions, the DMA FIFO should be cleared by setting bit 2 (CLF) in Chip Test Three (CTEST3), the SCSI FIFO should be cleared by setting bit 1 (CSF) in SCSI Test Three (STEST3), and the I/O should be retried.

### 2.2.13 SCSI Bus Interface

The SYM53C1010 performs SE and LVD transfers.

### 2.2.13.1 SCSI Bus Modes

To increase device connectivity and SCSI cable length, the SYM53C1010 features LVD Link technology, the LSI Logic implementation of LVD SCSI. LVD Link transceivers provide the inherent reliability of differential SCSI and a long-term migration path for faster SCSI transfer rates.

HVD is not supported by this device. Bit 2 (DIFF) of the SCSI Status Two (SSTAT2) register and bit 5 (DIF) of the SCSI Test Two (STEST2) register are now reserved. The A_ or B_DIFFSENS signals still detect the different input voltages for HVD, LVD, and SE but the HVD feature is not present.

### 2.2.13.2 SCSI Termination

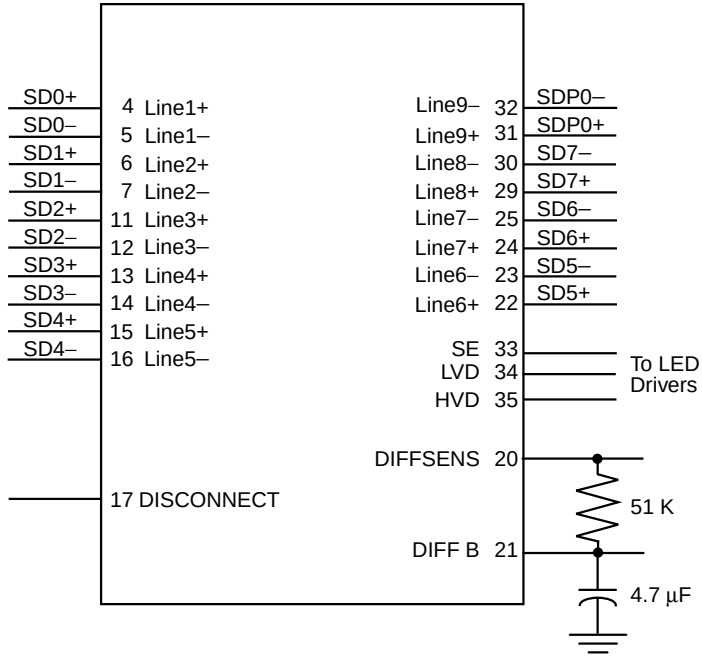
The terminator networks pull signals to an inactive voltage level and match the impedance seen at the end of the cable with the characteristic impedance of the cable. Terminators must be installed at the extreme ends of the SCSI chain, and only at the ends; no system should ever have more or less than two terminators. SCSI host adapters should provide a means of accommodating terminators. There should be a means of disabling the termination.

SE cables can use a 220 Ω pull-up resistor to the terminator power supply (Term Power) line and a 330 Ω pull-down resistor to ground. Because of the high-performance nature of the SYM53C1010, regulated (or active) termination is recommended. Figure 2.4 shows an active terminator. TolerANT technology active negation can be used with either termination network.

For information on terminators that support LVD, refer to the SPI-3 draft standard.

Note: If the SYM53C1010 is used in a design with an 8-bit SCSI bus, all 16 data lines must be terminated.

**Figure 2.4 Regulated Termination for Ultra3 SCSI**




DIFFSENS connects to the SCSI bus DIFFSENS line to detect what type of devices (SE, LVD, or HVD) are connected to the SCSI bus. DISCONNECT shuts down the terminator when it is not at the end of the bus. The disconnect pin low enables the terminator.

## 2.2.14 Select/Reselect During Selection/Reselection

In multithreaded SCSI I/O environments, it is not uncommon to be selected or reselected while trying to perform selection/reselection. This situation may occur when a SCSI controller (operating in the initiator mode) tries to select a target and is reselected by another. The Select SCRIPTS instruction has an alternate address to which the SCRIPTS will jump when this situation occurs. The analogous situation for target devices is being selected while trying to perform a reselection.

Once a change in operating mode occurs, either the initiator SCRIPTS issues a Set Initiator instruction or the target SCRIPTS issues a Set Target instruction. The Selection and Reselection Enable bits (SCSI Chip ID (SCID) bits 5 and 6, respectively) should both be asserted, enabling the SYM53C1010 to respond as an initiator or as a target. If only selection is enabled, the SYM53C1010 cannot be reselected as an

initiator. Status bits, in the SCSI Interrupt Status Zero (SIST0) register, and interrupt bits, in the SCSI Interrupt Enable Zero (SIEN0) register, indicate if the SYM53C1010 has been selected or reselected.

## 2.2.15 Synchronous Operation

The SYM53C1010 can transfer synchronous SCSI data in both the initiator and target modes. The SYM53C1010's SCLK input must be connected to a 40 MHz oscillator. The SCSI Transfer (SXFER) register controls the synchronous offset while the SCSI Control Three (SCNTL3) register controls the synchronous clock converters. These registers may be loaded by the CPU before SCRIPTS execution begins, from within SCRIPTS, with a Table Indirect I/O instruction, or with a Read-Modify-Write instruction.

The SYM53C1010 can receive data from the SCSI bus at a synchronous transfer period as short as 12.5 ns, regardless of the transfer period used to send data. The SYM53C1010 can receive data at one-fourth of the divided SCLK frequency. Depending on the SCLK frequency, the negotiated transfer period, and the synchronous clock divider, the SYM53C1010 can send synchronous data at intervals as short as 12.5 ns for Ultra3 SCSI, 25 ns for Ultra2 SCSI, 50 ns for Ultra SCSI, 100 ns for fast SCSI and 200 ns for SCSI-1.

### 2.2.15.1 Determining the Data Transfer Rate

Synchronous data transfer rates are controlled by bits in two different registers of the SYM53C1010. Following is a brief description of these bits. Figure 2.5 illustrates the clock division factors used in each register as well as the role of the register bits in determining the transfer rate.

### 2.2.15.2 SCSI Control Three (SCNTL3) Register, Bits [6:4] (SCF[2:0])

The SCF[2:0] bits select the factor by which the frequency of SCLK is divided before its presentation to the synchronous SCSI control logic.

The synchronous transfer speed is determined by the combination of the divided clock and the setting of the XCLKS_ST, XCLKS_DT, XCLKH_ST, and XCLKH_DT bits in the SCSI Control Four (SCNTL4) register. The table below gives the clock dividers available. Refer to Table 4.4, "Double Transition Transfer Rates", and Table 4.5, "Single Transition Transfer

Rates", located in the SCSI Control Four (SCNTL4) register description, for a full list of available synchronous transfer rates.

The SCF Divisor values are provided in Table 2.7.

**Table 2.7 SCF Divisor Values**

| SCF2 | SCF1 | SCF0 | SCLK Divisor |
|---|---|---|---|
| 0 | 0 | 0 | SCLK/3 |
| 0 | 0 | 1 | SCLK/1 |
| 0 | 1 | 0 | SCLK/1.5 |
| 0 | 1 | 1 | SCLK/2 |
| 1 | 0 | 0 | SCLK/3 |
| 1 | 0 | 1 | SCLK/4 |
| 1 | 1 | 0 | SCLK/6 |
| 1 | 1 | 1 | SCLK/8 |

### 2.2.15.3 SCSI Control Four (SCNTL4) Register, Bits [3:0]

The following extra clock bits add an extra clock of setup or hold to a ST or DT transaction.

Bit 3, XCLKH_DT (Extra Clock of Data Hold on DT transfer edge), adds a clock of data hold to synchronous DT SCSI transfers on the DT edge. This bit only impacts DT transfers as it only affects data hold to the DT edge. Setting this bit reduces the synchronous transfer send rate but will not reduce the rate at which the SYM53C1010 receives outbound REQs, ACKs, or data.

Bit 2, XCLKH_ST (Extra Clock of Data Hold on ST transfer edge), adds a clock of data hold to synchronous DT or ST SCSI transfers on the ST edge. This bit impacts DT and ST transfers as it affects data hold to the ST edge. Setting this bit reduces the synchronous transfer send rate but will not reduce the rate at which the SYM53C1010 receives outbound REQs, ACKs, or data.

Bit 1, XCLKS_DT (Extra Clock of Data Setup on DT transfer edge), adds a clock of data setup to synchronous DT SCSI transfers on the DT edge. This bit only impacts DT transfers as it only affects data hold to the DT

edge. Setting this bit reduces the synchronous transfer send rate but will not reduce the rate at which the SYM53C1010 receives outbound REQs, ACKs, or data.

Bit 0, XCLKS_ST (Extra Clock of Data Setup on ST transfer edge), adds a clock of data setup to synchronous DT or ST SCSI transfers on the ST edge. This bit impacts DT and ST transfers as it affects data hold to the ST edge. Setting this bit reduces the synchronous transfer send rate but will not reduce the rate at which the SYM53C1010 receives outbound REQs, ACKs, or data.

The synchronous receive rate can be calculated using the following formula:

$$\text{Receive Rate (DT)} = \frac{\text{Input Clock Rate}}{(\text{SCF Divisor} \times 2)} \quad \text{(megatransfers/s)}$$

$$\text{Receive Rate (ST)} = \frac{\text{Input Clock Rate}}{(\text{SCF Divisor} \times 4)} \quad \text{(megatransfers/s)}$$

Note: The receive rate is independent of the settings of the XCLKS_DT, XCLKS_ST, XCLKH_DT, XCLKH_ST bits.

The synchronous send rate, in units of megatransfers/s, can be calculated using the following formula:

$$\text{Send Rate (DT)} = \frac{\text{Input Clock Rate}}{\text{SCF Divisor} \times \left(2 + \frac{\text{XCLKS\_DT+XCLKS\_ST+XCLKH\_DT+XCLKH\_ST}}{2}\right)}$$

$$\text{Send Rate (ST)} = \frac{\text{Input Clock Rate}}{\text{SCF Divisor} \times (4 + \text{XCLKS\_ST+XCLKH\_ST})}$$

To configure the SYM53C1010 for Ultra3 DT transfers, perform the following steps:

Step 1. Enable the SCSI Clock Quadrupler – The SYM53C1010 can quadruple the frequency of a 40 MHz SCSI clock, allowing the system to perform Ultra3 SCSI transfers. This option is user selectable through bit settings in the SCSI Test One (STEST1) register. At power-up or reset, the quadrupler is disabled and powered down. Follow the steps in the bit description to enable the clock quadrupler.

Step 2. Program the Transfer Rate – Using SCSI Control Three (SCNTL3) and SCSI Control Four (SCNTL4), program the register to the 160 Mbytes/s transfer rate.

Step 3. Program the Maximum SCSI Offset – Using SCSI Transfer (SXFER), program the maximum SCSI DT Synchronous offset to 0x3E.

Step 4. Enable TolerANT – Set the TolerANT Enable bit, SCSI Test Three (STEST3), bit 7. Active negation must be enabled for the SYM53C1010 to perform Ultra3 SCSI transfers.

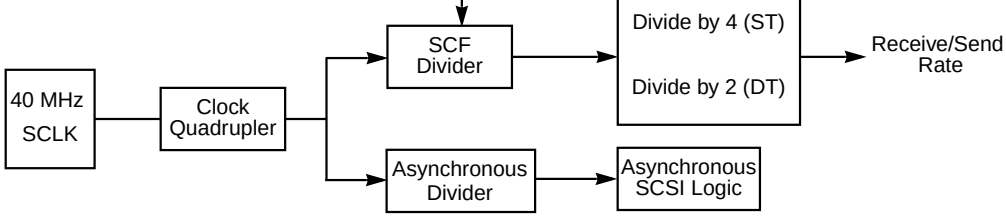
An example of configuring the Ultra3 SCSI transfer speed is:

1. Set SCNTL3 to 0x18.
2. Set SXFER to 0x3E.
3. Set SCNTL4 to 0x80.

These settings program the SYM53C1010 SCSI clocks to send and receive at 160 MHz with a synchronous SCSI offset of 0x3E.

**Figure 2.5 Determining the Synchronous Transfer Rate**

| SCF2 | SCF1 | SCF0 | SCF Divisor |
|---|---|---|---|
| 0 | 0 | 0 | 3 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1.5 |
| 0 | 1 | 1 | 2 |
| 1 | 0 | 0 | 3 |
| 1 | 0 | 1 | 4 |
| 1 | 1 | 0 | 6 |




$$\text{Received Rate (DT)} = \frac{\text{Input Clock Rate}}{(\text{SCF Divisor} \times 2)} \quad \text{(megatransfers/s)}$$

$$\text{Receive Rate (ST)} = \frac{\text{Input Clock Rate}}{(\text{SCF Divisor} \times 4)} \quad \text{(megatransfers/s)}$$

$$\text{Send Rate (DT)} = \frac{\text{Input Clock Rate}}{\text{SCF Divisor} \times \left(2 + \frac{\text{XCLKS\_DT+XCLKS\_ST+XCLKH\_DT+XCLKH\_ST}}{2}\right)}$$

$$\text{Send Rate (ST)} = \frac{\text{Input Clock Rate}}{\text{SCF Divisor} \times (4 + \text{XCLKS\_ST+XCLKH\_ST})}$$

## 2.2.16 Interrupt Handling

The SCRIPTS processors in the SYM53C1010 perform most functions independently of the host microprocessor. However, certain interrupt situations must be handled by the external microprocessor. This section explains all aspects of interrupts as they apply to the SYM53C1010.

### 2.2.16.1 Polling and Hardware Interrupts

The external microprocessor is informed of an interrupt condition by polling or hardware interrupts. Polling means that the microprocessor must continually loop and read a register until it detects a bit that is set

indicating an interrupt. This method is the fastest, but it diverts CPU time from other system tasks. The preferred method of detecting interrupts in most systems is hardware interrupts. In this case, the SYM53C1010 asserts the interrupt request (INTA/ or INTB/) line that interrupts the microprocessor, causing the microprocessor to execute an Interrupt Service Routine (ISR). A hybrid approach would use hardware interrupts for long waits and polling for short waits.

SCSI Function A is routed to PCI Interrupt INTA/. SCSI Function B is normally routed to INTB/, but can be routed to INTA/ if a pull-up is connected to MAD[4]. See Section 3.8, “MAD Bus Programming” for additional information.

#### 2.2.16.2 Registers

The registers in the SYM53C1010 used for detecting or defining interrupts are SCSI Interrupt Status Zero (SIST0), SCSI Interrupt Status One (SIST1), DMA Status (DSTAT), SCSI Interrupt Enable Zero (SIEN0), SCSI Interrupt Enable One (SIEN1), and DMA Interrupt Enable (DIEN). See the register descriptions in Chapter 4, “Registers” for additional information.

**ISTAT –** The ISTAT register includes the Interrupt Status Zero (ISTAT0), Interrupt Status One (ISTAT1), Mailbox Zero (MBOX0), and Mailbox One (MBOX1) registers. It is the only register that can be accessed as a slave during the SCRIPTS operation. Therefore, it is the register that is polled when polled interrupts are used. It is also the first register that should be read after the INTA/ (or INTB/) pin is asserted in association with a hardware interrupt.

The INTF (Interrupt-on-the-Fly) bit should be the first interrupt serviced. It must be written to one in order to clear it. This interrupt must be cleared before servicing any other interrupts indicated by SIP or DIP. Do not attempt to read the other chip status registers if the INTF bit is set, but SIP or DIP are not set.

If the SIP bit in the Interrupt Status Zero (ISTAT0) register is set, then a SCSI-type interrupt has occurred and the SCSI Interrupt Status Zero (SIST0) and SCSI Interrupt Status One (SIST1) registers should be read.

If the DIP bit in the Interrupt Status Zero (ISTAT0) register is set, then a DMA-type interrupt has occurred and the DMA Status (DSTAT) register should be read.

SCSI-type and DMA-type interrupts may occur simultaneously, so in some cases both SIP and DIP may be set. To avoid missing a SCSI interrupt the SCSI Interrupt Status Zero (SIST0) and SCSI Interrupt Status One (SIST1) registers should be read before the DMA Status (DSTAT) register is read.

When set, the SIRQD bit in Interrupt Status One (ISTAT1) disables the INT/ pin for the corresponding SCSI function. The interrupt is not lost or ignored but is merely masked at the pin. If the INT/ pin is already asserted when SIRQD is set the INT/ pin will remain asserted until the interrupt is serviced. Future interrupts will be masked at the pin until SIRQD is cleared.

Note that the host can read ISTAT as the SCRIPTS code is writing to ISTAT. In this case the data will be unstable so the read should be retried.

**SIST0 and SIST1 –** The SCSI Interrupt Status Zero (SIST0) and SCSI Interrupt Status One (SIST1) registers contain the status of SCSI-type interrupts whether they are enabled in SCSI Interrupt Enable Zero (SIEN0) and SCSI Interrupt Enable One (SIEN1) or not. Reading these registers determines the conditions that caused the SCSI-type interrupt, clears any bits that are set in SIST0 and SIST1, and clears the SIP bit in Interrupt Status Zero (ISTAT0). Since the SYM53C1010-33 SCSI functions stack interrupts, SIST0 and SIST1 are not necessarily cleared after a read; additional interrupts may still be pending.

If the SYM53C1010 is receiving data from the SCSI bus and a fatal interrupt condition occurs, the chip attempts to send the contents of the DMA FIFO to memory before generating the interrupt. Reading SCSI Interrupt Status Zero (SIST0) and SCSI Interrupt Status One (SIST1) will clear the CRC Error bit (bit 7) in the CRC Control One (CRCCNTL1) register.

If the SYM53C1010 is sending data to the SCSI bus and a fatal SCSI interrupt condition occurs, data could remain in the DMA FIFO. To determine if the DMA FIFO is empty, check the DMA FIFO Empty (DFE) bit in DMA Status (DSTAT) register. If this bit is cleared, set the CLF (Clear DMA FIFO) and CSF (Clear SCSI FIFO) bits before continuing.

The CLF bit is bit 2 in Chip Test Three (CTEST3) register. The CSF bit is bit 1 in SCSI Test Three (STEST3) register.

**DSTAT –** The DMA Status (DSTAT) register contains the status of DMA-type interrupts whether they are enabled in DIEN or not. Reading this register determines which condition(s) caused the DMA-type interrupt, clears any interrupt related bits in DSTAT, and clears the DIP bit in Interrupt Status Zero (ISTAT0). Since the SYM53C1010-33 SCSI functions stack interrupts, reading DSTAT does not necessarily clear the register as additional interrupts may be pending.

Bit 7 in DMA Status (DSTAT), DFE, is purely a status bit; it will not generate an interrupt and will not be cleared when read. DMA interrupts do not flush the DMA or SCSI FIFOs before generating the interrupt. Therefore, the DFE bit in the DSTAT register should be checked after any DMA interrupt. If the DFE bit is cleared, the FIFOs must either be cleared by setting the CLF (Clear DMA FIFO in CTEST3) and CSF (Clear SCSI FIFO in STEST3) bits, or flushed by setting the FLF (Flush DMA FIFO in CTEST3) bit.

**SIEN0 and SIEN1 –** The SCSI Interrupt Enable Zero (SIEN0) and SCSI Interrupt Enable One (SIEN1) registers are the interrupt enable registers for the SCSI interrupts in SCSI Interrupt Status Zero (SIST0) and SCSI Interrupt Status One (SIST1). Clearing the appropriate mask bit masks an interrupt.

**DIEN –** The DMA Interrupt Enable (DIEN) register is the interrupt enable register for DMA interrupts in DMA Status (DSTAT). Clearing the appropriate mask bit masks an interrupt.

#### 2.2.16.3 Fatal vs. Nonfatal Interrupts

A fatal interrupt, as the name implies, always causes the SCRIPTS to stop running. All nonfatal interrupts become fatal when they are enabled by setting the appropriate interrupt enable bit. Interrupt masking is discussed in Section 2.2.16.4, “Masking”. All DMA interrupts are fatal. The DMA interrupts are indicated by the DIP bit in Interrupt Status Zero (ISTAT0) and one or more bits in DMA Status (DSTAT).

Some SCSI interrupts are nonfatal. The SCSI interrupts are indicated by the SIP bit in the Interrupt Status Zero (ISTAT0) register and one or more

bits in SCSI Interrupt Status Zero (SIST0) register or SCSI Interrupt Status One (SIST1) register.

When the SYM53C1010 is operating in the Initiator mode, Interrupt-on-the-Fly, Function Complete (CMP), Selected (SEL), Reselected (RSL), General Purpose Timer Expired (GEN), and Handshake-to-Handshake Timer Expired (HTH) interrupts are nonfatal.

When operating in the Target mode, Interrupt-on-the-Fly, SATN/ active (M/A), CMP, SEL, RSL, GEN, and HTH are nonfatal. Refer to the description for the Disable Halt on a Parity/CRC/AIP Error or SATN/ active (Target Mode Only) bit, DHP, in the SCSI Control One (SCNTL1) register to configure the chip's behavior when the SATN/ interrupt is enabled during Target mode operation.

The reason for nonfatal interrupts is to prevent the SCRIPTS from stopping when an interrupt occurs that does not require service from the CPU. This prevents an interrupt when arbitration is complete (CMP set), when the SYM53C1010 is selected or reselected (SEL or RSL set), when the initiator asserts ATN (target mode: SATN/ active), or when the General Purpose or Handshake-to-Handshake timers expire. These interrupts are not needed for events that occur during high level SCRIPTS operation.

#### 2.2.16.4 Masking

Masking an interrupt means disabling or ignoring that interrupt. Clearing bits in the SCSI Interrupt Enable Zero (SIEN0) and SCSI Interrupt Enable One (SIEN1) registers will mask SCSI interrupts. Clearing bits in the DMA Interrupt Enable (DIEN) register will mask DMA interrupts. Masking an interrupt after INTA/ (or INTB/) is asserted does not cause INTA/ (or INTB/) to be negated. How the chip responds to masked interrupts depends on: whether polling or hardware interrupts are being used; whether the interrupt is fatal or nonfatal; and whether the chip is operating in the Initiator or Target mode.

If a nonfatal interrupt occurs while masked, SCRIPTS continues. The appropriate bit in the SCSI Interrupt Status Zero (SIST0) or SCSI Interrupt Status One (SIST1) is still set, the SIP bit in the Interrupt Status Zero (ISTAT0) is not set, and the INTA/ (or INTB/) pin is not asserted.

If a fatal interrupt occurs while masked, SCRIPTS halts. The appropriate bit in the DMA Status (DSTAT), SCSI Interrupt Status Zero (SIST0), or SCSI Interrupt Status One (SIST1) register is set, the SIP or DIP bit in the Interrupt Status Zero (ISTAT0) register is set, but the INTA/ (or INTB/) pin is not asserted.

Setting the SIRQD bit in the Interrupt Status One (ISTAT1) register disables the interrupt pin for the corresponding SCSI function. If an interrupt pin is already asserted and SIRQD is then set, the interrupt pin will remain asserted until serviced. Further interrupts will be blocked from the interrupt pin.

When the SYM53C1010 is initialized, enable all fatal interrupts if hardware interrupts are being used. If a fatal interrupt is disabled and that interrupt condition occurs, the SCRIPTS halts and the system never knows it unless it times out and checks the Interrupt Status Zero (ISTAT0), Interrupt Status One (ISTAT1), Mailbox Zero (MBOX0), and Mailbox One (MBOX1) registers after a certain period of inactivity.

If ISTAT is being polled instead of using hardware interrupts, then masking a fatal interrupt makes no difference since the SIP and DIP bits in the Interrupt Status Zero (ISTAT0) inform the system of interrupts, not the INTA/ (or INTB/) pin.

#### 2.2.16.5 Stacked Interrupts

The SYM53C1010 will stack interrupts, if they occur, one after the other. If the SIP or DIP bits in the Interrupt Status Zero (ISTAT0) register are set (first level), there is already at least one pending interrupt. Any future interrupts are stacked in extra registers behind the SCSI Interrupt Status Zero (SIST0), SCSI Interrupt Status One (SIST1), and DMA Status (DSTAT) registers (second level). When two interrupts have occurred and the two levels of the stack are full, any further interrupts set additional bits in the extra registers behind SIST0, SIST1, and DSTAT. When the first level of interrupts are cleared, all the later interrupts move into SIST0, SIST1, and DSTAT. After the first interrupt is cleared, the INTA/ (or INTB/) pin is deasserted for a minimum of three CLKs; the stacked interrupts move into SIST0, SIST1, or DSTAT; and the INTA/ (or INTB/) pin is asserted once again.

Since a masked nonfatal interrupt does not set the SIP or DIP bits, interrupt stacking does not occur. A masked, nonfatal interrupt still posts

the interrupt in SCSI Interrupt Status Zero (SIST0), but does not assert the INTA/ (or INTB/) pin. Since no interrupt is generated, future interrupts move into SCSI Interrupt Status Zero (SIST0) or SCSI Interrupt Status One (SIST1) instead of stacking behind another interrupt. When another interrupt condition occurs, the bit corresponding to the earlier masked nonfatal interrupt is set.

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

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

#### 2.2.16.6 Halting in an Orderly Fashion

When an interrupt occurs, the SYM53C1010 attempts to halt in an orderly fashion.

- If the interrupt occurs in the middle of an instruction fetch, the fetch is completed, except in the case of a Bus Fault. Execution does not begin, but the DSP points to the next instruction since it is updated when the current instruction is fetched.
- If the DMA direction is a write to memory and a SCSI interrupt occurs, the SYM53C1010 attempts to flush the DMA FIFO to memory before halting. Under any other circumstances, only the current cycle is completed before halting, so the DFE bit in DMA Status (DSTAT) should be checked to determine if any data remains in the DMA FIFO.
- SCSI SREQ/SACK handshakes that are in progress are completed before halting.
- The SYM53C1010 attempts to clean up any outstanding synchronous offset before halting.

- In the case of Transfer Control Instructions, once instruction execution begins it continues to completion before halting.
- In the case of a JUMP/CALL WHEN/IF <phase> instruction, the DMA SCRIPTS Pointer (DSP) is updated to the transfer address before halting.
- All other instructions may halt before completion.

#### 2.2.16.7 Sample Interrupt Service Routine

The following is a sample of an Interrupt Service Routine (ISR) for the SYM53C1010. It can be repeated if polling is used, or should be called when the INTA/ (or INTB/) pin is asserted if hardware interrupts are used.

1. Read Interrupt Status Zero (ISTAT0).
2. If the INTF bit is set, write it to a one to clear this status.
3. If only the SIP bit is set, read SCSI Interrupt Status Zero (SIST0) and SCSI Interrupt Status One (SIST1) to clear the SCSI interrupt condition and get the SCSI interrupt status. The bits in the SIST0 and SIST1 tell which SCSI interrupts occurred and determine what action is required to service the interrupts.
4. If only the DIP bit is set, read DMA Status (DSTAT) to clear the interrupt condition and determine the DMA interrupt status. The bits in the DSTAT register indicate which DMA interrupts occurred and determine what action is required to service the interrupts.
5. If both the SIP and DIP bits are set, read SCSI Interrupt Status Zero (SIST0), SCSI Interrupt Status One (SIST1), and DMA Status (DSTAT) to clear the SCSI and DMA interrupt condition and determine the interrupt status. If using 8-bit reads of the SIST0, SIST1, and DSTAT registers to clear interrupts, insert a 12 clock delay between the consecutive reads to ensure that the interrupts clear properly. Both the SCSI and DMA interrupt conditions should be handled before leaving the ISR. It is recommended that the DMA interrupt is serviced before the SCSI interrupt, because a serious DMA interrupt condition could influence how the SCSI interrupt is acted upon.
6. When using polled interrupts go back to Step 1 before leaving the ISR in case any stacked interrupts moved in when the first interrupt was cleared. When using hardware interrupts, the INTA/ (or INTB/)

pin is asserted again if there are any stacked interrupts. This should cause the system to re-enter the ISR.

### 2.2.17 Interrupt Routing

This section documents the recommended approach to RAID ready interrupt routing for the SYM53C1010. In order to be compatible with RAID upgrade products and the SYM53C1010, the following requirements must be met:

- When a RAID upgrade card is installed in the upgrade slot, interrupts from the motherboard SCSI controller(s) assigned to the RAID upgrade card must be routed to INTC/ and INTD/ of the upgrade slot and isolated from the motherboard interrupt controller. The system processor must not see interrupts from the SCSI controllers that are serviced by the RAID upgrade card. An upgrade slot is one that is connected to the interrupt routing logic for motherboard SCSI device(s). When a PCI RAID upgrade board is installed into the system, it will be plugged into this slot if it is to control motherboard SCSI device(s).
- The TDI pin of the upgrade slot must be connected to the INT_DIR/ pin of the SYM53C1010.
- When a RAID upgrade card is not installed, interrupts from a SCSI core must not be presented to the system's interrupt controller using multiple interrupt inputs.

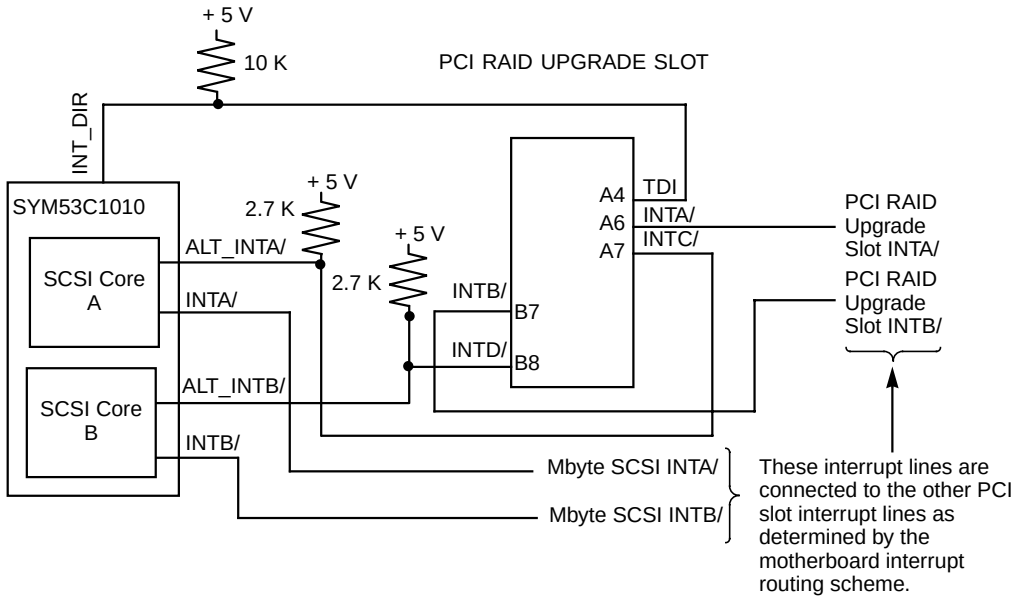
Figure 2.6 shows an example configuration. In this example the SYM53C1010 Dual Channel Ultra3 SCSI Multifunction Controller contains the interrupt routing logic.

The SYM53C1010 supports four different interrupt routing modes. Additional information for these modes may be found in the register description of SCSI Test One (STEST1) description in Chapter 4, "Registers". Each SCSI core within the chip may be configured independently by selecting the interrupt routing mode using bits [1:0] in the SCSI Test One (STEST1) register within each core. Mode 0 is the default mode and is compatible with RAID upgrade products.

If the implementation shown in Figure 2.6 is used, INTC/ and INTD/ of the PCI RAID upgrade slot cannot be used when a non-RAID upgrade card is installed in the slot. If this restriction is not acceptable, additional

buffer logic must be implemented on the motherboard. As long as the interrupt routing requirements stated above are satisfied, a motherboard designer could implement this design with external logic.

**Figure 2.6 Interrupt Routing Hardware Using the SYM53C1010**




There can only be one entity controlling a motherboard SCSI core or conflicts will occur. Typically a SCSI core will be controlled by the SCSI BIOS and an operating system driver. When a SCSI core is allocated to a RAID adapter, however, a mechanism must be implemented to prevent the SCSI BIOS and operating system driver from trying to access the SCSI core. The motherboard designer has several options to choose from for doing this:

- The **first option** is to have the SCSI core load its PCI Subsystem ID using a serial EPROM on power-up. If bit 15 in this ID is set, the LSI Logic Symbios® BIOS and operating system drivers will ignore the chip. This makes it possible to control the assignment of the motherboard SCSI cores using a configuration utility.
- The **second option** is to provide motherboard and system BIOS support for NVS. The SCSI core may then be enabled or disabled using the SCSI BIOS configuration utility. Not all versions of the Symbios drivers support this capability.

- The **third option** is to have the system BIOS not report the existence of the SCSI controller chips when the SCSI BIOS and operating systems make PCI BIOS calls. This approach requires modifications to the system BIOS and assumes the operating system uses PCI BIOS calls when searching for PCI devices.

## 2.2.18 Chained Block Moves

Since the SYM53C1010 has the capability to transfer 16-bit wide SCSI data, a unique situation occurs when dealing with odd bytes. The Chained Move (CHMOV) SCRIPTS instruction along with the Wide SCSI Send (WSS) and Wide SCSI Receive (WSR) bits in the SCSI Control Two (SCNTL2) register are used to facilitate these situations. The Chained Block Move instruction is illustrated in Figure 2.7.

### 2.2.18.1 Wide SCSI Send Bit

The WSS bit is set following a wide SCSI send operation (Data-Out for initiator mode or Data-In for target mode) when the SCSI core is holding a byte of chain data. The SCSI core holds the byte when the controller detects a partial transfer at the end of a chained Block Move SCRIPTS instruction. This flag is not set if a normal Block Move instruction is used. Under this condition, the SCSI controller does not send the low-order byte of the last partial memory transfer across the SCSI bus. Instead, the low-order byte is temporarily stored in the lower byte of the SCSI Output Data Latch (SODL) register for asynchronous transfers or in the chain byte holding register for synchronous transfers and the WSS flag is set.

The hardware uses the WSS bit to determine what behavior must occur at the start of the next data send transfer. If the WSS bit is set at the start of the next transfer, the first byte (the high-order byte) of the next data send transfer is “married” with the byte of chain data. The two bytes are sent out across the bus regardless of the type of Block Move instruction (normal or chained). The WSS bit is automatically cleared when the “married” word is sent. Performing either a SCSI receive operation or any narrow transfer also clears the bit. In addition, SCRIPTS and the microprocessor can clear the WSS bit as well as use it for error detection and recovery purposes.

#### 2.2.18.2 Wide SCSI Receive Bit

The WSR bit is set following a wide SCSI receive operation (Data-In for initiator mode or Data-Out for target mode) when the SCSI core is holding a byte of chain data. The SCSI core holds the byte when the controller detects a partial transfer at the end of a chained Block Move instruction. Under this condition the high-order byte is not transferred out of the DMA channel to memory. Instead, it is stored in the SCSI Wide Residue (SWIDE) register and the WSR flag is set.

The hardware uses the WSR bit to determine what behavior must occur at the start of the next data receive transfer. If set the stored high-order byte may be residual data, valid data for a subsequent data transfer, or overrun data. The byte may be read as normal by starting a data receive transfer. The WSR bit is automatically cleared at the start of the next data receive transfer. Performing either a SCSI send operation or any narrow transfer also clears the bit. In addition, SCRIPTS and the microprocessor can clear the WSR bit as well as use it for error detection and recovery purposes.

#### 2.2.18.3 SWIDE Register

For wide asynchronous receive data transfers, the SCSI Wide Residue (SWIDE) register holds the high-order byte of a partial SCSI transfer which has not yet been transferred to memory. This stored data may be a residue byte (and therefore ignored) or it may be valid data that is transferred to memory at the beginning of the next Block Move instruction.

#### 2.2.18.4 SODL Register

For wide asynchronous send data transfers, the low-order byte of the SCSI Output Data Latch (SODL) register holds the low-order byte of a partial memory transfer which has not yet been transferred across the SCSI bus. This stored data is usually “married” with the first byte of the next data send transfer, and both bytes are sent across the SCSI bus at the start of the next data send Block Move instruction.

#### 2.2.18.5 Chained Block Move SCRIPTS Instruction

A chained Block Move SCRIPTS instruction is primarily used to transfer consecutive data send or data receive blocks. Using the chained Block

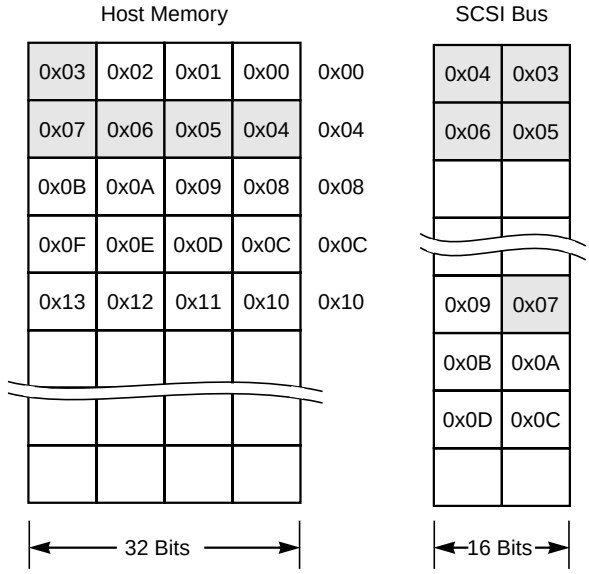
Move instruction facilitates partial receive transfers and allows correct partial send behavior without additional opcode overhead. Behavior of the chained Block Move instruction varies slightly for sending and receiving data.

For receive data (Data-In for the initiator or Data-Out for the target), a chained Block Move instruction indicates that if a partial transfer occurred at the end of the instruction the WSR flag is set. The high-order byte of the last SCSI transfer is stored in the SCSI Wide Residue (SWIDE) register rather than transferred to memory. The stored byte should be the first byte transferred to memory at the start of the chained Block Move or regular Block Move data stream. Since the byte count always represents data transfers to/from memory (as opposed to/from the SCSI bus), the stored byte transferred out is one of the bytes in the count. If the WSR bit is cleared when a receive data chained Block Move instruction is executed, the data transfer occurs similar to that of the regular Block Move instruction. It is recommended that all Block Move instructions be chained Block Moves.

For send data (Data-Out for the initiator or Data-In for the target), a chained Block Move instruction indicates that if a partial transfer terminates the chained block move the WSS flag is set. The low-order byte should be stored in the lower byte of the SCSI Output Data Latch (SODL) register for asynchronous transfers or in the outbound chain byte holding register for synchronous transfers and not sent across the SCSI bus. Without the chained Block Move instruction, the last low-order byte would be sent across the SCSI bus. The starting byte count represents data bytes transferred from memory but not to the SCSI bus when a partial transfer exists. For example, if the instruction is an initiator chained Block Move Data Out of five bytes (and WSS is not previously set), five bytes are transferred out of memory to the SCSI controller. Four bytes are transferred from the SCSI controller across the SCSI bus and one byte is temporarily stored as described above, waiting to be married with the first byte of the next Block Move instruction. If the WSS bit is set at the start of a data send command the first byte of the transfer is assumed to be the high-order byte and is “married” with the stored byte (which will be the low-order byte) before the two bytes are sent across the SCSI bus. It is recommended that all Block Move instructions be chained Block Moves.

Figure 2.7 provides examples of the Block Move and Chain Block Move operations.

**Figure 2.7 Block Move and Chained Block Move Instructions**




CHMOV 5, 3 when Data-Out

Moves five bytes from address 0x03 in the host memory to the SCSI bus. Bytes 0x03, 0x04, 0x05, and 0x06 are moved and byte 0x07 remains in the SCSI core (in the lower byte of the SCSI Output Data Latch (SODL) register for asynchronous transfers, in the outbound chain byte holding register for synchronous transfers). The stored byte is combined with the first byte of the following CHMOVE instruction.

Chain Move 0x5, 0x9 when Data-In

Moves five bytes from address 0x09 in the host memory to the SCSI bus. The data in address 0x09 is married with the stored data (0x07) and transferred to the SCSI bus.

## 2.3 Parallel ROM Interface

The SYM53C1010 supports up to 1 Mbyte of external memory in binary increments from 16 Kbytes to allow the use of expansion ROM for add-in PCI cards. Both functions of the device share the ROM interface. This interface is designed for low speed operations such as downloading

instruction code from ROM; it is not intended for dynamic activities such as executing instructions.

System requirements include the SYM53C1010, two or three external 8-bit address holding registers (HCT273 or HCT374), and the appropriate memory device. The 4.7 kΩ pull-up resistors on the MAD bus require HC or HCT external components to be used. Pull-up resistors on the 8-bit bidirectional memory bus at power-up determine the memory size and speed. The SYM53C1010 senses this bus shortly after the release of the Reset signal and configures the Expansion ROM Base Address (ERBA) register and the memory cycle state machines for the appropriate conditions.

The SYM53C1010 supports a variety of sizes and speeds of expansion ROM. An example set of interface drawings is in Appendix B, "External Memory Interface Diagram Examples". The encoding of pins MAD[3:1] allows the user to define how much external memory is available to the SYM53C1010. Table 2.8 shows the memory space associated with the possible values of MAD[3:1]. The MAD[3:1] pins are fully described in Chapter 3, "Signal Descriptions".

**Table 2.8 Parallel ROM Support**

| MAD[3:1] | Available Memory Space |
|---|---|
| 000 | 16 Kbytes |
| 001 | 32 Kbytes |
| 010 | 64 Kbytes |
| 011 | 128 Kbytes |
| 100 | 256 Kbytes |
| 101 | 512 Kbytes |
| 110 | 1024 Kbytes |
| 111 | No external memory present, ROM interface disabled |

To use one of the configurations mentioned above in a host adapter board design, put 4.7 kΩ pull-up resistors on the MAD pins corresponding to the available memory space. Each MAD pin has an internal static pull-down therefore no external pull-down resistors are

needed. For example, to connect to a 64 Kbytes external ROM use a pull-up on MAD[2]. If the external memory interface is not used, MAD[3:1] should be pulled high.

The SYM53C1010 allows the system to determine the size of the available external memory using the Expansion ROM Base Address (ERBA) register in the PCI configuration space. For more information on how this works, refer to the PCI specification or the Expansion ROM Base Address register description in Chapter 4, “Registers”.

MAD[0] is the slow ROM pin. When pulled up it enables two extra clock cycles of data access time to allow use of slower memory devices. The external memory interface also supports updates to flash memory.

## 2.4 Serial EEPROM Interface

For each SCSI function, the SYM53C1010 implements an interface permitting attachment of a serial EEPROM device to the GPIO[0] and GPIO[1] pins. There are two modes of operation relating to the serial EEPROM, the Subsystem ID (SID) register, and the Subsystem Vendor ID (SVID) register for each SCSI function. These modes are programmable through the MAD[7] pin, which is sampled at power-up.

### 2.4.1 Default Download Mode

In this mode, MAD[7] is pulled down internally, GPIO[0] is the serial data signal (SDA) and GPIO[1] is the serial clock signal (SCL). Certain data in the serial EEPROM is automatically loaded into chip registers at power-up.

The format of the serial EEPROM data is defined in Table 2.7. If the download is enabled and an EEPROM is not present or the checksum fails, the Subsystem ID (SID) and Subsystem Vendor ID (SVID) registers read back all zeros. At power-up, five bytes are loaded into the chip from locations 0xFB through 0xFF.

The Subsystem ID and Subsystem Vendor ID registers are read only in accordance with the PCI specification, with a default value of all zeros if the download fails.

Note: The speed of the serial EEPROM must be 400 Kbits/s.

**Table 2.9 Default Download Mode Serial EEPROM Data Format**

| Byte | Name | Description |
|---|---|---|
| 0xFB | SVID(0) | Subsystem Vendor ID (SVID), LSB. This byte is loaded into the least significant byte of the Subsystem Vendor ID register in the appropriate PCI configuration space at chip power-up. |
| 0xFC | SVID(1) | Subsystem Vendor ID (SVID), MSB. This byte is loaded into the most significant byte of the Subsystem Vendor ID register in the appropriate PCI configuration space at chip power-up. |
| 0xFD | SID(0) | Subsystem ID (SID), LSB. This byte is loaded into the least significant byte of the Subsystem ID register in the appropriate PCI configuration space at chip power-up. |
| 0xFE | SID(1) | Subsystem ID (SID), MSB. This byte is loaded into the most significant byte of the Subsystem ID register in the appropriate PCI configuration space at chip power-up. |
| 0xFF | CKSUM | Checksum (CKSUM). This 8-bit checksum is formed by adding, bytewise, each byte contained in locations 0xFB–0xFE to the seed value (0x55) and then taking the 2's complement of the result. |

### 2.4.2 No Download Mode

When MAD[7] is pulled up through an external resistor, the automatic download is disabled and data is not automatically loaded into chip registers at power-up. The Subsystem ID (SID) and Subsystem Vendor ID (SVID) registers are read only, per the PCI specification, with a default value of 0x1000 and 0x1000 respectively.

## 2.5 Power Management

The SYM53C1010 complies with the PCI Bus Power Management Interface Specification, Revision 1.1. The PCI Function Power States D0, D1, D2, and D3 are defined in that specification.

D0 is the maximum powered state, and D3 is the minimum powered state. Power state D3 is further categorized as D3hot or D3cold. A function that is powered off is said to be in the D3cold power state.

The SYM53C1010 power states are independently controlled through two power state bits that are located in the PCI Configuration Space Power Management Control/Status (PMCSR) register, 0x44–0x45. The power state bit settings are provided in Table 2.10.

**Table 2.10 Power States**

| Configuration Register (0x44), Bits [1:0] | Power State | Function |
|---|---|---|
| 00 | D0 | Maximum Power |
| 01 | D1 | Disables SCSI clock |
| 10 | D2 | Coma Mode |
| 11 | D3 | Minimum Power |

Although the PCI Bus Power Management Interface Specification does not allow power state transitions D2 to D1, D3 to D2, or D3 to D1, the SYM53C1010 hardware places no restriction on transitions between power states.

The PCI Function Power States D0, D1, D2, and D3 are described below in conjunction with each SCSI function. Power state actions are separate for each function.

As the device transitions from one power level to a lower one, the attributes that occur from the higher power state level are carried over into the lower power state level. For example, D1 disables the SCSI CLK. Therefore, D2 will include this attribute as well as the attributes defined in the Power State D2 section. The PCI Function Power States D0, D1, D2, and D3 are described below in conjunction with each SCSI function. Power state actions are separate for each function.

### 2.5.1 Power State D0

Power state D0 is the maximum power state and is the power-up default state for each function. The SYM53C1010 is fully functional in this state.

### 2.5.2 Power State D1

Power state D1 is a lower power state than D0. A function in this state places the SYM53C1010 core in the snooze mode and the SCSI clock

is disabled. In the snooze mode, a SCSI reset does not generate an IRQ/ signal.

### 2.5.3 Power State D2

Power state D2 is a lower power state than D1. A function in this state places the SYM53C1010 core in the coma mode. The following PCI Configuration Space Command register enable bits are suppressed:

- I/O Space Enable
- Memory Space Enable
- Bus Mastering Enable
- SERR/Enable
- Enable Parity Error Response

Thus, the function's memory and I/O spaces cannot be accessed, and the function cannot be a PCI bus master. Furthermore, SCSI and DMA interrupts are disabled when the function is in power state D2. If the function is changed from power state D2 to power state D1 or D0, the previous values of the PCI Command register are restored. Also, any pending interrupts before the function entered power state D2 are asserted.

### 2.5.4 Power State D3

Power state D3 is the minimum power state, which includes settings called D3hot and D3cold. D3hot allows the device to transition to D0 using software. The SYM53C1010 is considered to be in power state D3cold when power is removed from the device. D3cold can transition to D0 by applying $V_{CC}$ and resetting the device.

Power state D3 is a lower power level than power state D2. A function in this state places the SYM53C1010 core in the coma mode. Furthermore, the function's soft reset is continually asserted while in power state D3, which clears all pending interrupts and 3-states the SCSI bus. In addition, the function's PCI Command register is cleared. If both of the SYM53C1010 functions are placed in power state D3, the Clock Quadrupler is disabled, which results in additional power savings.

# Chapter 3
# Signal Descriptions

This chapter describes the input and output signals of the SYM53C1010-33. The chapter consists of the following sections:



## 3.1 Signal Organization

The SYM53C1010-33 has four major interfaces:

- PCI Interface
- SCSI Bus Interface
- Memory Interface
- Test Interface

Figure 3.1 illustrates the signals, their grouping, and their I/O direction. A slash (/) at the end of a signal name indicates that it is an active LOW signal. When the slash is absent, the signal is active at a HIGH voltage.

The PCI Interface contains many functional groups of signals. The SCSI Bus Interface contains two functional groups of signals as illustrated in Figure 3.1.

There are five signal type definitions:

**I** Input, a standard input-only signal.

**O** Output, a standard output driver (typically a Totem Pole output).

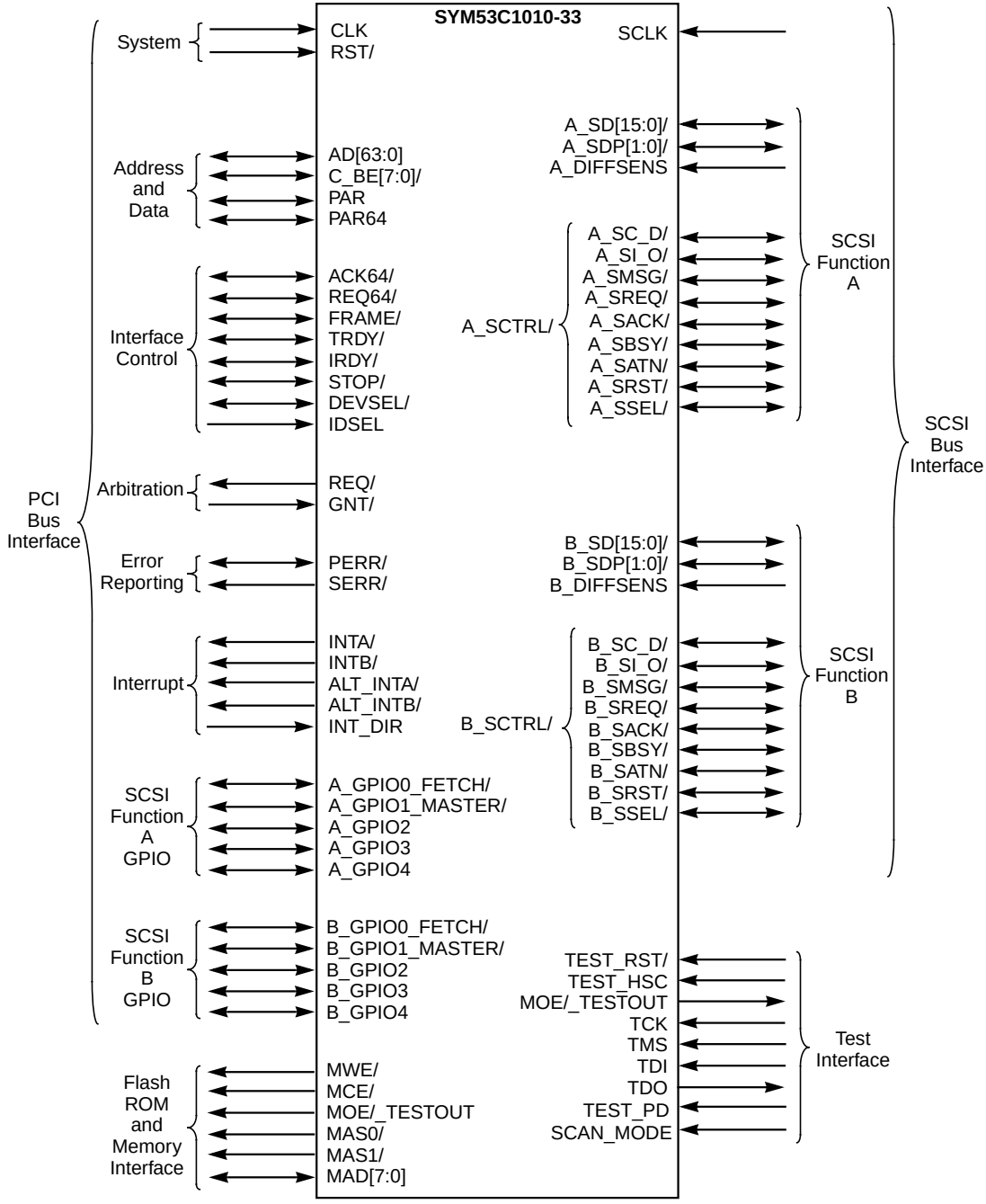
**I/O** Input and output (bidirectional).

**T/S** 3-state, a bidirectional, 3-state input/output signal.

**S/T/S** Sustained 3-state, an active low 3-state signal owned and driven by one and only one agent at a time.

This section of the chapter illustrates the signal groupings in Figure 3.1. Next, the signal locations on the 329 Ball Grid Array (BGA) are illustrated in Table 3.1 and Table 3.2. Then, the signal names are listed alphabetically, in Table 3.3, and numerically, in Table 3.5.

**Figure 3.1 SYM53C1010-33 Functional Signal Grouping**



## Table 3.1 Left Half of the SYM53C1010-33 329 BGA Chip - Top View

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | NC | TEST_PD | A_SD12+ | A_SD13+ | A_SD15+ | A_SD0+ | A_SD2– | A_SD4– | A_SD6– | A_SDP0– | $V_{DD}$– BIAS2 | A_SBSY+ | A |
| B | NC | $V_{DD}$_A | NC | A_SD13– | A_SD15– | A_SD0– | A_SD1+ | A_SD3+ | A_SD5+ | A_SD7+ | A_SATN– | A_SATN+ | B |
| C | TEST_ RST/ | $V_{SS}$_A | $V_{SS}$_IO | A_SD12– | A_SD14– | A_SDP1– | A_SD1– | A_SD2+ | A_SD4+ | A_SD6+ | A_SDP0+ | A_SBSY– | C |
| D | TCK | $V_{SS}$_ CORE | $V_{DD}$_ CORE | $V_{SS}$_IO | A_SD14+ | A_SDP1+ | $V_{DD}$_IO | A_SD3– | A_SD5– | $V_{DD}$_IO | A_SD7– | $V_{SS}$_IO | D |
| E | TDO | TDI | TMS | $V_{DD}$_ CORE | | | | | | | | | E |
| F | ALT_ INTA/ | INTB/ | $V_{SS}$_ CORE | INTA/ | | | | | | | | | F |
| G | RST/ | INT_DIR | ALT_ INTB/ | $V_{DD}$_IO | | | | | | | | | G |
| H | AD31 | REQ/ | CLK | GNT/ | | | | | | | | | H |
| J | AD27 | AD28 | AD30 | AD29 | | | | | | | | | J |
| K | C_BE3/ | AD24 | AD26 | $V_{DD}$_IO | | | | | | $V_{SS}$_IO | $V_{SS}$_IO | $V_{SS}$_IO | K |
| L | AD23 | AD22 | IDSEL | AD25 | | | | | | $V_{SS}$_IO | $V_{SS}$_IO | $V_{SS}$_IO | L |
| M | AD21 | AD19 | AD20 | $V_{SS}$_IO | | | | | | $V_{SS}$_IO | $V_{SS}$_IO | $V_{SS}$_IO | M |
| N | AD17 | AD18 | AD16 | IRDY/ | | | | | | $V_{SS}$_IO | $V_{SS}$_IO | $V_{SS}$_IO | N |
| P | C_BE2/ | FRAME/ | TRDY/ | $V_{DD}$_IO | | | | | | $V_{SS}$_IO | $V_{SS}$_IO | $V_{SS}$_IO | P |
| R | DEVSEL/ | STOP/ | SERR/ | PERR/ | | | | | | | | | R |
| T | PAR | C_BE1/ | AD14 | AD15 | | | | | | | | | T |
| U | AD13 | AD12 | AD11 | $V_{DD}$_IO | | | | | | | | | U |
| V | AD10 | AD9 | C_BE0/ | AD8 | | | | | | | | | V |
| W | AD7 | AD6 | AD4 | AD5 | | | | | | | | | W |
| Y | AD3 | AD2 | AD0 | $V_{SS}$_IO | AD63 | AD59 | $V_{DD}$_IO | AD52 | AD48 | $V_{DD}$_IO | AD44 | $V_{SS}$_IO | Y |
| AA | AD1 | REQ64/ | $V_{SS}$_IO | C_BE7/ | PAR64 | AD60 | AD56 | AD53 | AD49 | AD45 | AD41 | AD37 | AA |
| AB | ACK64/ | $V_{SS}$_ CORE | $V_{DD}$_ CORE | C_BE5/ | AD62 | AD58 | AD55 | AD51 | AD47 | AD43 | AD39 | AD36 | AB |
| AC | $V_{DD}$_ CORE | $V_{SS}$_ CORE | C_BE6/ | C_BE4/ | AD61 | AD57 | AD54 | AD50 | AD46 | AD42 | AD40 | AD38 | AC |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | |

## Table 3.2 Right Half of the SYM53C1010-33 329 BGA Chip - Top View

| | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | NC | A_SACK+ | A_SMSG+ | A_SC_D+ | A_SREQ+ | A_SD8– | A_SD10– | A_DIFFSENS | SCLK | $V_{SS}$_CORE | NC | A |
| B | NC | A_SRST– | A_SSEL– | NC | A_SI_O– | A_SD8+ | A_SD10+ | $V_{SS}$_A | NC | $V_{DD}$_CORE | $V_{DD}$_CORE | B |
| C | A_SACK– | A_SMSG– | A_SC_D– | A_SREQ– | A_SI_O+ | A_SD9+ | A_SD11+ | $V_{DD}$_A | $V_{SS}$_IO | SCAN_MODE | TEST_HSC | C |
| D | A_SRST+ | $V_{DD}$_IO | A_SSEL+ | NC | $V_{DD}$_IO | A_SD9– | A_SD11– | $V_{SS}$_IO | $V_{SS}$_CORE | B_SD12– | B_SD12+ | D |
| E | | | | | | | | B_SD13+ | B_SD13– | B_SD14– | B_SD14+ | E |
| F | | | | | | | | B_SD15+ | B_SD15– | B_SDP1– | B_SDP1+ | F |
| G | | | | | | | | $V_{DD}$_IO | B_SD0– | B_SD0+ | B_SD1– | G |
| H | | | | | | | | B_SD2– | B_SD1+ | B_SD2+ | B_SD3– | H |
| J | | | | | | | | B_SD4– | B_SD3+ | B_SD4+ | B_SD5– | J |
| K | $V_{SS}$_IO | $V_{SS}$_IO | | | | | | $V_{DD}$_IO | B_SD5+ | B_SD6+ | B_SD7– | K |
| L | $V_{SS}$_IO | $V_{SS}$_IO | | | | | | B_SD6– | B_SD7+ | B_SDP0+ | B_SDP0– | L |
| M | $V_{SS}$_IO | $V_{SS}$_IO | | | | | | $V_{SS}$_IO | RBIAS | $V_{DD}$_BIAS | B_SATN– | M |
| N | $V_{SS}$_IO | $V_{SS}$_IO | | | | | | B_SACK– | B_SBSY+ | B_SATN+ | B_SBSY– | N |
| P | $V_{SS}$_IO | $V_{SS}$_IO | | | | | | $V_{DD}$_IO | B_SACK+ | NC | NC | P |
| R | | | | | | | | B_SMSG– | B_SMSG+ | B_SRST+ | B_SRST– | R |
| T | | | | | | | | B_SC_D– | B_SC_D+ | B_SSEL+ | B_SSEL– | T |
| U | | | | | | | | $V_{DD}$_IO | B_SREQ– | NC | NC | U |
| V | | | | | | | | B_SI_O+ | B_SD8– | B_SI_O– | B_SREQ+ | V |
| W | | | | | | | | B_SD9+ | B_SD10– | B_SD9– | B_SD8+ | W |
| Y | $V_{DD}$_CORE | $V_{DD}$_IO | $V_{SS}$_CORE | A_GPIO1_MASTER/ | $V_{DD}$_IO | MOE/_TESTOUT | MAD7 | $V_{SS}$_IO | B_DIFFSENS | B_SD11– | B_SD10+ | Y |
| AA | AD33 | B_GPIO0_FETCH/ | B_GPIO3 | A_GPIO2 | MAS1/ | MCE/ | MAD6 | MAD3 | $V_{SS}$_IO | $V_{SS}$_CORE | B_SD11+ | AA |
| AB | AD35 | RESERVED | B_GPIO2 | A_GPIO0_FETCH/ | A_GPIO4 | $V_{DD}$_CORE | $V_{SS}$_CORE | MAD4 | MAD1 | NC | $V_{DD}$_CORE | AB |
| AC | AD34 | AD32 | B_GPIO1_MASTER/ | B_GPIO4 | A_GPIO3 | MAS0/ | MWE/ | MAD5 | $V_{SS}$_CORE | MAD2 | MAD0 | AC |
| | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | |

## Table 3.3 Signal Names and BGA Position

| Signal Name | BGA Pos | Signal Name | BGA Pos | Signal Name | BGA Pos | Signal Name | BGA Pos | Signal Name | BGA Pos |
|---|---|---|---|---|---|---|---|---|---|
| A_DIFFSENS | A20 | AD4 | W3 | B_GPIO2 | AB15 | FRAME/ | P2 | $V_{DD}$_IO | Y14 |
| A_GPIO0_FETCH/ | AB16 | AD5 | W4 | B_GPIO3 | AA15 | GNT/ | H4 | $V_{DD}$_IO | Y17 |
| A_GPIO1_MASTER/ | Y16 | AD6 | W2 | B_GPIO4 | AC16 | IDSEL | L3 | $V_{DD}$_A | B2 |
| A_GPIO2 | AA16 | AD7 | W1 | B_SACK– | N20 | INT_DIR | G2 | $V_{DD}$_A | C20 |
| A_GPIO3 | AC17 | AD8 | V4 | B_SACK+ | P21 | INTA/ | F4 | $V_{DD}$_BIAS | M22 |
| A_GPIO4 | AB17 | AD9 | V2 | B_SATN– | M23 | INTB/ | F2 | $V_{DD}$_BIAS2 | A11 |
| A_SACK– | C13 | AD10 | V1 | B_SATN+ | N22 | IRDY/ | N4 | $V_{DD}$_CORE | B22 |
| A_SACK+ | A14 | AD11 | U3 | B_SBSY– | N23 | MAD0 | AC23 | $V_{DD}$_CORE | B23 |
| A_SATN– | B11 | AD12 | U2 | B_SBSY+ | N21 | MAD1 | AB21 | $V_{DD}$_CORE | D3 |
| A_SATN+ | B12 | AD13 | U1 | B_SC_D– | T20 | MAD2 | AC22 | $V_{DD}$_CORE | E4 |
| A_SBSY– | C12 | AD14 | T3 | B_SC_D+ | T21 | MAD3 | AA20 | $V_{DD}$_CORE | Y13 |
| A_SBSY+ | A12 | AD15 | T4 | B_SD0– | G21 | MAD4 | AB20 | $V_{DD}$_CORE | AB3 |
| A_SC_D– | C15 | AD16 | N3 | B_SD0+ | G22 | MAD5 | AC20 | $V_{DD}$_CORE | AB18 |
| A_SC_D+ | A16 | AD17 | N1 | B_SD1– | G23 | MAD6 | AA19 | $V_{DD}$_CORE | AB23 |
| A_SD0– | B6 | AD18 | N2 | B_SD1+ | H21 | MAD7 | Y19 | $V_{DD}$_CORE | AC1 |
| A_SD0+ | A6 | AD19 | M2 | B_SD2– | H20 | MAS0/ | AC18 | $V_{SS}$_IO | C3 |
| A_SD1– | C7 | AD20 | M3 | B_SD2+ | H22 | MAS1/ | AA17 | $V_{SS}$_IO | C21 |
| A_SD1+ | B7 | AD21 | M1 | B_SD3– | H23 | MCE/ | AA18 | $V_{SS}$_IO | D4 |
| A_SD2– | A7 | AD22 | L2 | B_SD3+ | J21 | MOE/_TESTOUT | Y18 | $V_{SS}$_IO | D12 |
| A_SD2+ | C8 | AD23 | L1 | B_SD4– | J20 | MWE/ | AC19 | $V_{SS}$_IO | D20 |
| A_SD3– | D8 | AD24 | K2 | B_SD4+ | J22 | NC | A1 | $V_{SS}$_IO | K10 |
| A_SD3+ | B8 | AD25 | L4 | B_SD5– | J23 | NC | A13 | $V_{SS}$_IO | K11 |
| A_SD4– | A8 | AD26 | K3 | B_SD5+ | K21 | NC | A23 | $V_{SS}$_IO | K12 |
| A_SD4+ | C9 | AD27 | J1 | B_SD6– | L20 | NC | B1 | $V_{SS}$_IO | K13 |
| A_SD5– | D9 | AD28 | J2 | B_SD6+ | K22 | NC | B3 | $V_{SS}$_IO | K14 |
| A_SD5+ | B9 | AD29 | J4 | B_SD7– | K23 | NC | B13 | $V_{SS}$_IO | L10 |
| A_SD6– | A9 | AD30 | J3 | B_SD7+ | L21 | NC | B16 | $V_{SS}$_IO | L11 |
| A_SD6+ | C10 | AD31 | H1 | B_SD8– | V21 | NC | B21 | $V_{SS}$_IO | L12 |
| A_SD7– | D11 | AD32 | AC14 | B_SD8+ | W23 | NC | D16 | $V_{SS}$_IO | L13 |
| A_SD7+ | B10 | AD33 | AA13 | B_SD9– | W22 | NC | U22 | $V_{SS}$_IO | L14 |
| A_SD8– | A18 | AD34 | AC13 | B_SD9+ | W20 | NC | U23 | $V_{SS}$_IO | M4 |
| A_SD8+ | B18 | AD35 | AB13 | B_SD10– | W21 | NC | P22 | $V_{SS}$_IO | M10 |
| A_SD9– | D18 | AD36 | AB12 | B_SD10+ | Y23 | NC | P23 | $V_{SS}$_IO | M11 |
| A_SD9+ | C18 | AD37 | AA12 | B_SD11– | Y22 | NC | AB22 | $V_{SS}$_IO | M12 |
| A_SD10– | A19 | AD38 | AC12 | B_SD11+ | AA23 | PAR | T1 | $V_{SS}$_IO | M13 |
| A_SD10+ | B19 | AD39 | AB11 | B_SD12– | D22 | PAR64 | AA5 | $V_{SS}$_IO | M14 |
| A_SD11– | D19 | AD40 | AC11 | B_SD12+ | D23 | PERR/ | R4 | $V_{SS}$_IO | M20 |
| A_SD11+ | C19 | AD41 | AA11 | B_SD13– | E21 | RBIAS | M21 | $V_{SS}$_IO | N10 |
| A_SD12– | C4 | AD42 | AC10 | B_SD13+ | E20 | REQ/ | H2 | $V_{SS}$_IO | N11 |
| A_SD12+ | A3 | AD43 | AB10 | B_SD14– | E22 | REQ64/ | AA2 | $V_{SS}$_IO | N12 |
| A_SD13– | B4 | AD44 | Y11 | B_SD14+ | E23 | RESERVED | AB14 | $V_{SS}$_IO | N13 |
| A_SD13+ | A4 | AD45 | AA10 | B_SD15– | F21 | RST/ | G1 | $V_{SS}$_IO | N14 |
| A_SD14– | C5 | AD46 | AC9 | B_SD15+ | F20 | SCAN_MODE | C22 | $V_{SS}$_IO | P10 |
| A_SD14+ | D5 | AD47 | AB9 | B_SDP0– | L23 | SCLK | A21 | $V_{SS}$_IO | P11 |
| A_SD15– | B5 | AD48 | Y9 | B_SDP0+ | L22 | SERR/ | R3 | $V_{SS}$_IO | P12 |
| A_SD15+ | A5 | AD49 | AA9 | B_SDP1– | F22 | STOP/ | R2 | $V_{SS}$_IO | P13 |
| A_SDP0– | A10 | AD50 | AC8 | B_SDP1+ | F23 | TCK | D1 | $V_{SS}$_IO | P14 |
| A_SDP0+ | C11 | AD51 | AB8 | B_SI_O– | V22 | TDI | E2 | $V_{SS}$_IO | Y4 |
| A_SDP1– | C6 | AD52 | Y8 | B_SI_O+ | V20 | TDO | E1 | $V_{SS}$_IO | Y12 |
| A_SDP1+ | D6 | AD53 | AA8 | B_SMSG– | R20 | TEST_HSC | C23 | $V_{SS}$_IO | Y20 |
| A_SI_O– | B17 | AD54 | AC7 | B_SMSG+ | R21 | TMS | E3 | $V_{SS}$_IO | AA3 |
| A_SI_O+ | C17 | AD55 | AB7 | B_SREQ– | U21 | TRDY/ | P3 | $V_{SS}$_IO | AA21 |
| A_SMSG– | C14 | AD56 | AA7 | B_SREQ+ | V23 | TEST_PD | A2 | $V_{SS}$_A | B20 |
| A_SMSG+ | A15 | AD57 | AC6 | B_SRST– | R23 | TEST_RST/ | C1 | $V_{SS}$_A | C2 |
| A_SREQ– | C16 | AD58 | AB6 | B_SRST+ | R22 | $V_{DD}$_IO | D7 | $V_{SS}$_CORE | A22 |
| A_SREQ+ | A17 | AD59 | Y6 | B_SSEL– | T23 | $V_{DD}$_IO | D10 | $V_{SS}$_CORE | D2 |
| A_SRST– | B14 | AD60 | AA6 | B_SSEL+ | T22 | $V_{DD}$_IO | D14 | $V_{SS}$_CORE | D21 |
| A_SRST+ | D13 | AD61 | AC5 | C_BE0/ | V3 | $V_{DD}$_IO | D17 | $V_{SS}$_CORE | F3 |
| A_SSEL– | B15 | AD62 | AB5 | C_BE1/ | T2 | $V_{DD}$_IO | G4 | $V_{SS}$_CORE | Y15 |
| A_SSEL+ | D15 | AD63 | Y5 | C_BE2/ | P1 | $V_{DD}$_IO | K4 | $V_{SS}$_CORE | AB2 |
| ACK64/ | AB1 | ALT_INTA/ | F1 | C_BE3/ | K1 | $V_{DD}$_IO | K20 | $V_{SS}$_CORE | AA22 |
| AD0 | Y3 | ALT_INTB/ | G3 | C_BE4/ | AC4 | $V_{DD}$_IO | P4 | $V_{SS}$_CORE | AB19 |
| AD1 | AA1 | B_DIFFSENS | Y21 | C_BE5/ | AB4 | $V_{DD}$_IO | P20 | $V_{SS}$_CORE | AC2 |
| AD2 | Y2 | B_GPIO0_FETCH/ | AA14 | C_BE6/ | AC3 | $V_{DD}$_IO | U4 | $V_{SS}$_CORE | AC21 |
| AD3 | Y1 | B_GPIO1_MASTER/ | AC15 | C_BE7/ | AA4 | $V_{DD}$_IO | U20 | | |
| | | | | CLK | H3 | $V_{DD}$_IO | Y7 | | |
| | | | | DEVSEL/ | R1 | $V_{DD}$_IO | Y10 | | |

## Table 3.4 Signal Names by BGA Position

| Signal Name | BGA Pos | Signal Name | BGA Pos | Signal Name | BGA Pos | Signal Name | BGA Pos | Signal Name | BGA Pos |
|---|---|---|---|---|---|---|---|---|---|
| NC | A1 | MAD4 | AB20 | A_SI_O+ | C17 | B_SD3+ | J21 | NC | P23 |
| TEST_PD | A2 | MAD1 | AB21 | A_SD9+ | C18 | B_SD4+ | J22 | DEVSEL/ | R1 |
| A_SD12+ | A3 | NC | AB22 | A_SD11+ | C19 | B_SD5– | J23 | STOP/ | R2 |
| A_SD13+ | A4 | $V_{DD}$_CORE | AB23 | $V_{DD}$_A | C20 | C_BE3/ | K1 | SERR/ | R3 |
| A_SD15+ | A5 | $V_{DD}$_CORE | AC1 | $V_{SS}$_IO | C21 | AD24 | K2 | PERR/ | R4 |
| A_SD0+ | A6 | $V_{SS}$_CORE | AC2 | SCAN_MODE | C22 | AD26 | K3 | B_SMSG– | R20 |
| A_SD2– | A7 | C_BE6/ | AC3 | TEST_HSC | C23 | $V_{DD}$_IO | K4 | B_SMSG+ | R21 |
| A_SD4– | A8 | C_BE4/ | AC4 | TCK | D1 | $V_{SS}$_IO | K10 | B_SRST+ | R22 |
| A_SD6– | A9 | AD61 | AC5 | $V_{SS}$_CORE | D2 | $V_{SS}$_IO | K11 | B_SRST– | R23 |
| A_SDP0– | A10 | AD57 | AC6 | $V_{DD}$_CORE | D3 | $V_{SS}$_IO | K12 | PAR | T1 |
| $V_{DD}$-BIAS2 | A11 | AD54 | AC7 | $V_{SS}$_IO | D4 | $V_{SS}$_IO | K13 | C_BE1/ | T2 |
| A_SBSY+ | A12 | AD50 | AC8 | A_SD14+ | D5 | $V_{SS}$_IO | K14 | AD14 | T3 |
| NC | A13 | AD46 | AC9 | A_SDP1+ | D6 | $V_{DD}$_IO | K20 | AD15 | T4 |
| A_SACK+ | A14 | AD42 | AC10 | $V_{DD}$_IO | D7 | B_SD5+ | K21 | B_SC_D– | T20 |
| A_SMSG+ | A15 | AD40 | AC11 | A_SD3– | D8 | B_SD6+ | K22 | B_SC_D+ | T21 |
| A_SC_D+ | A16 | AD38 | AC12 | A_SD5– | D9 | B_SD7– | K23 | B_SSEL+ | T22 |
| A_SREQ+ | A17 | AD34 | AC13 | $V_{DD}$_IO | D10 | AD23 | L1 | B_SSEL– | T23 |
| A_SD8– | A18 | AD32 | AC14 | A_SD7– | D11 | AD22 | L2 | AD13 | U1 |
| A_SD10– | A19 | B_GPIO1_ MASTER/ | AC15 | $V_{SS}$_IO | D12 | IDSEL | L3 | AD12 | U2 |
| A_DIFFSENS | A20 | B_GPIO4 | AC16 | A_SRST+ | D13 | AD25 | L4 | AD11 | U3 |
| SCLK | A21 | A_GPIO3 | AC17 | $V_{DD}$_IO | D14 | $V_{SS}$_IO | L10 | $V_{DD}$_IO | U4 |
| $V_{SS}$_CORE | A22 | MAS0/ | AC18 | A_SSEL+ | D15 | $V_{SS}$_IO | L11 | $V_{DD}$_IO | U20 |
| NC | A23 | MWE/ | AC19 | NC | D16 | $V_{SS}$_IO | L12 | B_SREQ– | U21 |
| AD1 | AA1 | MAD5 | AC20 | $V_{DD}$_IO | D17 | $V_{SS}$_IO | L13 | NC | U22 |
| REQ64/ | AA2 | $V_{SS}$_CORE | AC21 | A_SD9– | D18 | $V_{SS}$_IO | L14 | NC | U23 |
| $V_{SS}$_IO | AA3 | MAD2 | AC22 | A_SD11– | D19 | B_SD6– | L20 | AD10 | V1 |
| C_BE7/ | AA4 | MAD0 | AC23 | $V_{SS}$_IO | D20 | B_SD7+ | L21 | AD9 | V2 |
| PAR64 | AA5 | NC | B1 | $V_{SS}$_CORE | D21 | B_SDP0+ | L22 | C_BE0/ | V3 |
| AD60 | AA6 | $V_{DD}$_A | B2 | B_SD12– | D22 | B_SDP0– | L23 | AD8 | V4 |
| AD56 | AA7 | NC | B3 | B_SD12+ | D23 | AD21 | M1 | B_SI_O+ | V20 |
| AD53 | AA8 | A_SD13– | B4 | TDO | E1 | AD19 | M2 | B_SD8– | V21 |
| AD49 | AA9 | A_SD15– | B5 | TDI | E2 | AD20 | M3 | B_SI_O– | V22 |
| AD45 | AA10 | A_SD0– | B6 | TMS | E3 | $V_{SS}$_IO | M4 | B_SREQ+ | V23 |
| AAD41 | AA11 | A_SD1+ | B7 | $V_{DD}$_CORE | E4 | $V_{SS}$_IO | M10 | AD7 | W1 |
| AD37 | AA12 | A_SD3+ | B8 | B_SD13+ | E20 | $V_{SS}$_IO | M11 | AD6 | W2 |
| AD33 | AA13 | A_SD5+ | B9 | B_SD13– | E21 | $V_{SS}$_IO | M12 | AD4 | W3 |
| B_GPIO0_ FETCH/ | AA14 | A_SD7+ | B10 | B_SD14– | E22 | $V_{SS}$_IO | M13 | AD5 | W4 |
| B_GPIO3 | AA15 | A_SATN– | B11 | B_SD14+ | E23 | $V_{SS}$_IO | M14 | B_SD9+ | W20 |
| A_GPIO2 | AA16 | A_SATN+ | B12 | ALT_INTA/ | F1 | $V_{SS}$_IO | M20 | B_SD10– | W21 |
| MAS1/ | AA17 | NC | B13 | INTB/ | F2 | RBIAS | M21 | B_SD9– | W22 |
| MCE/ | AA18 | A_SRST– | B14 | $V_{SS}$_CORE | F3 | $V_{DD}$-BIAS | M22 | B_SD8+ | W23 |
| MAD6 | AA19 | A_SSEL– | B15 | INTA/ | F4 | B_SATN– | M23 | AD3 | Y1 |
| MAD3 | AA20 | NC | B16 | B_SD15+ | F20 | AD17 | N1 | AD2 | Y2 |
| $V_{SS}$_IO | AA21 | A_SI_O– | B17 | B_SD15– | F21 | AD18 | N2 | AD0 | Y3 |
| $V_{SS}$_CORE | AA22 | A_SD8+ | B18 | B_SDP1– | F22 | AD16 | N3 | $V_{SS}$_IO | Y4 |
| B_SD11+ | AA23 | A_SD10+ | B19 | B_SDP1+ | F23 | IRDY/ | N4 | AD63 | Y5 |
| ACK64/ | AB1 | $V_{SS}$_A | B20 | RST/ | G1 | $V_{SS}$_IO | N10 | AD59 | Y6 |
| $V_{SS}$_CORE | AB2 | NC | B21 | INT_DIR | G2 | $V_{SS}$_IO | N11 | $V_{DD}$_IO | Y7 |
| $V_{DD}$_CORE | AB3 | $V_{DD}$_CORE | B22 | ALT_INTB/ | G3 | $V_{SS}$_IO | N12 | AD52 | Y8 |
| C_BE5/ | AB4 | $V_{DD}$_CORE | B23 | $V_{DD}$_IO | G4 | $V_{SS}$_IO | N13 | AD48 | Y9 |
| AD62 | AB5 | TEST_RST/ | C1 | $V_{DD}$_IO | G20 | $V_{SS}$_IO | N14 | $V_{DD}$_IO | Y10 |
| AD58 | AB6 | $V_{SS}$_A | C2 | B_SD0– | G21 | B_SACK– | N20 | AD44 | Y11 |
| AD55 | AB7 | $V_{SS}$_IO | C3 | B_SD0+ | G22 | B_SBSY+ | N21 | $V_{SS}$_IO | Y12 |
| AD51 | AB8 | A_SD12– | C4 | B_SD1– | G23 | B_SATN+ | N22 | $V_{DD}$_CORE | Y13 |
| AD47 | AB9 | A_SD14– | C5 | AD31 | H1 | B_SBSY– | N23 | $V_{DD}$_IO | Y14 |
| AD43 | AB10 | A_SDP1– | C6 | REQ/ | H2 | C_BE2/ | P1 | $V_{SS}$_CORE | Y15 |
| AD39 | AB11 | A_SD1– | C7 | CLK | H3 | FRAME/ | P2 | A_GPIO1_ MASTER/ | Y16 |
| AD36 | AB12 | A_SD2+ | C8 | GNT/ | H4 | TRDY/ | P3 | $V_{DD}$_IO | Y17 |
| AD35 | AB13 | A_SD4+ | C9 | B_SD2– | H20 | $V_{DD}$_IO | P4 | MOE/ TESTOUT | Y18 |
| RESERVED | AB14 | A_SD6+ | C10 | B_SD1+ | H21 | $V_{SS}$_IO | P10 | MAD7 | Y19 |
| B_GPIO2 | AB15 | A_SDP0+ | C11 | B_SD2+ | H22 | $V_{SS}$_IO | P11 | $V_{SS}$_IO | Y20 |
| A_GPIO0_ FETCH/ | AB16 | A_SBSY– | C12 | B_SD3– | H23 | $V_{SS}$_IO | P12 | B_DIFFSENS | Y21 |
| A_GPIO4 | AB17 | A_SACK– | C13 | AD27 | J1 | $V_{SS}$_IO | P13 | B_SD11– | Y22 |
| $V_{DD}$_CORE | AB18 | A_SMSG– | C14 | AD28 | J2 | $V_{SS}$_IO | P14 | B_SD10+ | Y23 |
| $V_{SS}$_CORE | AB19 | A_SC_D– | C15 | AD30 | J3 | $V_{DD}$_IO | P20 | | |
| | | A_SREQ– | C16 | AD29 | J4 | B_SACK+ | P21 | | |
| | | | | B_SD4– | J20 | NC | P22 | | |

## 3.2 Internal Pull-ups and Pull-downs on SYM53C1010 Signals

Several SYM53C1010 signals use internal pull-ups and pull-downs. Table 3.5 describes the conditions that enable these pull-ups and pull-downs.

**Table 3.5 SYM53C1010 Internal Pull-ups and Pull-downs**

| Pin Name | Pull-up Current | Conditions for Pull-up |
|---|---|---|
| INTA/, INTB/, ALT_INTA/, ALT_INTB/ | 25 µA | Pull-up enabled when the “AND-tree” mode is enabled by driving TEST_RST/ low or when the IRQ mode bit (bit 3 of DCNTL, 0x3B) is cleared.[1] |
| INT_DIR, TCK, TDI, TEST_RST/, TMS | 25 µA | Pulled up internally. |
| AD[63:32], C_BE[7:4]/, PAR64 | 25 µA | Pulled down internally. |
| GPIO[4:0] | –25 µA | Pulled up internally. |
| MAD[7:0] | –25 µA | Pulled down internally. |
| TEST_HSC, TEST_PD, SCAN_MODE | –25 µA | Pulled down internally. |

1. When bit 3 of the DMA Control (DCNTL) register is set, the pad becomes a totem pole output pad and drives both high and low.

## 3.3 PCI Bus Interface Signals

The PCI Bus Interface Signals section contains tables describing the signals for the following signal groups: System Signals, Address and Data Signals, Interface Control Signals, Arbitration Signals, Error Reporting Signals, Interrupt Signals, SCSI Function A GPIO Signals, and SCSI Function B GPIO Signals.

### 3.3.1 System Signals

Table 3.6 describes the System Signals group.

**Table 3.6 System Signals**

| Name | Bump | Type | Strength | Description |
|---|---|---|---|---|
| CLK | H3 | I | N/A | **Clock** provides timing for all transactions on the PCI bus and is an input to every PCI device. All other PCI signals are sampled on the rising edge of CLK. Other timing parameters are defined with respect to this edge. |
| RST/ | G1 | I | N/A | **Reset** forces the PCI sequencer of each device to a known state. All T/S and S/T/S signals are forced to a high impedance state, and all internal logic is reset. The RST/ input is synchronized internally to the rising edge of CLK. To properly reset the device, the CLK input must be active while RST/ is active. |

### 3.3.2 Address and Data Signals

Table 3.7 describes the Address and Data Signals group.

**Table 3.7 Address and Data Signals**

| Name | Bump | Type | Strength | Description |
|---|---|---|---|---|
| AD[63:0] | Y5, AB5, AC5, AA6, Y6, AB6, AC6, AA7, AB7, AC7, AA8, Y8, AB8, AC8, AA9, Y9, AB9, AC9, AA10, Y11, AB10, AC10, AA11, AC11, AB11, AC12, AA12, AB12, AB13, AC13, AA13, AC14, H1, J3, J4, J2, J1, K3, L4, K2, L1, L2, M1, M3, M2, N2, N1, N3, T4, T3, U1–U3, V1, V2, V4, W1, W2, W4, W3, Y1, Y2, AA1, Y3. | T/S | 8 mA PCI | Physical longword **Address and Data** are multiplexed on the same PCI pins. A bus transaction consists of an address phase followed by one or more data phases. During the first clock of a transaction, AD[63:0] contain a 64-bit physical byte address. If the command is a dual address cycle (DAC), implying a 64-bit address, AD[31:0] will contain the upper 32 bits of the address during the second clock of the transaction. During subsequent clocks, AD[63:0] will contain data. PCI supports both read and write bursts. AD[7:0] define the least significant byte, and AD[63:56] define the most significant byte. |
| C_BE[7:0]/ | AA4, AC3, AB4, AC4, K1, P1, T2, V3. | T/S | 8 mA PCI | Bus **Command and Byte Enables** are multiplexed on the same PCI pins. During the address phase of a transaction, C_BE[3:0]/ define the bus command. If the transaction is a DAC, C_BE[3:0]/ contain the DAC command and C_BE[7:4]/ define the bus command. C_BE[3:0]/ define the bus command during the second clock of the transaction. During the data phase, C_BE[7:0]/ are used as byte enables. The byte enables determine which byte lanes carry meaningful data: C_BE[0]/ applies to byte 0, and C_BE[7]/ to byte 7. |
| PAR | T1 | T/S | 8 mA PCI | **Parity** is the even parity bit that protects the AD[31:0] and C_BE[3:0]/ lines. During the address phase, both the address and command bits are covered. During the data phase, both the data and byte enables are covered. |

**Table 3.7 Address and Data Signals (Cont.)**

| Name | Bump | Type | Strength | Description |
|---|---|---|---|---|
| PAR64 | AA5 | T/S | 8 mA PCI | **Parity64** is the even parity bit that protects the AD[63:32] and C_BE[7:4]/ lines. During the address phase, the address and command bits are covered. During the data phase, both the data and byte enables are covered. |

### 3.3.3 Interface Control Signals

Table 3.8 describes the Interface Control Signals group.

**Table 3.8 Interface Control Signals**

| Name | Bump | Type | Strength | Description |
|---|---|---|---|---|
| ACK64/ | AB1 | S/T/S | 8 mA PCI | **Acknowledge 64**-bit transfer is driven by the current bus target to indicate its ability to transfer 64-bit data. |
| REQ64/ | AA2 | S/T/S | 8 mA PCI | **Request 64**-bit transfer is driven by the current bus master to indicate a request to transfer 64-bit data. |
| FRAME/ | P2 | S/T/S | 8 mA PCI | **Cycle Frame** is driven by the current master to indicate the beginning and duration of an access. FRAME/ is asserted to indicate that a bus transaction is beginning. While FRAME/ is deasserted, either the transaction is in the final data phase or the bus is idle. |
| TRDY/ | P3 | S/T/S | 8 mA PCI | **Target Ready** indicates the target's ability to complete the current data phase of the transaction. TRDY/ is used with IRDY/. A data phase is completed on any clock when both TRDY/ and IRDY/ are sampled asserted. During a read, TRDY/ indicates that valid data is present on the AD bus. During a write, it indicates that the target is prepared to accept data. |
| IRDY/ | N4 | S/T/S | 8 mA PCI | **Initiator Ready** indicates the initiator's ability to complete the current data phase of the transaction. IRDY/ is used with TRDY/. A data phase is completed on any clock when both IRDY/ and TRDY/ are sampled asserted. During a write, IRDY/ indicates that valid data is present on the AD bus. During a read, it indicates that the master is prepared to accept data. |

**Table 3.8 Interface Control Signals (Cont.)**

| Name | Bump | Type | Strength | Description |
|---|---|---|---|---|
| STOP/ | R2 | S/T/S | 8 mA PCI | **Stop** indicates that the selected target is requesting the master to stop the current transaction. |
| DEVSEL/ | R1 | S/T/S | 8 mA PCI | **Device Select** indicates that the driving device has decoded its address as the target of the current access. As an input, it indicates to a master whether any device on the bus has been selected. |
| IDSEL | L3 | I | N/A | **Initialization Device Select** is used as a chip select, in place of the upper 24 address lines, during configuration read and write transactions. |

## 3.3.4 Arbitration Signals

Table 3.9 describes the Arbitration Signals group.

**Table 3.9 Arbitration Signals**

| Name | Bump | Type | Strength | Description |
|---|---|---|---|---|
| REQ/ | H2 | O | 8 mA PCI | **Request** indicates to the system arbiter that this agent requests use of the PCI bus. This is a point-to-point signal. Every master has its own REQ/ signal. |
| GNT/ | H4 | I | N/A | **Grant** indicates to a specific agent that access to the PCI bus has been granted. This is a point-to-point signal. Every master has its own GNT/ signal. |

### 3.3.5 Error Reporting Signals

Table 3.10 describes the Error Reporting Signals group.

**Table 3.10 Error Reporting Signals**

| Name | Bump | Type | Strength | Description |
|---|---|---|---|---|
| PERR/ | R4 | S/T/S | 8 mA PCI | **Parity Error** may be pulsed active by an agent that detects a data parity error. PERR/ can be used by any agent to signal data corruption. On detection of a PERR/ pulse, a nonmaskable interrupt is generated to the host CPU, which often implies the system is unable to continue operation once error processing is complete. |
| SERR/ | R3 | O | 8 mA PCI | **System Error** is an open drain output used to report address parity errors as well as critical errors other than parity. |

### 3.3.6 Interrupt Signals

Table 3.11 describes the Interrupt Signals group.

**Table 3.11 Interrupt Signals**

| Name[1] | Bump | Type | Strength | Description |
|---|---|---|---|---|
| INTA/ | F4 | O | 8 mA PCI | **Interrupt Function A.** This signal, when asserted low, indicates that an interrupting condition in SCSI Function A requires service from the host CPU. The output drive of this pin is open drain. See SCSI Test One (STEST1) for additional information about disabling this interrupt in a RAID environment. This interrupt pin is disabled if INT_DIR is driven low.<br><br>If the SCSI Function B interrupt is rerouted at power-up using the INTA/ enable sense resistor (pull-up on MAD4), this signal indicates that an interrupting condition has occurred in either SCSI Function A or SCSI Function B. |

**Table 3.11 Interrupt Signals (Cont.)**

| Name[1] | Bump | Type | Strength | Description |
|---|---|---|---|---|
| INTB/ | F2 | O | 8 mA PCI | **Interrupt Function B.** This signal, when asserted low, indicates that an interrupting condition in SCSI Function B requires service from the host CPU. The output drive of this pin is open drain. See SCSI Test One (STEST1) for additional information about disabling this interrupt in a RAID environment. This interrupt pin is disabled if INT_DIR is driven low.<br><br>At power-up, this interrupt can be rerouted to INTA/ using the INTA/ enable sense resistor (pull-up on MAD4). This causes the SYM53C1010 to program the SCSI Function B PCI Interrupt Pin (IP) register to 0x01. |
| ALT_INTA/ | F1 | O | 8 mA PCI | **Alt Interrupt Function A.** This signal, when asserted low, indicates that an interrupting condition in SCSI Function A requires service. The output drive of this pin is open drain. See SCSI Test One (STEST1) for additional information about disabling this interrupt in a RAID environment.<br><br>If the SCSI Function B interrupt is rerouted at power-up using the INTA/ enable sense resistor (pull-up on MAD4), this signal indicates that an interrupting condition has occurred in either SCSI Function A or SCSI Function B. |
| ALT_INTB/ | G3 | O | 8 mA PCI | **Alt Interrupt Function B.** This signal, when asserted low, indicates that an interrupting condition in SCSI Function B requires service. The output drive of this pin is open drain. See SCSI Test One (STEST1) for additional information about disabling this interrupt in a RAID environment.<br><br>At power-up, this interrupt can be rerouted to INTA/ using the INTA/ enable sense resistor (pull-up on MAD4). This causes the SYM53C1010 to program the SCSI Function B PCI Interrupt Pin (IP) register to 0x01. |

**Table 3.11 Interrupt Signals (Cont.)**

| Name[1] | Bump | Type | Strength | Description |
|---|---|---|---|---|
| INT_DIR | G2 | I | N/A | **Interrupt Direction.** This input signal indicates whether internally generated interrupts are presented on INTA/ and INTB/. If INT_DIR is high, internal interrupts are generated on both the INTx/ pins and the ALT_INTx pin. If INT_DIR is low, the internal interrupts are generated only on the ALT_INTx/ pin. This pin has no effect if either bit 0 or 1 in STEST1 is set. This pin has a static pull-up. |

1. See register 0x4D, SCSI Test One (STEST1) in Chapter 4, "Registers" for additional information on these signals.

## 3.3.7 SCSI Function A GPIO Signals

Table 3.12 describes the SCSI Function A GPIO Signals group.

**Table 3.12 SCSI Function A GPIO Signals**

| Name | Bump | Type | Strength | Description |
|---|---|---|---|---|
| A_GPIO0_ FETCH/ | AB16 | I/O | 8 mA | **SCSI Function A General Purpose I/O pin 0.** This pin is programmable at power-up, through the MAD7 pin, to serve as the data signal for the serial EEPROM interface. When GPIO_0 is not in the process of downloading EEPROM data it can be used to drive a SCSI Activity LED, if bit 5 in the General Purpose Pin Control (GPCNTL) register is set. Or, it can be used to indicate that the next bus request will be an opcode fetch if bit 6 in the GPCNTL register is set. |
| A_GPIO1_ MASTER/ | Y16 | I/O | 8 mA | **SCSI Function A General Purpose I/O pin 1.** This pin is programmable at power-up, through the MAD7 pin, to serve as the clock signal for the serial EEPROM interface. If bit 7 of the General Purpose Pin Control (GPCNTL) register is set, this pin drives low when the SYM53C1010 is the bus master. |
| A_GPIO2 | AA16 | I/O | 8 mA | **SCSI Function A General Purpose I/O pin 2.** This pin powers up as an input. |
| A_GPIO3 | AC17 | I/O | 8 mA | **SCSI Function A General Purpose I/O pin 3.** This pin powers up as an input. |
| A_GPIO4 | AB17 | I/O | 8 mA | **SCSI Function A General Purpose I/O pin 4.** This pin powers up as an output. It can be used as the enable line for $V_{PP}$, the 12 V power supply to the external flash memory interface. |

### 3.3.8 SCSI Function B GPIO Signals

Table 3.13 describes the SCSI Function B GPIO Signals group.

**Table 3.13 SCSI Function B GPIO Signals**

| Name | Bump | Type | Strength | Description |
|---|---|---|---|---|
| B_GPIO0_ FETCH/ | AA14 | I/O | 8 mA | **SCSI Function B General Purpose I/O pin 0.** This pin is programmable at power-up through the MAD7 pin to serve as the data signal for the serial EEPROM interface. When GPIO_0 is not in the process of downloading EEPROM data, it can be used to drive a SCSI Activity LED if bit 5 in the General Purpose Pin Control (GPCNTL) register is set. Or, it can be used to indicate that the next bus request will be an opcode fetch if bit 6 in the GPCNTL register is set. |
| B_GPIO1_ MASTER/ | AC15 | I/O | 8 mA | **SCSI Function B General Purpose I/O pin 1.** This pin is programmable at power-up through the MAD7 pin to serve as the clock signal for the serial EEPROM interface. If bit 7 of the General Purpose Pin Control (GPCNTL) register is set, this pin is driven low when the SYM53C1010 is the bus master. |
| B_GPIO2 | AB15 | I/O | 8 mA | **SCSI Function B General Purpose I/O pin 2.** This pin powers up as an input. |
| B_GPIO3 | AA15 | I/O | 8 mA | **SCSI Function B General Purpose I/O pin 3.** This pin powers up as an input. |
| B_GPIO4 | AC16 | I/O | 8 mA | **SCSI Function B General Purpose I/O pin 4.** This pin powers up as an output. It can be used as the enable line for $V_{PP}$, the 12 V power supply to the external flash memory interface. |

## 3.4 SCSI Bus Interface Signals

The SCSI Bus Interface Signals section contains tables describing the signals for the following signal groups: SCSI Bus Interface Signals, SCSI Function A Signals, and SCSI Function B Signals. SCSI Function A signals and SCSI Function B signals each have a subgroup: the SCSI Function A Control Signals and the SCSI Function B Control Signals.

Table 3.14 contains signals that are common to both SCSI buses.

**Table 3.14 SCSI Bus Interface Signals**

| Name | Bump | Type | Strength | Description |
|---|---|---|---|---|
| SCLK | A21 | I | N/A | **SCSI Clock** is used to derive all SCSI related timings. The speed of this clock must be 40 MHz. The clock frequency can be quadrupled to create the 160 MHz clock required internally by both SCSI functions. |

### 3.4.1 SCSI Function A Signals

This section describes the signals for the SCSI Function A signals group. It is divided into two tables: Table 3.15 describes SCSI Function A signals and Table 3.16 describes SCSI Function A Control signals.

**Table 3.15 SCSI Function A Signals**

| Name | Bump | Type | Strength | Description |
|---|---|---|---|---|
| A_SD[15:0]– | B5, C5, B4, C4, D19, A19, D18, A18, D11, A9, D9, A8, D8, A7, C7, B6. | I/O | SE: 48 mA SCSI<br><br>LVD: 12 mA UniLVD | **SCSI Function A Data.**<br><br>LVD Mode: Negative half of the LVD Link pair for the SCSI data lines. A_SD[15:0]– form the 16-bit SCSI data bus.<br><br>SE Mode: A_SD[15:0]– form the 16-bit SCSI data bus. |
| A_SD[15:0]+ | A5, D5, A4, A3, C19, B19, C18, B18, B10, C10, B9, C9, B8, C8, B7, A6. | I/O | SE: 48 mA SCSI<br><br>LVD: 12 mA UniLVD | **SCSI Function A Data.**<br><br>LVD Mode: Positive half of the LVD Link pair for the SCSI data lines. A_SD[15:0]+ form the 16-bit data bus.<br><br>SE Mode: A_SD[15:0]+ are at 0 V. |

**Table 3.15 SCSI Function A Signals (Cont.)**

| Name | Bump | Type | Strength | Description |
|---|---|---|---|---|
| A_SDP[1:0]– | C6, A10. | I/O | SE: 48 mA SCSI<br><br>LVD: 12 mA UniLVD | **SCSI Function A Parity.**<br><br>LVD Mode: Negative half of the LVD Link pair for the SCSI parity lines. A_SDP[1:0]– are the SCSI data parity lines.<br><br>SE Mode: A_SDP[1:0]– are the SCSI data parity lines. |
| A_SDP[1:0]+ | D6, C11. | I/O | SE: 48 mA SCSI<br><br>LVD: 12 mA UniLVD | **SCSI Function A Parity.**<br><br>LVD Mode: Positive half of the LVD Link pair for the SCSI parity lines. A_SDP[1:0]+ are the SCSI data parity lines.<br><br>SE Mode: A_SDP[1:0]+ are at 0 V. |
| A_DIFFSENS | A20 | I | N/A | **SCSI Function A Differential Sense** pin detects the present mode of the SCSI bus when connected to the DIFFSENS signal on the physical SCSI bus.<br><br>LVD Mode: If a voltage between 0.7 V and 1.9 V is present, SCSI Function A operates in the LVD mode.<br><br>SE Mode: When this pin is driven low (below 0.5 V), SE operation is indicated. SCSI Function A operates in the SE mode.<br><br>HVD Mode: When this pin is detected high (above 2.0 V), HVD operation is indicated. SCSI Function A is driven to the high impedance state. This pin is 5 V tolerant. HVD Mode is not supported. |

**Table 3.16 SCSI Function A Control Signals**

| Name[1] | Bump | Type | Strength | Description |
|---|---|---|---|---|
| **SCSI Function A Control** includes the following signals: | | | | |
| A_SC_D–<br>A_SC_D+ | C15<br>A16 | I/O | SE: 48 mA SCSI<br><br>LVD: 12 mA UniLVD | SCSI phase line, command/data. |
| A_SI_O–<br>A_SI_O+ | B17<br>C17 | | | SCSI phase line, input/output. |
| A_SMSG–<br>A_SMSG+ | C14<br>A15 | | | SCSI phase line, message. |
| A_SREQ–<br>A_SREQ+ | C16<br>A17 | | | Data handshake line from target device. |
| A_SACK–<br>A_SACK+ | C13<br>A14 | | | Data handshake signal from the initiator device. |
| A_SBSY–<br>A_SBSY+ | C12<br>A12 | | | SCSI bus arbitration signal, busy. |
| A_SATN–<br>A_SATN+ | B11<br>B12 | | | SCSI Attention, the initiator is requesting a message out phase. |
| A_SRST–<br>A_SRST+ | B14<br>D13 | | | SCSI bus reset. |
| A_SSEL–<br>A_SSEL+ | B15<br>D15 | | | SCSI bus arbitration signal, select device. |

1. LVD Mode: Negative and positive halves of the LVD Link signal pairs are shown for SCSI Function A Control. SE Mode: The SCSI Function A Control signals are shown. All the positive (+) signals are at 0 Volts.

### 3.4.2 SCSI Function B Signals

This section describes the SCSI Function B Signals group. It is divided into two tables: Table 3.17 describes SCSI Function B Signals and Table 3.18 describes SCSI Function B Control Signals.

**Table 3.17 SCSI Function B Signals**

| Name | Bump | Type | Strength | Description |
|---|---|---|---|---|
| B_SD[15:0]– | F21, E22, E21, D22, Y22, W21, W22, V21, K23, L20, J23, J20, H23, H20, G23, G21. | I/O | SE: 48 mA SCSI<br><br>LVD: 12 mA UniLVD | **SCSI Function B Data.**<br><br>LVD Mode: Negative half of the LVD Link pair for the SCSI data lines. B_SD[15:0]– form the 16-bit SCSI data bus.<br><br>SE Mode: B_SD[15:0]– form the 16-bit SCSI data bus. |
| B_SD[15:0]+ | F20, E23, E20, D23, AA23, Y23, W20, W23, L21, K22, K21, J22, J21, H22, H21, G22. | I/O | SE: 48 mA SCSI<br><br>LVD: 12 mA UniLVD | **SCSI Function B Data.**<br><br>LVD Mode: Positive half of the LVD Link pair for the SCSI data lines. B_SD[15:0]+ form the 16-bit data bus.<br><br>SE Mode: B_SD[15:0]+ are at 0 V. |
| B_SDP[1:0]– | F22, L23. | I/O | SE: 48 mA SCSI<br><br>LVD: 12 mA UniLVD | **SCSI Function B Parity.**<br><br>LVD Mode: Negative half of the LVD Link pair for the SCSI parity lines. B_SDP[1:0]– are the SCSI data parity lines.<br><br>SE Mode: B_SDP[1:0]– are the SCSI data parity lines. |

**Table 3.17 SCSI Function B Signals (Cont.)**

| Name | Bump | Type | Strength | Description |
|---|---|---|---|---|
| B_SDP[1:0]+ | F23, L22. | I/O | SE: 48 mA SCSI<br><br>LVD: 12 mA UniLVD | **SCSI Function B Parity.**<br><br>LVD Mode: Positive half of the LVD Link pair for the SCSI parity lines. B_SDP[1:0]+ are the SCSI data parity lines.<br><br>SE Mode: B_SDP[1:0]+ are at 0 V. |
| B_DIFFSENS | Y21 | I | N/A | **SCSI Function B Differential Sense** pin detects the present mode of the SCSI bus when connected to the DIFFSENS signal on the physical SCSI bus.<br><br>LVD Mode: If a voltage between 0.7 V and 1.9 V is present, SCSI Function B operates in the LVD mode.<br><br>SE Mode: If a voltage below 0.5 V is present, SE operation is indicated. SCSI Function B operates in the SE mode.<br><br>HVD Mode: When this pin is detected high (above 2.0 V) HVD operation is indicated. SCSI Function B is driven to the high impedance state. This pin is 5 V tolerant. HVD mode is not supported. |

**Table 3.18 SCSI Function B Control Signals**

| Name[1] | Bump | Type | Strength | Description |
|---|---|---|---|---|
| **SCSI Function B Control** includes the following signals: | | | | |
| B_SC_D–<br>B_SD_D+ | T20<br>T21 | I/O | SE: 48 mA SCSI<br><br>LVD: 12 mA UniLVD | SCSI phase line, command/data. |
| B_SI_O–<br>B_SI_O+ | V22<br>V20 | | | SCSI phase line, input/output. |
| B_SMSG–<br>B_SMSG+ | R20<br>R21 | | | SCSI phase line, message. |
| B_SREQ–<br>B_SREQ+ | U21<br>V23 | | | Data handshake line from target device. |
| B_SACK–<br>B_SACK+ | N20<br>P21 | | | Data handshake signal from the initiator device. |
| B_SBSY–<br>B_SBSY+ | N23<br>N21 | | | SCSI bus arbitration signal, busy. |
| B_SATN–<br>B_SATN+ | M23<br>N22 | | | SCSI Attention, the initiator is requesting a message out phase. |
| B_SRST–<br>B_SRST+ | R23<br>R22 | | | SCSI bus reset. |
| B_SSEL–<br>B_SSEL+ | T23<br>T22 | | | SCSI bus arbitration signal, select device. |

1. LVD Mode: Negative and positive halves of the LVD Link signal pairs are shown for the SCSI Function B Control. SE Mode: The SCSI Function B Control signals are shown. All the positive (+) signals are at 0 Volts.

## 3.5 Flash ROM and Memory Interface Signals

Table 3.19 describes the signals for the Flash ROM and Memory Interface Signals group.

**Table 3.19 Flash ROM and Memory Interface Signals**

| Name | Bump | Type | Strength | Description |
|---|---|---|---|---|
| MWE/ | AC19 | O | 4 mA | **Memory Write Enable.** This pin is used as a write enable signal to an external flash memory. |
| MCE/ | AA18 | O | 4 mA | **Memory Chip Enable.** This pin is used as a chip enable signal to an external EPROM or flash memory device. |
| MOE/_ TESTOUT | Y18 | O | 4 mA | **Memory Output Enable.** This pin is used as an output enable signal to an external EPROM or flash memory during read operations. It is also used to test the connectivity of the SYM53C1010 signals in the "AND-tree" test mode. |
| MAS0/ | AC18 | O | 4 mA | **Memory Address Strobe 0**. This pin is used to latch in the least significant address byte (bits [7:0]) of an external EPROM or flash memory. Since the SYM53C1010 moves addresses eight bits at a time, this pin connects to the clock of an external bank of flip-flops that assemble up to a 20-bit address for the external memory. |
| MAS1/ | AA17 | O | 4 mA | **Memory Address Strobe 1.** This pin is used to latch in the most significant address byte (bits [15:8]) of an external EPROM or flash memory. Since the SYM53C1010 moves addresses eight bits at a time, this pin connects to the clock of an external bank of flip-flops that assemble up to a 20-bit address for the external memory. |
| MAD[7:0] | Y19, AA19, AC20, AB20, AA20, AC22, AB21, AC23. | I/O | 4 mA | **Memory Address/Data Bus.** This bus is used in conjunction with the memory address strobe pins and external address latches to assemble up to a 20-bit address for an external EPROM or flash memory. This bus first issues the least significant byte and finishes with the most significant bits. It is also used to write data to a flash memory or read data into the chip from external EPROM/flash memory. These pins have static pull-downs. |

## 3.6 Test Interface Signals

Table 3.20 describes the Test Interface Signals group. Table 3.20 is divided into Internal Test Signals and JTAG Signals.

**Table 3.20 Test Interface Signals**

| Name | Bump | Type | Strength | Description |
|---|---|---|---|---|
| **Internal Test Signals:** | | | | |
| SCAN_MODE | C22 | I | N/A | **Scan Mode**. For LSI Logic test purposes only. This pin has a static pull-down. |
| TEST_HSC | C23 | I | N/A | **Test Halt SCSI Clock.** For LSI Logic test purposes only. Pulled LOW internally. This signal can also cause a full chip reset. |
| TEST_PD | A2 | I | N/A | **Test Power Down.** For LSI Logic test purposes only. This pin has a static pull-down. |
| TEST_RST/ | C1 | I | N/A | **Test Reset.** For LSI Logic test purposes only. Pulled HIGH internally. |
| MOE/_ TESTOUT | Y18 | O | 4 mA | **Memory Output Enable.** This pin is used as an output enable signal to an external EPROM or flash memory during read operations. It is also used to test the connectivity of the SYM53C1010 signals in the “AND-tree” test mode. |
| **JTAG Signals:** | | | | |
| TCK | D1 | I | N/A | **Test Clock.** This pin provides the clock for the JTAG test logic. This pin has a static pull-up. |
| TMS | E3 | I | N/A | **Test Mode Select**. The signal received at TMS is decoded by the TAP controller to control JTAG test operations. This pin has a static pull-up. |
| TDI | E2 | I | N/A | **Test Data In.** This pin receives the serial test instructions for the JTAG test logic. This pin has a static pull-up. |
| TDO | E1 | O | 4 mA | **Test Data Out.** This pin is the serial output for test instructions and data from the JTAG test logic. |
| Reserved | AB14 | | | **Reserved.** Not Used. |

## 3.7 Power and Ground Signals

Table 3.21 describes the Power and Ground Signals group.

**Table 3.21 Power and Ground Signals**

| Name[1] | Bump | Type | Strength | Description |
|---|---|---|---|---|
| $V_{SS\_IO}$ | C3, C21, D4, D12, D20, K10–14, L10–14, M4, M10–14, M20, N10–14, P10–14, Y4, Y12, Y20, AA3, AA21. | G | N/A | Ground for PCI bus drivers/receivers, SCSI bus drivers, local memory interface drivers, and other I/O pins. |
| $V_{DD\_IO}$ | D7, D10, D14, D17, G4, G20, K4, K20, P4, P20, U4, U20, Y7, Y10, Y14, Y17. | P | N/A | Power for PCI bus drivers/receivers, SCSI bus drivers/receivers, local memory interface drivers/receivers, and other I/O pins. |
| $V_{DD}$_CORE | B22, B23, D3, E4, Y13, AB3, AB18, AB23, AC1. | P | N/A | Power for core logic. |
| $V_{SS}$_CORE | A22, D2, D21, F3, Y15, AB2, AA22, AB19, AC2, AC21. | G | N/A | Ground for core logic. |
| $V_{DD}$_A[2] | B2, C20 | P | N/A | Power for analog cells (clock quadrupler and DIFFSENS logic). |
| $V_{SS}$_A[2] | B20, C2 | G | N/A | Ground for analog cells (clock quadrupler and DIFFSENS logic). |
| $V_{DD}$_Bias | M22 | P | N/A | Power for SCSI Function A RBIAS circuit. |
| $V_{DD}$_Bias2 | A11 | P | N/A | Power for SCSI Function B RBIAS circuit. |
| RBIAS | M21 | I | N/A | Used to connect an external resistor to generate the LVD Link pad bias current. The resistor value should be 10 kΩ. Connect the other end of the resistor to $V_{DD}$. |
| NC | A1, A13, A23, B1, B3, B13, B16, B21, D16, P22, P23, U22, U23, AB22. | N/A | N/A | These pins are reserved or have no internal connection. |

1. The I/O driver pad rows and digital core have isolated power supplies as indicated by the “I/O” and “CORE” extensions on their respective $V_{SS}$ and $V_{DD}$ names. Connect the power and ground pins directly to the primary power and ground planes of the circuit board. Apply bypass capacitors of 0.01 µF between adjacent $V_{SS}$ and $V_{DD}$ pairs wherever possible. Do not connect bypass capacitors between $V_{SS}$ and $V_{DD}$ pairs that cross power and ground bus boundaries.
2. To reduce signal noise that can affect FSN functionality, place a ferrite bead in series with the $V_{DD}$-A and $V_{SS}$-A pins. The recommended rating of the bead is 150 Ohms at 100 MHz.

## 3.8 MAD Bus Programming

The MAD[7:0] pins, in addition to serving as the address/data bus for the local memory interface, also are used to program power-up options for the chip. A particular option is programmed allowing the internal pull-down current sink to pull the pin LOW at reset or by connecting a 4.7 kΩ resistor between the appropriate MAD[x] pin and $V_{DD}$. The pull-down resistors require that HC or HCT external components are used for the memory interface. A description of the MAD bus pins follows:

- **MAD[7] Serial EEPROM programmable option** – When pulled LOW by the internal pull-down current sink, the automatic data download is enabled. When pulled HIGH by an external resistor, the automatic data download is disabled. See Section 2.4, "Serial EEPROM Interface" and the Subsystem ID (SID) and Subsystem Vendor ID (SVID) register descriptions.
- **MAD[6]** – Reserved.
- **MAD[5]** – Reserved.
- **MAD[4] INTA/ routing enable** – Placing a pull-up resistor on this pin causes SCSI Function B interrupt requests to appear on the INTA/ pin, along with SCSI Function A interrupt requests, instead of on INTB/. Placing a pull-up resistor on this pin also programs the SCSI Function B Interrupt Pin (IP) register in PCI configuration space to 0x01 instead of 0x02.

  Placing no resistor on this pin causes SCSI Function B interrupt requests to appear on the INTB/ pin and programs the SCSI Function B Interrupt Pin (IP) register in PCI configuration space to 0x02.

- **MAD[3:1] pins** – These pins set the size of the external expansion ROM device attached. The encoding for these pins is listed in the following table. A “0” indicates a pull-down resistor is attached while a “1” indicates a pull-up resistor attached.

**Table 3.22 MAD[3:1] Pin Decoding**

| MAD[3:1] | Available Memory Space |
|---|---|
| 000 | 16 Kbytes |
| 001 | 32 Kbytes |
| 010 | 64 Kbytes |
| 011 | 128 Kbytes |
| 100 | 256 Kbytes |
| 101 | 512 Kbytes |
| 110 | 1024 Kbytes |
| 111 | no external memory present |

- **MAD[0] slow ROM pin** – When pulled up, it enables use of slower memory devices by including two extra data access cycles.

  Note: All MAD pins have internal pull-down resistors.

# Chapter 4
# Registers

This section contains descriptions of all SYM53C1010-33 registers. The term “set” refers to bits programmed to a binary one. Similarly, the term “cleared” refers to bits programmed to a binary zero. Do not access reserved bits. Reserved bit functions may change at any time. Unless otherwise indicated, all bits in the registers are active high; the feature is enabled by setting the bit. The bottom row of every register diagram presents the default register values, which are enabled after the chip is powered on or reset.

This chapter contains the following sections:



## 4.1 PCI Configuration Registers

To access the PCI Configuration registers, perform a configuration read or write to a device with its IDSEL pin asserted. The appropriate address value is in AD[10:8] during the address phase of the transaction. SCSI Function A is identified by a binary value of 0b000, and SCSI Function B by a value of 0b001. Each SCSI function contains the same register set with identical default values, except the Interrupt Pin (IP) register. Table 2.1 shows the PCI configuration registers implemented in the SYM53C1010.

All PCI-compliant devices, such as the SYM53C1010, support Vendor ID, Device ID, Command, and Status registers. Support of other PCI-compliant registers is optional. In the SYM53C1010, registers that are not supported are not writable and return all zeros when read. Only

those registers and bits that are currently supported by the SYM53C1010 are described in this chapter. Do not access bits marked as Reserved.

**Table 4.1 PCI Configuration Register Map**

| 31 | | 16 15 | 0 | Address | Page |
|---|---|---|---|---|---|
| Device ID | | Vendor ID | | 0x00 | 4-3 |
| Status | | Command | | 0x04 | 4-3 |
| Class Code (CC) | | | Revision ID (RID) | 0x08 | 4-7 |
| Reserved | Header Type (HT) | Latency Timer (LT) | Cache Line Size (CLS) | 0x0C | 4-7 |
| Base Address Register Zero (BAR0) (I/O) | | | | 0x10 | 4-9 |
| Base Address Register One (BAR1) (MEMORY) bits [31:0] | | | | 0x14 | 4-10 |
| Base Address Register Two (BAR2) (MEMORY) bits [31:0] | | | | 0x18 | 4-10 |
| Base Address Register Three (BAR3) (SCRIPTS RAM) bits [31:0] | | | | 0x1C | 4-11 |
| Base Address Register Four (BAR4) (SCRIPTS RAM) bits [31:0] | | | | 0x20 | 4-11 |
| Reserved | | | | 0x24 | 4-12 |
| Reserved | | | | 0x28 | 4-12 |
| Subsystem ID (SID) | | Subsystem Vendor ID (SVID) | | 0x2C | 4-12 |
| Expansion ROM Base Address (ERBA) | | | | 0x30 | 4-14 |
| Reserved | | | Capabilities Pointer (CP) | 0x34 | 4-15 |
| Reserved | | | | 0x38 | 4-15 |
| Max_Lat (ML) | Min_Gnt (MG) | Interrupt Pin (IP) | Interrupt Line (IL) | 0x3C | 4-16 |
| Power Management Capabilities (PMC) | | Next Item Pointer (NIP) | Capability ID (CID) | 0x40 | 4-18 |
| Data | Bridge Support Extensions (PMCSR_BSE) | Power Management Control/Status (PMCSR) | | 0x44 | 4-20 |

## Registers: 0x00–0x01

**Vendor ID**
**Read Only**

| 15 | | | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VID | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**VID** **Vendor ID** **[15:0]**
This 16-bit register identifies the manufacturer of the device. The Vendor ID is 0x1000.

## Registers: 0x02–0x03

**Device ID**
**Read Only**

| 15 | | | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DID | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

**DID** **Device ID** **[15:0]**
This 16-bit register identifies the particular device. The SYM53C1010-33 Device ID is 0x0020.

## Registers: 0x04–0x05

**Command**
**Read/Write**

| 15 | | | | | | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | | | | | | | SE | R | EPER | R | WIE | R | EBM | EMS | EIS |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

The SCSI Command register provides coarse control over a device's ability to generate and respond to PCI cycles. When a zero is written to this register, the SYM53C1010 is logically disconnected from the PCI bus for all accesses except configuration accesses.

| | | |
|---|---|---|
| **R** | **Reserved** | **[15:9]** |
| **SE** | **SERR/ Enable**<br>When this bit is set, the SYM53C1010 enables the SERR/ driver. SERR/ is disabled when this bit is cleared. The default value of this bit is zero. This bit and bit 6 must be set to report address parity errors. | **8** |
| **R** | **Reserved** | **7** |
| **EPER** | **Enable Parity Error Response**<br>When this bit is set, the SYM53C1010 detects parity errors on the PCI bus and reports these errors to the system. Only data parity checking is affected with this bit. The SYM53C1010 always generates parity for the PCI bus. | **6** |
| **R** | **Reserved** | **5** |
| **WIE** | **Write and Invalidate Enable**<br>When this bit is set, the SYM53C1010 can generate write and invalidate commands on the PCI bus. The WRIE bit in the Chip Test Three (CTEST3) register must also be set for the device to generate write and invalidate commands. | **4** |
| **R** | **Reserved** | **3** |
| **EBM** | **Enable Bus Mastering**<br>This bit controls the ability of the SYM53C1010 to act as a master on the PCI bus. A value of zero disables this device from generating PCI bus master accesses. A value of one allows the SYM53C1010 to behave as a bus master. The device must be a bus master to fetch SCRIPTS instructions and transfer data. | **2** |
| **EMS** | **Enable Memory Space**<br>This bit controls the ability of the SYM53C1010 to respond to Memory space accesses. A value of zero disables the device response. A value of one allows the SYM53C1010 to respond to Memory Space accesses at the address range specified by the Base Address Register One (BAR1) (MEMORY), Base Address Register Two (BAR2) (MEMORY), Base Address Register Three (BAR3) (SCRIPTS RAM), and the Base Address Register | **1** |

Four (BAR4) (SCRIPTS RAM) registers in the PCI configuration space.

**EIS** **Enable I/O Space** **0**
This bit controls the SYM53C1010 response to I/O space accesses. Clearing this bit disables the device response. Setting this bit allows the SYM53C1010 to respond to I/O Space accesses at the address range specified by the Base Address Register Zero (BAR0) (I/O) register in the PCI configuration space.

## Registers: 0x06–0x07

**Status**
**Read/Write**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DPE | SSE | RMA | RTA | R | DT[1:0] | | DPR | FBBC | R | 66C | NC | R | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

Reads to this register behave normally. Writes are slightly different in that bits can be cleared, but not set. A bit is cleared whenever the register is written, and the data in the corresponding bit location is a one. For example, to clear bit 15 and not affect any other bits, write the value 0x8000 to the register.

**DPE** **Detected Parity Error (from Slave)** **15**
This bit is set by the SYM53C1010 upon the detection of a data parity error, even if data parity error handling is disabled.

**SSE** **Signaled System Error** **14**
This bit is set whenever the device asserts the SERR/ signal.

**RMA** **Received Master Abort (from Master)** **13**
A master device should set this bit when its transaction (except for Special Cycle) is terminated with Master Abort.

**RTA** **Received Target Abort (from Master)** **12**
A master device should set this bit whenever its transaction is terminated by target abort.

**R** **Reserved** **11**

**DT[1:0]** **DEVSEL/ Timing** **[10:9]**
These bits encode the timing of DEVSEL/. The timings are encoded as:

| | |
|---|---|
| 0b00 | fast |
| 0b01 | medium |
| 0b10 | slow |
| 0b11 | reserved. |

These bits are read only and indicate the slowest time that a device asserts DEVSEL/ for any bus command except Configuration Read and Configuration Write. The SYM53C1010 supports a value of 0b01.

**DPR** **Data Parity Error Reported** **8**
This bit is set when the following conditions are met:

- The bus agent asserts PERR/ itself or observes PERR/ asserted and;
- The agent setting this bit acted as the bus master for the errant operation and;
- The Parity Error Response bit in the Command register is set.

**FBBC** **Fast Back-to-Back Capable** **7**
This bit is zero.

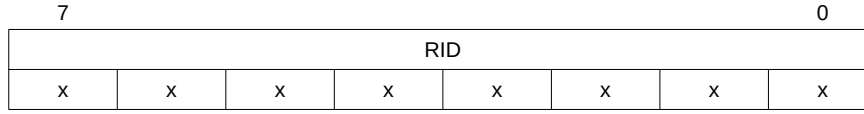
**R** **Reserved** **6**

**66C** **Reserved** **5**
This bit is always set to zero.

**NC** **New Capabilities** **4**
This bit is set to indicate a list of extended capabilities such as PCI Power Management. This bit is read only.

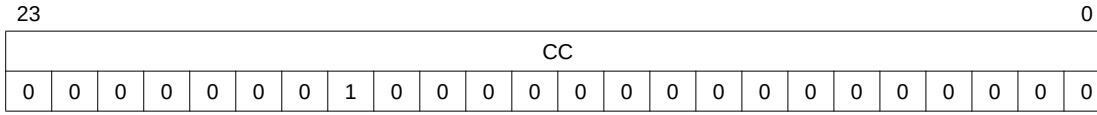
**R** **Reserved** **[3:0]**

## Register: 0x08
**Revision ID (RID)**
**Read Only**



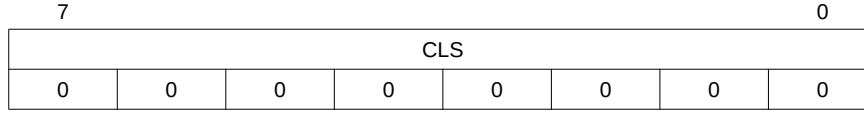

**RID** **Revision ID** **[7:0]**
This register indicates the current revision level of the device.

## Registers: 0x09–0x0B
**Class Code (CC)**
**Read Only**




**CC** **Class Code** **[23:0]**
This 24-bit register identifies the generic function of the device. The upper byte of this register is a base class code, the middle byte is a subclass code, and the lower byte identifies a specific register-level programming interface. The value of this register is 0x010000, which identifies a SCSI controller.

## Register: 0x0C
**Cache Line Size (CLS)**
**Read/Write**

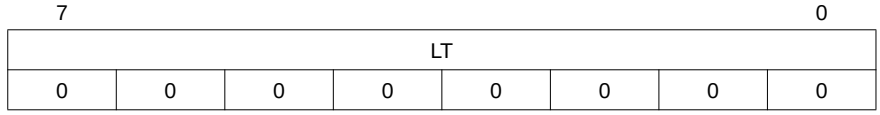



**CLS** **Cache Line Size** **[7:0]**
This register specifies the system cache line size in units of 32-bit words. The value in this register is used by the device to determine whether to use Write and Invalidate or Write commands for performing write cycles, and whether to use Read, Read Line, or Read Multiple

commands for performing read cycles as a bus master. Devices participating in the caching protocol use this field to determine when to retry burst accesses at cache line boundaries. These devices can ignore the PCI cache support lines (SDONE and SB0/) if this register is set to 0. If this register is programmed to a number which is not a power of 2, the device does not use PCI performance commands to execute data transfers.

## Register: 0x0D

**Latency Timer (LT)**
**Read/Write**




**LT** **Latency Timer** **[7:0]**

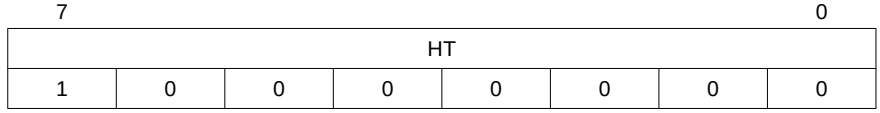
The Latency Timer register specifies, in units of PCI bus clocks, the value of the Latency Timer for this PCI bus master. The SCSI functions of the SYM53C1010 support this timer. All eight bits are writable, allowing latency values of 0–255 PCI clocks. Use the following equation to calculate an optimum latency value for the SCSI functions of the SYM53C1010.

Latency = 2 + (Burst Size * (typical wait states + 1))

Values greater than optimum are also acceptable.

## Register: 0x0E

**Header Type (HT)**
**Read Only**

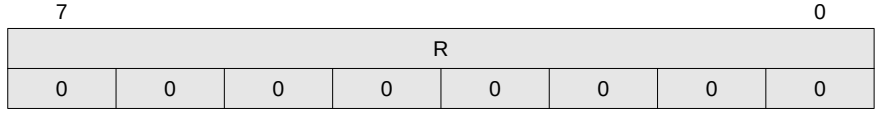



**HT** **Header Type** **[7:0]**

This 8-bit register identifies the layout of bytes 0x10 through 0x3F in configuration space and also whether or not the device contains multiple functions. Since the SYM53C1010 is a multifunction controller the value of this register is 0x80.

## Register: 0x0F

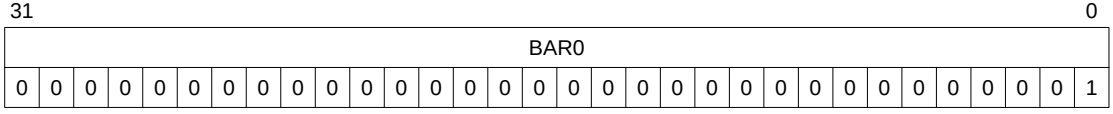
**Reserved**




This register is reserved.

## Registers: 0x10–0x13

**Base Address Register Zero (BAR0) (I/O)**
**Read/Write**

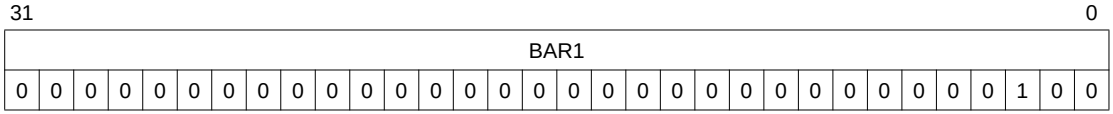



**BAR0** **Base Address Register Zero - I/O** **[31:0]**

This base address register is used to map the operating register set into I/O space. The SYM53C1010 requires 256 bytes of I/O space for this base address register. Bit 0 is hardwired to one. Bit 1 is reserved and returns a zero on all reads. All other bits are used to map the device into I/O space. For detailed information on the operation of this register, refer to the PCI 2.2 specification.

## Registers: 0x14–0x17

**Base Address Register One (BAR1) (MEMORY)**
**Read/Write**

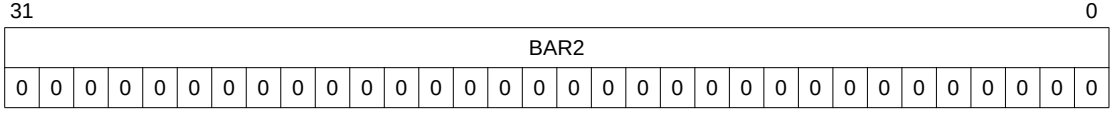



**BAR1** **Base Address Register One** **[31:0]**

This base address register, in conjunction with Base Address Register Two (BAR2) (MEMORY), maps SCSI operating registers into memory space and represents the lower 32 bits of the memory address. Bits [9:0] are hardwired to 0b0000000100. The default value of this register is 0x00000004. The SYM53C1010 requires 1024 bytes of memory space. For detailed information on the operation of this register, refer to the PCI 2.2 specification.

## Registers: 0x18–0x1B

**Base Address Register Two (BAR2) (MEMORY)**
**Read/Write**

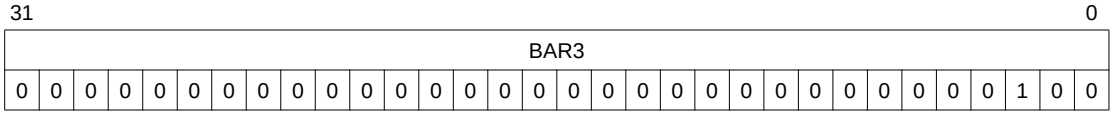



**BAR2** **Base Address Register Two** **[31:0]**

This base address register, in conjunction with Base Address Register One (BAR1) (MEMORY), maps SCSI operating registers into memory space and represents the upper 32 bits of the memory address. The default value of this register is 0x00000000. The SYM53C1010 requires 1024 bytes of memory space. For detailed information on the operation of this register, refer to the PCI 2.2 specification.

## Registers: 0x1C–0x1F

**Base Address Register Three (BAR3) (SCRIPTS RAM)**
**Read/Write**

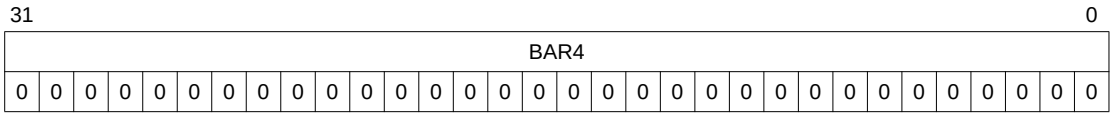



**BAR3** **Base Address Register Three** **[31:0]**

This base address register, in conjunction with Base Address Register Four (BAR4) (SCRIPTS RAM), is used to map the SCRIPTS RAM into memory space and represents the lower 32 bits of the memory address. Bits [12:0] are hardwired to 0b0000000000100. The default value of this register is 0x00000004. The SYM53C1010 requires 8192 bytes of memory space for SCRIPTS RAM. For detailed information on the operation of this register, refer to the PCI 2.2 specification.

## Registers: 0x20–0x23

**Base Address Register Four (BAR4) (SCRIPTS RAM)**
**Read/Write**




**BAR4** **Base Address Register Four** **[31:0]**

This base address register, in conjunction with Base Address Register Three (BAR3) (SCRIPTS RAM), is used to map the SCRIPTS RAM into memory space and represents the upper 32 bits of the memory address. The default value of this register is 0x00000000. The SYM53C1010 requires 8192 bytes of memory space for SCRIPTS RAM. For detailed information on the operation of this register, refer to the PCI 2.2 specification.

## Registers: 0x24–0x27
**Reserved**

| 31 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

This register is reserved.

## Registers: 0x28–0x2B
**Reserved**

| 31 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

This register is reserved.

## Registers: 0x2C–0x2D
**Subsystem Vendor ID (SVID)**
**Read Only**

| 15 | | | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SVID | | | | | | | | | | | | | | | |
| If MAD7 is HIGH | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| If MAD7 is LOW | | | | | | | | | | | | | | | |
| x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |

**SVID** **Subsystem Vendor ID** **[15:0]**

This 16-bit register is used to uniquely identify the vendor manufacturing the add-in board or subsystem where this PCI device resides. It provides a mechanism for an add-in card vendor to distinguish its cards from another vendor's cards, even if the cards have the same PCI controller installed on them (and therefore the same Vendor ID and Device ID).

If the external serial EEPROM interface is enabled (MAD[7] LOW), this register is automatically loaded at power-up from the external serial EEPROM and contains

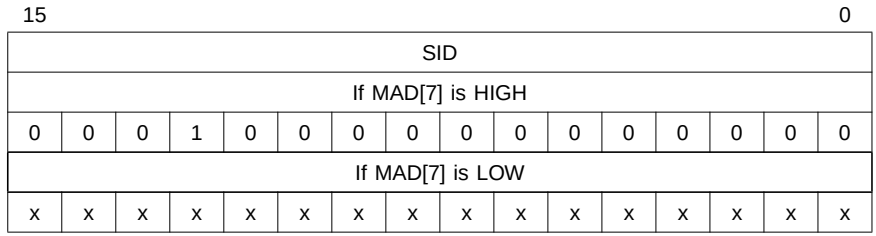
the value downloaded from the serial EEPROM or, if the download fails, a value of 0x0000.

If the external serial EEPROM interface is disabled (MAD[7] HIGH), this register returns a value of 0x1000. The 16-bit value that should be stored in the external serial EEPROM for this register is the vendor's PCI Vendor ID. This value must be obtained from the PCI Special Interest Group (SIG). Please see Section 2.4, "Serial EEPROM Interface" for more information on downloading a value for this register.

## Registers: 0x2E–0x2F

### Subsystem ID (SID)

### Read Only



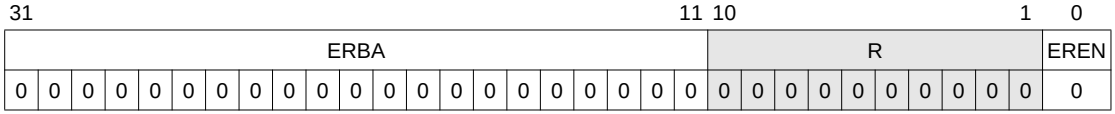

**SID** **Subsystem ID** **[15:0]**

This 16-bit register is used to uniquely identify the add-in board or subsystem where this PCI device resides. It provides a mechanism for an add-in card vendor to distinguish its cards from one another even if the cards have the same PCI controller installed on them (and therefore the same Vendor ID and Device ID).

If the external serial EEPROM interface is enabled (MAD[7] is LOW), this register is automatically loaded at power-up from the external serial EEPROM and contains the value downloaded from the serial EEPROM or, if the download fails, a value of 0x0000.

If the external serial EEPROM is disabled (MAD[7] pulled HIGH), the register returns a value of 0x1000. The 16-bit value stored in the external serial EEPROM is vendor specific. Please see Section 2.4, "Serial EEPROM Interface" for additional information on downloading a value for this register.

## Registers: 0x30–0x33
### Expansion ROM Base Address (ERBA)
### Read/Write




This four-byte register handles the base address and size information for the expansion ROM.

**ERBA** **Expansion ROM Base Address** **[31:11]**

Bits [31:11] correspond to the upper 21 bits of the expansion ROM base address. The host system detects the size of the external memory by first writing the Expansion ROM Base Address (ERBA) register with all ones and then reading back the register. The SCSI functions of the SYM53C1010 respond with zeros in all don't care locations. The least significant one that remains represents the binary version of the external memory size. For example, to indicate an external memory size of 32 Kbytes, this register, when written with ones and read back, returns ones in the upper 17 bits.

The size of the external memory is set through MAD[3:1]. Please see Section 3.8, "MAD Bus Programming" for the possible size encodings available.

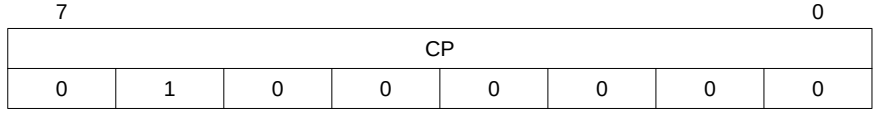
**R** **Reserved** **[10:1]**

**EREN** **Expansion ROM Enable** **0**

The expansion ROM Enable bit, bit 0, is used to control whether or not the device accepts accesses to its expansion ROM. When the bit is set, address decoding is enabled, and a device is used with or without an expansion ROM depending on the system configuration.

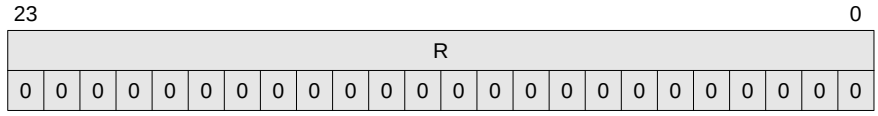
Note: To access the external memory interface, also set the Memory Space bit in the Command register.

## Register: 0x34
**Capabilities Pointer (CP)**
**Read Only**



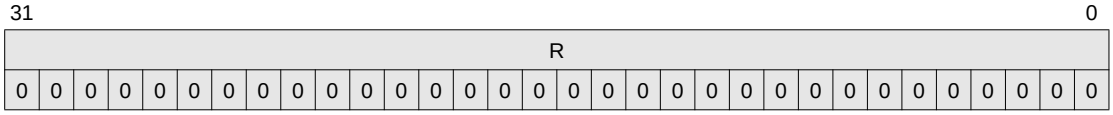

**CP** **Capabilities Pointer** **[7:0]**
This register indicates that the first extended capability register is located at offset 0x40 in PCI Configuration Space.

## Registers: 0x35–0x37
**Reserved**




This register is reserved.

## Registers: 0x38–0x3B
**Reserved**




This register is reserved.

## Register: 0x3C
**Interrupt Line (IL)**
**Read/Write**

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| IL | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**IL** **Interrupt Line** **[7:0]**
This register is used to communicate interrupt line routing information. POST software writes the routing information into this register as it configures the system. The value in this register tells which input of the system interrupt controller(s) the device's interrupt pin is connected to. Values in this register are specified by system architecture.

## Register: 0x3D
**Interrupt Pin (IP)**
**Read Only**

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| IP | | | | | | | |
| SCSI Function A | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| SCSI Function B if MAD[4] pulled LOW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| SCSI Function B if MAD[4] pulled HIGH | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

**IP** **Interrupt Pin** **[7:0]**
This register is unique to each SCSI function. It tells which interrupt pin the device uses. Its value is set to 0x01 for the Function A (INTA/) signal, and 0x02 for the Function B (INTB/) signal at power-up if MAD[4] is pulled LOW. The Function B value is set to 0x01 (INTA/) if MAD[4] is pulled HIGH.

Note: Please see Section 3.8, "MAD Bus Programming" for additional information.

## Register: 0x3E

**Min_Gnt (MG)**
**Read Only**

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| MG | | | | | | | |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |

**MG** **Min_Gnt** **[7:0]**

This register is used to specify the desired settings for latency timer values. Min_Gnt is used to specify how long a burst period the device needs. The value specified in these registers is in units of 0.25 microseconds. The SYM53C1010 sets this register to 0x11.

## Register: 0x3F

**Max_Lat (ML)**
**Read Only**

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| ML | | | | | | | |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |

**ML** **Max_Lat** **[7:0]**

This register is used to specify the desired settings for latency timer values. Max_Lat is used to specify how often the device needs to gain access to the PCI bus. The value specified in this register is in units of 0.25 microseconds. The SYM53C1010 SCSI function sets this register to 0x12 indicating it needs the bus every 4.5 µs to maintain a data stream of 160 Mbytes/s.

## Register: 0x40

**Capability ID (CID)**
**Read Only**

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| CID | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

**CID** **Capability ID** **[7:0]**

This register indicates the type of data structure currently being used. It is set to 0x01, indicating the Power Management Data Structure.

## Register: 0x41

**Next Item Pointer (NIP)**
**Read Only**

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| NIP | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**NIP** **Next Item Pointer** **[7:0]**

Bits [7:0] contain the offset location of the next item in the function's capabilities list. The SYM53C1010 has these bits set to zero indicating no further extended capabilities registers exist.

## Registers: 0x42–0x43

**Power Management Capabilities (PMC)**
**Read Only**

| 15 | | | | 11 | 10 | 9 | 8 | | 6 | 5 | 4 | 3 | 2 | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PMES | | | | | D2S | D1S | AUX_C | | | DSI | R | PMEC | VER[2:0] | | |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

**PMES** **PME_Support** **[15:11]**

Bits [15:11] define the power management states in which the SYM53C1010 will assert the PME pin. These bits are all set to zero because the SYM53C1010 does not provide a PME signal.

| | | |
|---|---|---|
| **D2S** | **D2_Support**<br>The SYM53C1010 sets this bit to indicate support for power management state D2. Bits 9 and 10 are set to indicate support for the D1 and D2 power states. | **10** |
| **D1S** | **D1_Support**<br>The SYM53C1010 sets this bit to indicate support for power management state D1. Bits 9 and 10 are set to indicate support for the D1 and D2 power states. | **9** |
| **AUX_C** | **Aux_Current**<br>The SYM53C1010 always returns zeros. This feature is not supported. | **[8:6]** |
| **DSI** | **Device Specific Initialization**<br>This bit is cleared to indicate that the SYM53C1010 requires no special initialization before the generic class device driver is able to use it. | **5** |
| **R** | **Reserved** | **4** |
| **PMEC** | **PME Clock**<br>Bit 3 is cleared because the SYM53C1010 does not provide a PME pin. | **3** |
| **VER[2:0]** | **Version**<br>These three bits are set to 0b010 to indicate that the SYM53C1010 complies with Revision 1.1 of the PCI Power Management Interface Specification. | **[2:0]** |

## Registers: 0x44–0x45

**Power Management Control/Status (PMCSR)**
**Read/Write**

| 15 | 14 | 13 | 12 | | | 9 | 8 | 7 | | | | | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PST | DSCL | | DSLT | | | | PEN | R | | | | | | PWS[1:0] | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**PST** **PME_Status** **15**
The SYM53C1010 always returns a zero for this bit, indicating that PME signal generation is not supported from D3cold.

**DSCL** **Data_Scale** **[14:13]**
The SYM53C1010 does not support the Data register. Therefore, these two bits are always cleared.

**DSLT** **Data_Select** **[12:9]**
The SYM53C1010 does not support the Data register. Therefore, these four bits are always cleared.

**PEN** **PME_Enable** **8**
The SYM53C1010 always returns zero for this bit to indicate that PME assertion is disabled.

**R** **Reserved** **[7:2]**

**PWS[1:0]** **Power State** **[1:0]**
Bits [1:0] are used to determine the current power state of the SYM53C1010. They are used to place the SYM53C1010 in a new power state. Power states are defined as:

| | |
|---|---|
| 0b00 | D0 |
| 0b01 | D1 |
| 0b10 | D2 |
| 0b11 | D3 hot |

Note: See the Section 2.5, “Power Management” for descriptions of the Power Management States.

## Register: 0x46

**Bridge Support Extensions (PMCSR_BSE)**
**Read Only**

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| BSE | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**BSE** **Bridge Support Extensions** **[7:0]**
This register indicates PCI Bridge specific functionality. The SYM53C1010 always returns 0x00.

## Register: 0x47

**Data**
**Read Only**

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| DATA | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**DATA** **Data** **[7:0]**
This register provides an optional mechanism for the function to report state-dependent operating data. The SYM53C1010 always returns 0x00.

## 4.2 SCSI Registers

The control registers for the SCSI core are directly accessible from the PCI bus using Memory or I/O mapping. SCSI Function A and SCSI Function B contain the same register set with identical default values, except the Interrupt Pin registers. The address map of the SCSI registers is shown in Table 4.2.

The eight 32-bit phase mismatch registers contain the byte count and addressing information required to update the Direct, Indirect, or Table Indirect BMOV instructions with new byte counts and addresses. The phase mismatch registers are the Phase Mismatch Jump Address One (PMJAD1), Phase Mismatch Jump Address Two (PMJAD2), Remaining Byte Count (RBC), Updated Address (UA), Entry Storage Address (ESA), Instruction Address (IA), SCSI Byte Count (SBC), and the Cumulative SCSI Byte Count (CSBC). All the phase mismatch registers can be read/written using the Load and Store SCRIPTS instructions, Memory-to-Memory Moves, Read/Write SCRIPTS instructions, or the CPU with SCRIPTS not running.

Note: The only registers that the host CPU can access while the SYM53C1010 is executing SCRIPTS are the Interrupt Status Zero (ISTAT0), Interrupt Status One (ISTAT1), Mailbox Zero (MBOX0), and Mailbox One (MBOX1) registers; attempts to access other registers interfere with the operation of the chip. However, all operating registers are accessible with SCRIPTS. All read data is synchronized and stable when presented to the PCI bus.

Do not access reserved bits or registers.

## Table 4.2 SCSI Register Map

| 31 24 | 23 16 | 15 8 | 7 0 | Address | Page |
|---|---|---|---|---|---|
| SCNTL3 | SCNTL2 | SCNTL1 | SCNTL0 | 0x00 | 4-24 |
| GPREG | SDID | SXFER | SCID | 0x04 | 4-33 |
| SBCL | SSID | SOCL | SFBR | 0x08 | 4-37 |
| SSTAT2 | SSTAT1 | SSTAT0 | DSTAT | 0x0C | 4-40 |
| DSA | | | | 0x10 | 4-47 |
| MBOX1 | MBOX0 | ISTAT1 | ISTAT0 | 0x14 | 4-47 |
| CTEST3 | CTEST2 | CTEST1 | CTEST0 | 0x18 | 4-53 |
| TEMP | | | | 0x1C | 4-56 |
| CTEST6 | CTEST5 | CTEST4 | Reserved | 0x20 | 4-57 |
| DCMD | DBC | | | 0x24 | 4-61 |
| DNAD | | | | 0x28 | 4-62 |
| DSP | | | | 0x2C | 4-63 |
| DSPS | | | | 0x30 | 4-63 |
| SCRATCH A | | | | 0x34 | 4-64 |
| DCNTL | SBR | DIEN | DMODE | 0x38 | 4-64 |
| ADDER | | | | 0x3C | 4-71 |
| SIST1 | SIST0 | SIEN1 | SIEN0 | 0x40 | 4-71 |
| GPCNTL | Reserved | SWIDE | Reserved | 0x44 | 4-79 |
| RESPID1 | RESPID0 | STIME1 | STIME0 | 0x48 | 4-81 |
| STEST3 | STEST2 | STEST1 | STEST0 | 0x4C | 4-85 |
| CSO | STEST4 | SIDL | | 0x50 | 4-90 |
| CCNTL1 | CCNTL0 | SODL | | 0x54 | 4-92 |
| CCNTL3 | Reserved | SBDL | | 0x58 | 4-96 |
| SCRATCH B | | | | 0x5C | 4-98 |
| SCRATCH C–SCRATCH R | | | | 0x60–0x9F | 4-99 |
| MMRS | | | | 0xA0 | 4-99 |
| MMWS | | | | 0xA4 | 4-100 |
| SFS | | | | 0xA8 | 4-100 |
| DRS | | | | 0xAC | 4-101 |
| SBMS | | | | 0xB0 | 4-101 |
| DBMS | | | | 0xB4 | 4-102 |
| DNAD64 | | | | 0xB8 | 4-102 |
| AIPCNTL1 | AIPCNTL0 | Reserved | SCNTL4 | 0xBC | 4-102 |
| PMJAD1 | | | | 0xC0 | 4-112 |
| PMJAD2 | | | | 0xC4 | 4-113 |
| RBC | | | | 0xC8 | 4-113 |
| UA | | | | 0xCC | 4-114 |
| ESA | | | | 0xD0 | 4-115 |
| IA | | | | 0xD4 | 4-115 |
| Reserved | SBC | | | 0xD8 | 4-116 |
| CSBC | | | | 0xDC | 4-117 |
| CRCCNTL1 | CRCCNTL0 | CRCPBV | | 0xE0 | 4-117 |
| CRCD | | | | 0xE4 | 4-120 |
| Reserved | | | | 0xE8–0xEF | 4-121 |
| Reserved | | DFBC | | 0xF0 | 4-122 |
| Reserved | | | | 0xF4–0xFF | 4-122 |

## Register: 0x00
### SCSI Control Zero (SCNTL0)
### Read/Write

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ARB[1:0] | | START | WATN | EPC | CRCRP | AAP | TRG |
| 1 | 1 | 0 | 0 | 0 | x | 0 | 0 |

**ARB[1:0]** **Arbitration Mode Bits 1 and 0** **[7:6]**
A combination of the ARB bits selects either Simple or Full arbitration.

| ARB1 | ARB0 | Arbitration Mode |
|---|---|---|
| 0 | 0 | Simple arbitration |
| 0 | 1 | Reserved |
| 1 | 0 | Reserved |
| 1 | 1 | Full arbitration, selection/reselection |

**Simple Arbitration**

1. The SYM53C1010 SCSI function waits for a bus free condition to occur.
2. It asserts SBSY/ and its SCSI ID, contained in the SCSI Chip ID (SCID) register, onto the SCSI bus. If the SSEL/ signal is asserted by another SCSI device, the SYM53C1010 SCSI function deasserts SBSY/, deasserts its ID, and sets the Lost Arbitration bit (bit 3) in the SCSI Status Zero (SSTAT0) register.
3. After an arbitration delay, the CPU should read the SCSI Bus Data Lines (SBDL) register to check if a higher priority SCSI ID is present. If no higher priority ID bit is set, and the Lost Arbitration bit is not set, the SYM53C1010 SCSI function wins arbitration.
4. Once the SYM53C1010 SCSI function wins arbitration, SSEL/ must be asserted using the SCSI Output Control Latch (SOCL) for a bus clear and a bus settle delay (1.2 µs). Then, a low level selection is performed.

**Full Arbitration, Selection/Reselection**

1. The SYM53C1010 SCSI function waits for a bus free condition.
2. It asserts SBSY/ and its SCSI ID onto the SCSI bus. The SCSI ID asserted is the highest priority ID stored in the SCSI Chip ID (SCID) register.
3. If the SSEL/ signal is asserted by another SCSI device or if the SYM53C1010 SCSI function detects a higher priority ID, the SYM53C1010 SCSI function deasserts SBSY/, deasserts its ID, sets the Lost Arbitration bit, bit 3 in the SCSI Status Zero (SSTAT0) register, and waits until the next bus free state to try arbitration again.
4. The SYM53C1010 SCSI function repeats arbitration until it wins control of the SCSI bus. When it wins, the Won Arbitration bit is set in the SCSI Status Zero (SSTAT0) register, bit 2.
5. The SYM53C1010 SCSI function performs selection by asserting SSEL/, the target's ID (stored in the SCSI Destination ID (SDID) register), and the SYM53C1010's ID (stored in the SCSI Chip ID (SCID) register) onto the SCSI bus.
6. After a selection is complete, the Function Complete bit is set in the SCSI Interrupt Status Zero (SIST0) register, bit 6.
7. If a selection time-out occurs, the Selection Time-Out bit is set in the SCSI Interrupt Status One (SIST1) register, bit 2.

**START** **Start Sequence** **5**

When this bit is set, the SYM53C1010 starts the arbitration sequence indicated by the Arbitration Mode bits. The Start Sequence bit is accessed directly in low level mode; during SCSI SCRIPTS operations, this bit is controlled by the SCRIPTS processor. Do not start an arbitration sequence if the connected (CON) bit in the SCSI Control One (SCNTL1) register, bit 4, is set. This bit indicates that the SYM53C1010 is already connected to the SCSI bus. This bit is automatically cleared when

the arbitration sequence is complete. If a sequence is aborted, check bit 4 in the SCSI Control One (SCNTL1) register to verify that the SYM53C1010 is not connected to the SCSI bus.

**WATN** **Select with SATN/ on a Start Sequence** **4**

When this bit is set and the SCSI function is in the initiator mode, the SATN/ signal is asserted during selection of a SCSI target device. This is to inform the target that the SYM53C1010 SCSI function has a message to send. If a selection time-out occurs while attempting to select a target device, SATN/ is deasserted at the same time SSEL/ is deasserted. When this bit is cleared, the SATN/ signal is not asserted during selection. When executing SCSI SCRIPTS, this bit is controlled by the SCRIPTS processor, but manual setting is possible in the low level mode.

**EPC** **Enable Parity/CRC/AIP Checking** **3**

When this bit is set and the SCSI transfers are asynchronous or ST synchronous, the SCSI data bus is checked for odd parity when data is received from the SCSI bus in either the initiator or the target mode. If a parity error is detected, bit 0 of the SCSI Interrupt Status Zero (SIST0) register is set and an interrupt may be generated.

When this bit is set SCSI transfers are DT synchronous, the CRC is checked when the target requests a CRC transfer using the DP0 signal on the SCSI bus. If a CRC error is detected, bit 0 of the SCSI Interrupt Status Zero (SIST0) register is set and an interrupt may be generated.

If the SYM53C1010 SCSI function is operating in the initiator mode and a parity error or CRC error is detected, SATN/ can optionally be asserted, but the transfer continues until the target changes phase or the block move in which the parity error was detected completes. When this bit is clear, parity and CRC errors are not reported.

**CRCRP** **CRC Request Pending** **2**

When this bit is set, the SYM53C1010 SCSI function has an outstanding CRC request pending. When this bit is set and the SYM53C1010 is in target mode, a Block Move of

zero should not be issued. If a Block Move of zero is issued, back-to-back CRC requests are issued. Back-to-back CRC requests are illegal.

**AAP** **Assert SATN/ on Parity/CRC/AIP Error** **1**

When this bit is set, the SYM53C1010 SCSI function automatically asserts the SATN/ signal upon detection of a parity error or CRC error. SATN/ is only asserted in the initiator mode. The SATN/ signal is asserted before deasserting SACK/ during the byte transfer with the parity error. Also set the Enable Parity/CRC/AIP Checking bit for the SYM53C1010 SCSI function to assert SATN/ in this manner. A parity error or CRC error is detected on data received from the SCSI bus.

If the Assert SATN/ on Parity/CRC/AIP Error bit is cleared or the Enable Parity/CRC/AIP Checking bit is cleared, SATN/ is not automatically asserted on the SCSI bus when a Parity/CRC/AIP error is received.

**TRG** **Target Mode** **0**

This bit determines the default operating mode of the SYM53C1010 SCSI function. The user must manually set the target or initiator mode. This is done using the SCRIPTS language (SET TARGET or CLEAR TARGET). When this bit is set, the chip is a target device. When this bit is cleared, the SYM53C1010 SCSI function is an initiator device.

Caution: Writing this bit while not connected may cause the loss of a selection or reselection due to the changing of target or initiator modes.

## Register: 0x01
### SCSI Control One (SCNTL1)
### Read/Write

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| R | ADB | DHP | CON | RST | AESP | IARB | R |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**R** **Reserved** **7**

**ADB** **Assert SCSI Data Bus** **6**

When this bit is set, the SYM53C1010 SCSI function drives the contents of the SCSI Output Data Latch (SODL) register onto the SCSI data bus. When the SYM53C1010 SCSI function is an initiator, the SCSI I/O signal must be inactive to assert the SODL contents onto the SCSI bus. When the SYM53C1010 SCSI function is a target, the SCSI I/O signal must be active to assert the SODL contents onto the SCSI bus. The contents of the SODL register can be asserted at any time, even before the SYM53C1010 SCSI function is connected to the SCSI bus. Clear this bit when executing SCSI SCRIPTS. It is normally used only for diagnostics testing or operation in low level mode.

**DHP** **Disable Halt on Parity/CRC/AIP Error or ATN (Target Only)** **5**

The DHP bit is only defined for the target mode. When this bit is cleared, the SYM53C1010 SCSI function halts the SCSI data transfer when a Parity/CRC/AIP error is detected or when the SATN/ signal is asserted. If SATN/ or a Parity/CRC/AIP error is received in the middle of a data transfer, the SYM53C1010 SCSI function may transfer up to three additional bytes before halting to synchronize between internal core cells. During synchronous operation, the SYM53C1010 SCSI function transfers data until there are no outstanding synchronous offsets. If the SYM53C1010 SCSI function is receiving data, any data residing in the DMA FIFO is sent to memory before halting.

When this bit is set, the SYM53C1010 SCSI function does not halt the SCSI transfer when SATN/ or a Parity/CRC/AIP error is received.

**CON** **Connected** **4**

This bit is automatically set any time the SYM53C1010 SCSI function is connected to the SCSI bus as an initiator or as a target. It is set after the SYM53C1010 SCSI function successfully completes arbitration or when it has responded to a bus-initiated selection or reselection. This bit is also set after the chip wins simple arbitration when operating in low level mode. When this bit is cleared, the SYM53C1010 SCSI function is not connected to the SCSI bus.

The CPU can force a connected or disconnected condition by setting or clearing this bit.

**RST** **Assert SCSI RST/ Signal** **3**

Setting this bit asserts the SRST/ signal. The SRST/ output remains asserted until this bit is cleared. The 25 µs minimum assertion time defined in the SCSI specification must be timed out by the controlling microprocessor or a SCRIPTS loop.

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

When this bit is set, the SYM53C1010 SCSI function asserts even parity. It forces a SCSI parity error on each byte sent to the SCSI bus from the chip. If parity checking is enabled, then the SYM53C1010 SCSI function checks data received for odd parity. This bit is used for diagnostics testing and is cleared for normal operation.

**IARB** **Immediate Arbitration** **1**

Setting this bit causes the SCSI core to immediately begin arbitration once a Bus Free phase is detected following an expected SCSI disconnect. This bit is useful for multithreaded applications. The ARB[1:0] bits in SCSI Control Zero (SCNTL0) are set for full arbitration and selection before setting this bit.

Arbitration is retried until won. At that point, the SYM53C1010 SCSI function holds SBSY and SSEL asserted, and waits for a select or reselect sequence. The Immediate Arbitration bit is cleared automatically when the selection or reselection sequence is completed or times out.

An unexpected disconnect condition clears IARB with it attempting arbitration. See the SCSI Disconnect

Unexpected bit (SCSI Control Two (SCNTL2), bit 7) for more information on expected versus unexpected disconnects.

It is possible to abort an immediate arbitration sequence. First, set the Abort bit in the Interrupt Status Zero (ISTAT0) register. Then one of two things eventually happens:

- The Won Arbitration bit (SCSI Status Zero (SSTAT0), bit 2) will be set. In this case, the Immediate Arbitration bit needs to be cleared. This completes the abort sequence and disconnects the chip from the SCSI bus. If it is not acceptable to go to the Bus Free phase immediately following the arbitration phase, it is possible to perform a low level selection instead.
- The abort completes because the SYM53C1010 SCSI function loses arbitration. This is detected by the clearing of the Immediate Arbitration bit. Do not use the Lost Arbitration bit (SCSI Status Zero (SSTAT0), bit 3) to detect this condition. In this case take no further action.

**R** **Reserved** **0**

## Register: 0x02

### SCSI Control Two (SCNTL2)

### Read/Write

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SDU | CHM | R | | WSS | VUE0 | VUE1 | WSR |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**SDU** **SCSI Disconnect Unexpected** **7**

This bit is valid in the initiator mode only. When this bit is set, the SCSI core is not expecting the SCSI bus to enter the Bus Free phase. If it does, an unexpected disconnect error is generated (see the Unexpected Disconnect bit in the SCSI Interrupt Status Zero (SIST0) register, bit 2). During normal SCRIPTS mode operation, this bit is set automatically whenever the SCSI core is reselected, or successfully selects another SCSI device. The SDU bit should be cleared with a register write (Move 0x00 to SCSI Control Two (SCNTL2)) before the SCSI core

expects a disconnect to occur, normally prior to sending an Abort, Abort Tag, Bus Device Reset, Clear Queue or Release Recovery message, or before deasserting SACK/ after receiving a Disconnect command or Command Complete message.

**CHM** **Chained Mode** **6**

This bit determines whether or not the SCSI core is programmed for chained SCSI mode. This bit is automatically set by the Chained Block Move (CHMOV) SCRIPTS instruction and is automatically cleared by the Block Move SCRIPTS instruction (MOVE). For more information, refer to Section 2.2.18, “Chained Block Moves.”

**R** **Reserved** **[5:4]**

**WSS** **Wide SCSI Send** **3**

When read, this bit returns the value of the Wide SCSI Send (WSS) flag. Asserting this bit clears the WSS flag. This clearing function is self-clearing. For more information refer to Section 2.2.18, “Chained Block Moves.”

**VUE0** **Vendor Unique Enhancements, Bit 0** **2**

This bit is a read only value indicating whether the group code field in the SCSI instruction is standard or vendor unique. If cleared, the bit indicates standard group codes; if set, the bit indicates vendor unique group codes. The value in this bit is reloaded at the beginning of all asynchronous target receives.

**VUE1** **Vendor Unique Enhancement, Bit 1** **1**

This bit is used to disable the automatic byte count reload during Block Move instructions in the Command phase. If this bit is cleared, the device reloads the Block Move byte count if the first byte received is one of the standard group codes. If this bit is set, the device does not reload the Block Move byte count, regardless of the group code.

**WSR** **Wide SCSI Receive** **0**

When read, this bit returns the value of the Wide SCSI Receive (WSR) flag. Setting this bit clears the WSR flag. This bit is self-clearing.

For more information refer to Section 2.2.18, “Chained Block Moves.”

## Register: 0x03

### SCSI Control Three (SCNTL3)

### Read/Write

| 7 | 6 | | 4 | 3 | 2 | | 0 |
|---|---|---|---|---|---|---|---|
| R | SCF[2:0] | | | EWS | R | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

This register is automatically loaded when a Table Indirect Select or Reselect SCRIPTS instruction is executed.

**R** **Reserved** **7**

**SCF[2:0]** **Synchronous Clock Conversion Factor** **[6:4]**
These bits select a factor by which the frequency of SCLK is divided before being presented to the synchronous SCSI control logic. The synchronous transfer speed is determined by the combination of the divided clock and the setting of the XCLKS and XCLKH bits in the SCSI Control Four (SCNTL4) register. The table below shows the clock dividers that are available. See the table in the SCSI Control Four (SCNTL4) register description for a full list of available transfer rates.

| SCF2 | SCF1 | SCF0 | Factor Frequency |
|---|---|---|---|
| 0 | 0 | 0 | SCLK/3 |
| 0 | 0 | 1 | SCLK/1 |
| 0 | 1 | 0 | SCLK/1.5 |
| 0 | 1 | 1 | SCLK/2 |
| 1 | 0 | 0 | SCLK/3 |
| 1 | 0 | 1 | SCLK/4 |
| 1 | 1 | 0 | SCLK/6 |
| 1 | 1 | 1 | SCLK/8 |

**EWS** **Enable Wide SCSI** **3**
When this bit is cleared, all information transfer phases are assumed to be eight bits, transmitted on SD[7:0]/ and SDP0/. When this bit is asserted, data transfers are performed 16 bits at a time; the least significant byte is

on SD[7:0]/ and SDP0/, and the most significant byte is on SD[15:8]/, SDP1/. Command, Status, and Message phases are not affected by this bit. Because Ultra3 DT SCSI transfers are always wide this bit must be set. If it is not set, a SGE interrupt will occur.

**R** **Reserved** **[2:0]**

## Register: 0x04

**SCSI Chip ID (SCID)**
**Read/Write**

| 7 | 6 | 5 | 4 | 3 | | | 0 |
|---|---|---|---|---|---|---|---|
| R | RRE | SRE | R | ENC[3:0] | | | |
| x | 0 | 0 | x | 0 | 0 | 0 | 0 |

**R** **Reserved** **7**

**RRE** **Enable Response to Reselection** **6**
When this bit is set, the SYM53C1010 SCSI function is enabled to respond to bus-initiated reselection at the chip ID in the Response ID Zero (RESPID0) and Response ID One (RESPID1) registers. Note that the chip does not automatically reconfigure itself to the initiator mode as a result of being reselected.

**SRE** **Enable Response to Selection** **5**
When this bit is set, the SYM53C1010 SCSI function is able to respond to bus-initiated selection at the chip ID in the Response ID Zero (RESPID0) and Response ID One (RESPID1) registers. Note that the chip does not automatically reconfigure itself to target mode as a result of being selected.

**R** **Reserved** **4**

**ENC[3:0]** **Encoded Chip SCSI ID** **[3:0]**
These bits are used to store the SYM53C1010 SCSI function encoded SCSI ID. This is the ID which the chip asserts when arbitrating for the SCSI bus. The IDs that the SYM53C1010 SCSI function responds to when selected or reselected are configured in the Response ID Zero (RESPID0) and Response ID One (RESPID1)

registers. The priority of the 16 possible IDs, in descending order is:

| Highest | | | | | | | | Lowest | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |

## Register: 0x05
### SCSI Transfer (SXFER)
### Read/Write

| 7 | 6 | 5 | | | | | 0 |
|---|---|---|---|---|---|---|---|
| R | | MO[5:0] | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

This register is automatically loaded when a Table Indirect Select or Reselect SCRIPTS instruction is executed.

**R** **Reserved** **[7:6]**

**MO[5:0]** **Max SCSI Synchronous Offset** **[5:0]**
These bits describe the maximum SCSI synchronous offset used by the SYM53C1010 SCSI function when transferring synchronous SCSI data in either the initiator or target mode. Table 4.3 describes the possible combinations and their relationship to the synchronous data offset used by the SYM53C1010 SCSI function. These bits determine the SYM53C1010 SCSI function's method of transfer for ST/DT Data-In and ST/DT Data-Out phases only; all other information transfers occur asynchronously. Please note that the SCSI offset for Ultra3 transfers is counted as the maximum number of data transfers allowed to be outstanding, not the maximum REQ pulses allowed to be outstanding.

During ST Data-In or ST Data-Out transfers the maximum supported offset is 31 (MO[5:0] = 0x1F).

During DT Data-In or DT Data-Out transfers the maximum supported offset is 62 (MO[5:0] = 0x3E).

Setting offset values outside the allowable range will result in data corruption.

A value of 0 in these bits program the device to perform asynchronous transfers. A value of 1 during DT transfers is illegal and will result in data corruption.

**Table 4.3 Maximum Synchronous Offset**

| MO5 | MO4 | MO3 | MO2 | MO1 | MO0 | Synchronous Offset |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0-Asynchronous |
| 0 | 0 | 0 | 0 | 0 | 1 | Reserved |
| 0 | 0 | 0 | 0 | 1 | 0 | 2 |
| 0 | 0 | 0 | 0 | 1 | 1 | 3 |
| ... | ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... | ... |
| 1 | 1 | 1 | 0 | 0 | 1 | 57 |
| 1 | 1 | 1 | 0 | 1 | 0 | 58 |
| 1 | 1 | 1 | 0 | 1 | 1 | 59 |
| 1 | 1 | 1 | 1 | 0 | 0 | 60 |
| 1 | 1 | 1 | 1 | 0 | 1 | 61 |
| 1 | 1 | 1 | 1 | 1 | 0 | 62 |
| 1 | 1 | 1 | 1 | 1 | 1 | Reserved |

## Register: 0x06

### SCSI Destination ID (SDID)

### Read/Write

| 7 | | | 4 | 3 | | | 0 |
|---|---|---|---|---|---|---|---|
| R | | | | ENC | | | |
| x | x | x | x | 0 | 0 | 0 | 0 |

**R** **Reserved** **[7:4]**

**ENC** **Encoded Destination SCSI ID** **[3:0]**

Writing these bits sets the SCSI ID of the intended initiator or target during SCSI reselection or selection phases, respectively. When executing SCRIPTS, the SCRIPTS processor writes the destination SCSI ID to this register. The SCSI ID is defined by the user in a SCRIPTS Select or Reselect instruction. The value written is the binary-encoded ID. The priority of the 16 possible IDs, in descending order, is:

| Highest | | | | | | | | Lowest | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |

## Register: 0x07
### General Purpose (GPREG)
### Read/Write

| 7 | | | 5 | 4 | | | 0 |
|---|---|---|---|---|---|---|---|
| R | | | | GPIO | | | |
| x | x | x | | 0 | x | x | x | x |

A write to this register will cause the data written to be output to the appropriate GPIO pin if it is set to output mode in that function's General Purpose Pin Control (GPCNTL) register.

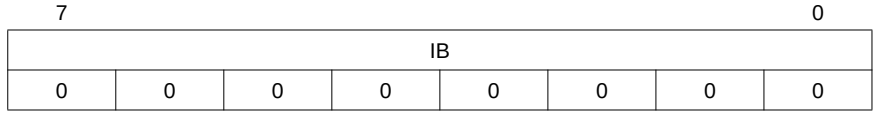
**R** **Reserved** **[7:5]**

**GPIO** **General Purpose I/O** **[4:0]**
These bits are programmed through the General Purpose Pin Control (GPCNTL) register as inputs, outputs, or to perform special functions. As an output, these pins can be used to enable or disable external terminators. It is also possible to program these signals as live inputs and sense them through a SCRIPTS register to register Move instruction. GPIO[3:0] default as inputs and GPIO4 defaults as an output pin. When configured as inputs, an internal pull-up is enabled.

LSI Logic Symbios software uses the GPIO[1:0] signals to access serial EEPROM. GPIO1 is used as a clock, with the GPIO0 pin serving as data.

LSI Logic Symbios software also reserves the use of GPIO[4:2]. If there is a need to use GPIO[4:2], please check with LSI Logic for additional information.

## Register: 0x08
### SCSI First Byte Received (SFBR)
### Read/Write




**SFBR** **SCSI First Byte Received** **[7:0]**

This register contains the first byte received in any asynchronous information transfer phase. For example, when a SYM53C1010 SCSI function is operating in the initiator mode, this register contains the first byte received in the Message-In, Status, and Data-In phases.

When a Block Move instruction is executed for a particular phase, the first byte received is stored in this register – even if the present phase is the same as the last phase. The first byte received value for a particular input phase is not valid until after a MOVE instruction is executed.

This register is also the accumulator for register read-modify-writes with the SCSI First Byte Received (SFBR) as the destination. This allows bit testing after an operation.

The SCSI First Byte Received (SFBR) is not writable using the CPU, and therefore not by a Memory Move. However, it can be loaded using SCRIPTS Read/Write operations. To load the SFBR with a byte stored in system memory, the byte must first be moved to an intermediate SYM53C1010 SCSI function register (such as the SCRATCH register), and then to the SFBR.

This register also contains the state of the lower eight bits of the SCSI data bus during the Selection phase if the COM bit in the DMA Control (DCNTL) register is clear.

If the COM bit is cleared, do not access this register using SCRIPTS operations, as indeterminate operations may occur. (This includes SCRIPTS Read/Write operations and conditional transfer control instructions that initialize the SCSI First Byte Received (SFBR) register.)

## Register: 0x09

**SCSI Output Control Latch (SOCL)**
**Read/Write**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| REQ | ACK | BSY | SEL | ATN | MSG | C_D | I_O |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

This register is used primarily for diagnostics testing or programmed I/O operation. It is controlled by the SCRIPTS processor when executing SCSI SCRIPTS. SCSI Output Control Latch (SOCL) is used only when transferring data using programmed I/O. Some bits are set or cleared when executing SCSI SCRIPTS. Do not write to the register once the SYM53C1010 SCSI function starts executing normal SCSI SCRIPTS.

| | | |
|---|---|---|
| **REQ** | **Assert SCSI REQ/ Signal** | **7** |
| **ACK** | **Assert SCSI ACK/ Signal** | **6** |
| **BSY** | **Assert SCSI BSY/ Signal** | **5** |
| **SEL** | **Assert SCSI SEL/ Signal** | **4** |
| **ATN** | **Assert SCSI ATN/ Signal** | **3** |
| **MSG** | **Assert SCSI MSG/ Signal** | **2** |
| **C_D** | **Assert SCSI C_D/ Signal** | **1** |
| **I_O** | **Assert SCSI I_O/ Signal** | **0** |

## Register: 0x0A

**SCSI Selector ID (SSID)**

**Read Only**

| 7 | 6 | | 4 | 3 | | | 0 |
|---|---|---|---|---|---|---|---|
| VAL | R | | | ENID | | | |
| 0 | x | x | x | 0 | 0 | 0 | 0 |

**VAL** **SCSI Valid** **7**

If VAL is asserted, then the two SCSI IDs are detected on the bus during a bus-initiated selection or reselection, and the encoded destination SCSI ID bits below are valid. If VAL is deasserted, only one ID is present and the contents of the encoded destination ID are meaningless.

**R** **Reserved** **[6:4]**

**ENID** **Encoded Destination SCSI ID** **[3:0]**

Reading the SSID register immediately after the SYM53C1010 SCSI function is selected or reselected returns the binary-encoded SCSI ID of the device that performed the operation. These bits are invalid for targets that are selected under the single initiator option of the SCSI-1 specification. This condition is detected by examining the VAL bit above.

## Register: 0x0B
**SCSI Bus Control Lines (SBCL)**
**Read Only**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| REQ | ACK | BSY | SEL | ATN | MSG | C_D | I_O |
| x | x | x | x | x | x | x | x |

This register returns the SCSI control line status. A bit is set when the corresponding SCSI control line is asserted. These bits are not latched; they are a true representation of what is on the SCSI bus at the time the register is read. The resulting read data is synchronized before being presented to the PCI bus to prevent parity errors from being passed to the system. This register is used for diagnostics testing or operation in the low level mode.

| | | |
|---|---|---|
| **REQ** | **Assert SCSI REQ/ Signal** | **7** |
| **ACK** | **Assert SCSI ACK/ Signal** | **6** |
| **BSY** | **Assert SCSI BSY/ Signal** | **5** |
| **SEL** | **Assert SCSI SEL/ Signal** | **4** |
| **ATN** | **Assert SCSI ATN/ Signal** | **3** |
| **MSG** | **Assert SCSI MSG/ Signal** | **2** |
| **C_D** | **Assert SCSI C_D/ Signal** | **1** |
| **I_O** | **Assert SCSI I_O/ Signal** | **0** |

## Register: 0x0C
**DMA Status (DSTAT)**
**Read Only**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DFE | MDPE | BF | ABRT | SSI | SIR | R | IID |
| 1 | 0 | 0 | 0 | 0 | 0 | x | 0 |

Reading this register clears any bits that are set at the time the register is read, but does not necessarily clear the register in case additional interrupts are pending (the SYM53C1010 SCSI functions stack

interrupts). The DIP bit in the Interrupt Status Zero (ISTAT0) register is also cleared. It is possible to mask DMA interrupt conditions individually through the DMA Interrupt Enable (DIEN) register.

When performing consecutive 8-bit reads of the DMA Status (DSTAT), SCSI Interrupt Status Zero (SIST0) and SCSI Interrupt Status One (SIST1) registers (in any order), insert a delay equivalent to 12 CLK periods between the reads to ensure that the interrupts clear properly. See Chapter 2, "Functional Description" for more information on interrupts.

**DFE** **DMA FIFO Empty** **7**
This status bit is set when the DMA FIFO is empty. It is possible to use it to determine if any data resides in the FIFO when an error occurs and an interrupt is generated. This bit is a pure status bit and does not cause an interrupt.

**MDPE** **Master Data Parity Error** **6**
This bit is set when the SYM53C1010 SCSI function, acting as a PCI master, detects a data parity error, or, acting as a target device, signals a parity error during a data phase. This bit is completely disabled by the Master Parity Error Enable bit (bit 3 of Chip Test Four (CTEST4)).

**BF** **Bus Fault** **5**
This bit is set when a PCI bus fault condition is detected. A PCI bus fault can only occur when the SYM53C1010 SCSI function is bus master, and is defined as a cycle that ends with a Bad Address or Target Abort Condition.

**ABRT** **Aborted** **4**
This bit is set when an abort condition occurs. An abort condition occurs when a software abort command is issued by setting bit 7 of the Interrupt Status Zero (ISTAT0) register.

**SSI** **Single Step Interrupt** **3**
If the Single-Step Mode bit in the DMA Control (DCNTL) register is set, this bit is set and an interrupt generated after successful execution of each SCRIPTS instruction.

**SIR** **SCRIPTS Interrupt Instruction Received** **2**
This status bit is set whenever an interrupt instruction is evaluated as true.

| | | |
|---|---|---|
| **R** | **Reserved** | **1** |
| **IID** | **Illegal Instruction Detected**<br>This status bit is set any time an illegal or reserved instruction opcode is detected, whether the SYM53C1010 SCSI function is operating in single-step mode or automatically executing SCSI SCRIPTS.<br><br>Any of the following conditions during instruction execution also sets this bit:<br><br>• The SYM53C1010 SCSI function is executing a Wait Disconnect instruction and the SCSI REQ line is asserted without a disconnect occurring.<br><br>• A Block Move instruction is executed as an initiator with 0x000000 loaded into the DMA Byte Counter (DBC) register, indicating there are zero bytes to move.<br><br>• During a Transfer Control instruction, the Compare Data (bit 18) and Compare Phase (bit 17) bits are set in the DMA Byte Counter (DBC) register while the SYM53C1010 SCSI function is in target mode.<br><br>• During a Transfer Control instruction, the Carry Test bit (bit 21) is set and either the Compare Data (bit 18) or Compare Phase (bit 17) bit is set.<br><br>• A Transfer Control instruction is executed with the reserved bit 22 set.<br><br>• A Transfer Control instruction is executed with the Wait for Valid phase bit (bit 16) set while the chip is in the target mode.<br><br>• A Load and Store instruction is issued with the memory address mapped to the operating registers of the chip, not including ROM or RAM.<br><br>• A Load and Store instruction is issued when the register address is not aligned with the memory address.<br><br>• A Load and Store instruction is issued with bit 5 in the DMA Command (DCMD) register cleared or bits 3 or 2 set.<br><br>• A Load and Store instruction when the count value in the DMA Byte Counter (DBC) register is not set at 1 to 4. | **0** |

- A Load and Store instruction attempts to cross a Dword boundary.
- A Memory Move instruction is executed with one of the reserved bits in the DMA Command (DCMD) register set.
- A Memory Move instruction is executed with the source and destination addresses not aligned.
- A 64-bit Table Indirect Block Move instruction is executed with a selector index value greater than 0x16.
- If the Select with ATN/ bit is set for any I/O instruction other than a select instruction.

## Register: 0x0D

**SCSI Status Zero (SSTAT0)**

**Read Only**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ILF | R | OLF | ARBIP | LOA | WOA | RST | SDP0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**ILF** — **SIDL Least Significant Byte Full** — **7**

This bit is set when the least significant byte in the SCSI Input Data Latch (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 SCSI Input Data Latch (SIDL) register contains SCSI data received asynchronously. Synchronous data received does not flow through this register.

**R** — **Reserved** — **6**

**OLF** — **SODL Least Significant Byte Full** — **5**

This bit is set when the least significant byte in the SCSI Output Data Latch (SODL) contains data. The SODL register is the interface between the DMA logic and the SCSI bus for asynchronous send operations. In the asynchronous mode, data is transferred from the host bus to the SODL register, and then to the SCSI bus. It is possible to use this bit to determine how many bytes reside in the device when an error occurs.

**ARBIP** **Arbitration in Progress** **4**

Arbitration in Progress (ARBIP = 1) indicates that the SYM53C1010 SCSI function has detected a Bus Free condition, asserted SBSY, and asserted its SCSI ID onto the SCSI bus.

**LOA** **Lost Arbitration** **3**

When set, LOA indicates that the SYM53C1010 SCSI function has detected a bus free condition, arbitrated for the SCSI bus, and lost arbitration due to another SCSI device asserting the SSEL/ signal.

**WOA** **Won Arbitration** **2**

When set, WOA indicates that the SYM53C1010 SCSI function has detected a Bus Free condition, arbitrated for the SCSI bus and won arbitration. The arbitration mode selected in the SCSI Control Zero (SCNTL0) register must be full arbitration and selection to set this bit.

**RST** **SCSI RST/ Signal** **1**

This bit reports the current status of the SCSI RST/ signal, and the RST signal (bit 3) in the SCSI Control One (SCNTL1) register. This bit is not latched and may change as it is read.

**SDP0** **SCSI SDP0 Parity Signal** **0**

This bit represents the present state of the SCSI SDP0/ parity signal. This signal is not latched and may change as it is read.

## Register: 0x0E

**SCSI Status One (SSTAT1)**

**Read Only**

| 7 | | | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| R | | | | SDP0L | MSG | C_D | I_O |
| 0 | 0 | 0 | 0 | x | x | x | x |

**R** **Reserved** **[7:4]**

**SDP0L** **Latched SCSI Parity** **3**

This bit reflects the SCSI parity signal (SDP0/), corresponding to the data latched in the SCSI Input Data Latch (SIDL). It changes when a new byte is latched into the least significant byte of the SIDL register. This bit is active high, in other words, it is set when the parity signal is active.

**MSG** **SCSI MSG/ Signal** **2**

This SCSI phase status bit is latched on the asserting edge of SREQ/ when operating in either the initiator or target mode. This bit is set when the corresponding signal is active. It is useful when operating in the low level mode.

**C_D** **SCSI C_D/ Signal** **1**

This SCSI phase status bit is latched on the asserting edge of SREQ/ when operating in either the initiator or target mode. This bit is set when the corresponding signal is active. It is useful when operating in the low level mode.

**I_O** **SCSI I_O/ Signal** **0**

This SCSI phase status bit is latched on the asserting edge of SREQ/ when operating in either the initiator or target mode. This bit is set when the corresponding signal is active. It is useful when operating in the low level mode.

## Register: 0x0F
### SCSI Status Two (SSTAT2)
### Read Only

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ILF1 | R | OLF1 | R | SPL1 | R | LDSC | SDP1 |
| 0 | 0 | 0 | 0 | x | 0 | 1 | x |

**ILF1** — **SIDL Most Significant Byte Full** — **7**
This bit is set when the most significant byte in the SCSI Input Data Latch (SIDL) contains data. Data is transferred from the SCSI bus to the SCSI Input Data Latch register before being sent to the DMA FIFO. The data is then sent to the host bus. The SIDL register contains SCSI data received asynchronously. Synchronous data received does not flow through this register.

**R** — **Reserved** — **6**

**OLF1** — **SODL Most Significant Byte Full** — **5**
This bit is set when the most significant byte in the SCSI Output Data Latch (SODL) contains data. The SODL register is the interface between the DMA logic and the SCSI bus for asynchronous send operations. In the asynchronous mode, data is transferred from the host bus to the SCSI Output Data Latch (SODL) register, and then to the SCSI bus. This bit can be used to determine how many bytes reside in the device when an error occurs.

**R** — **Reserved** — **4**

**SPL1** — **Latched SCSI parity for SD[15:8]** — **3**
This active HIGH bit reflects the SCSI odd parity signal corresponding to the data latched into the most significant byte in the SCSI Input Data Latch (SIDL) register.

**R** — **Reserved** — **2**

**LDSC** — **Last Disconnect** — **1**
This bit is used in conjunction with the Connected (CON) bit in SCSI Control One (SCNTL1). It allows the user to detect the case in which a target device disconnects, and then a SCSI device selects or reselects the

SYM53C1010 SCSI function. If the Connected bit and the LDSC bit are asserted, a disconnect is indicated. This bit is set when the Connected bit in SCNTL1 is off. This bit is cleared when a Block Move instruction is executed while the Connected bit in SCNTL1 is set.

**SDP1** **SCSI SDP1 Parity Signal** **0**

This bit represents the present state of the SCSI SDP1/ parity signal. It is not latched and may change as it is read.

## Registers: 0x10–0x13

**Data Structure Address (DSA)**
**Read/Write**

| 31 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DSA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**DSA** **Data Structure Address** **[31:0]**

This 32-bit register contains the base address used for all Table Indirect calculations. The DSA register is usually loaded prior to starting an I/O, but it is possible for a SCRIPTS Memory Move to load the DSA during the I/O.

During any Memory-to-Memory Move operation, the contents of this register are preserved. The power-up value of this register is indeterminate.

## Register: 0x14

**Interrupt Status Zero (ISTAT0)**
**Read/Write**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ABRT | SRST | SIGP | SEM | CON | INTF | SIP | DIP |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

This is the only register that is accessible by the host CPU while a SYM53C1010 SCSI function is executing SCRIPTS (without interfering in the operation of the function). It is used to poll for interrupts if hardware interrupts are disabled. Read this register after servicing an interrupt to check for stacked interrupts.

**ABRT** **Abort Operation** **7**

Setting this bit aborts the current operation under execution by the SYM53C1010 SCSI function. If this bit is set and an interrupt is received, clear this bit before reading the DMA Status (DSTAT) register to prevent further aborted interrupts from being generated. The sequence to abort any operation is:

1.Set this bit.

2. Wait for an interrupt.

3. Read the Interrupt Status Zero (ISTAT0) register.

4. If the SCSI Interrupt Pending bit is set, read the SCSI Interrupt Status Zero (SIST0) or SCSI Interrupt Status One (SIST1) register to determine the cause of the SCSI Interrupt and return to Step 2.

5. If the SCSI Interrupt Pending bit is cleared and the DMA Interrupt Pending bit is set, write 0x00 to this register.

6. Read the DMA Status (DSTAT) register to verify the aborted interrupt and to determine if any other interrupting conditions have occurred.

**SRST** **Software Reset** **6**

Setting this bit resets the SYM53C1010 SCSI function. All operating registers are cleared to their respective default values and all SCSI signals are deasserted. Setting this bit does not assert the SCSI RST/ signal. This reset does not clear the ID Mode bit or any of the PCI configuration registers. This bit is not self-clearing; it must be cleared to clear the reset condition. A hardware reset also clears this bit.

Note: If SCRIPTS are running, then the ABRT bit (bit 7) must be set prior to setting the SRST bit.

**SIGP** **Signal Process** **5**

SIGP is a R/W bit that is writable at any time. It is polled and reset using Chip Test Two (CTEST2). The SIGP bit is used in various ways to pass a flag to or from a running SCRIPTS instruction.

The only SCRIPTS instruction directly affected by the SIGP bit is Wait for Selection/Reselection. Setting the

SIGP bit causes this instruction to jump to the alternate address immediately. The instructions at the alternate jump address should check the status of SIGP to determine the cause of the jump. The SIGP bit is usable at any time and is not restricted to the wait for selection/reselection condition.

**SEM** **Semaphore** **4**

The SCRIPTS processor may set this bit using a SCRIPTS register write instruction. An external processor may also set it while the SYM53C1010 SCSI function is executing a SCRIPTS operation. This bit enables the SCSI function to notify an external processor of a predefined condition while SCRIPTS are running. The external processor may also notify the SYM53C1010 SCSI function of a predefined condition and the SCRIPTS processor may take action while SCRIPTS are executing.

**CON** **Connected** **3**

This bit is automatically set any time the SYM53C1010 SCSI function is connected to the SCSI bus as an initiator or as a target. It is set after successfully completing selection or when the SYM53C1010 SCSI function responds to a bus-initiated selection or reselection. It is also set after the SCSI function wins arbitration when operating in the low level mode. When this bit is cleared, the SYM53C1010 SCSI function is not connected to the SCSI bus.

**INTF** **Interrupt-on-the-Fly** **2**

This bit is asserted by an INTFLY instruction during SCRIPTS execution. SCRIPTS programs do not halt when the interrupt occurs. This bit can be used to notify a service routine, running on the main processor while the SCRIPTS processor is still executing a SCRIPTS program. If this bit is set, when the Interrupt Status Zero (ISTAT0) register is read it is not automatically cleared. To clear this bit, write a one to it. The reset operation is self-clearing.

Note: If the INTF bit is set but SIP or DIP is not set, do not attempt to read the other chip status registers. An interrupt-

on-the-fly must be cleared before servicing any other interrupts indicated by SIP or DIP.

After it has been set, this bit must be written to one to clear it.

**SIP** **SCSI Interrupt Pending** **1**

This status bit is set when an interrupt condition is detected in the SCSI portion of the SYM53C1010 SCSI function. The following conditions cause a SCSI interrupt to occur:

- A phase mismatch (initiator mode) or SATN/ becomes active (target mode)
- An arbitration sequence completes
- A selection or reselection time-out occurs
- The SYM53C1010 SCSI function is selected
- The SYM53C1010 SCSI function is reselected
- A SCSI gross error occurs
- An unexpected disconnect occurs
- A SCSI reset occurs
- A parity error is detected
- The handshake-to-handshake timer expires
- The general purpose timer expires

To determine which condition(s) caused the interrupt, read the SCSI Interrupt Status Zero (SIST0) and SCSI Interrupt Status One (SIST1) registers.

**DIP** **DMA Interrupt Pending** **0**

This status bit is set when an interrupt condition is detected in the DMA portion of the SYM53C1010 SCSI function. The following conditions cause a DMA interrupt to occur:

- A PCI parity error is detected
- A bus fault is detected
- An abort condition is detected
- A SCRIPTS instruction is executed in the single-step mode
- A SCRIPTS interrupt instruction is executed

- An illegal instruction is detected

To determine exactly which condition(s) caused the interrupt, read the DMA Status (DSTAT) register.

## Register: 0x15

**Interrupt Status One (ISTAT1)**
**Read/Write**

| 7 | | | | | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | | | | | | FLSH | SRUN | SI |
| 0 | 0 | 0 | 0 | 0 | | 0 | 0 | 0 |

**R** **Reserved** **[7:3]**

**FLSH** **Flushing** **2**
If this bit is set, the chip is flushing data from the DMA FIFO. If this bit is cleared, no flushing is occurring. This bit is read only. Writes do not affect the value of this bit.

**SRUN** **SCRIPTS Running** **1**
If this bit is set, the SCRIPTS engine is currently fetching and executing SCRIPTS instructions. If this bit is cleared, the SCRIPTS engine is not active. This bit is read only. Writes do not affect the value of this bit.

**SI** **SYNC_IRQD** **0**
Setting this bit disables the INTA/ pin for Function A and the INTB/ pin for Function B. Clearing this bit enables normal operation of the INTA/ (or INTB/) pin. If the INTA/ (or INTB/) is already asserted and this bit is set, INT remains asserted until the interrupt is serviced. At this point the interrupt line is blocked for future interrupts until this bit is cleared. In addition, this bit may be read and written while SCRIPTS are executing.

## Register: 0x16
**Mailbox Zero (MBOX0)**
**Read/Write**

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| MBOX0 | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**MBOX0** **Mailbox Zero** **[7:0]**
These are general purpose bits that may be read or written while SCRIPTS are running. They also may be read or written by the SCRIPTS processor.

Note: The host and the SCRIPTS processor code could access the same mailbox byte at the same time. Using one mailbox register as read only and the other as write only prevents this conflict.

## Register: 0x17
**Mailbox One (MBOX1)**
**Read/Write**

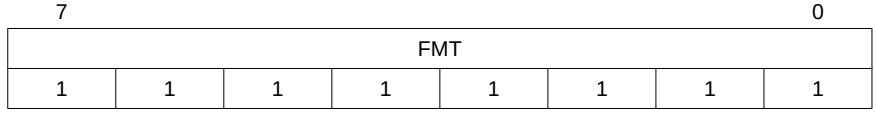
| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| MBOX1 | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**MBOX1** **Mailbox One** **[7:0]**
These are general purpose bits that may be read or written while SCRIPTS are running. They also may be read or written by the SCRIPTS processor.

Note: The host and the SCRIPTS processor code could access the same mailbox byte at the same time. Using one mailbox register as read only and the other as write only prevents this conflict.

## Register: 0x18

**Chip Test Zero (CTEST0)**
**Read/Write**



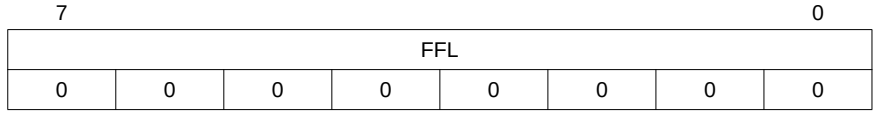
| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| FMT | | | | | | | |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**FMT** **Byte Empty in DMA FIFO** **[7:0]**

These bits identify the lower bytes in the DMA FIFO that are empty. Each bit corresponds to a byte lane in the DMA FIFO. For example, if byte lane three is empty, then FMT bit 3 is set. The FMT flags indicate the status of bytes at the bottom of the FIFO. Therefore, if all FMT bits are set, the DMA FIFO is empty.

## Register: 0x19

**Chip Test One (CTEST1)**
**Read Only**



| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| FFL | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**FFL** **Byte Full in DMA FIFO** **[7:0]**

These status bits identify the upper bytes in the DMA FIFO that are full. Each bit corresponds to a byte lane in the DMA FIFO. For example, if byte lane three is full then FFL bit 3 is set. The FFL flags indicate the status of bytes at the top of the FIFO. Therefore, if all FFL bits are set, the DMA FIFO is full.

## Register: 0x1A

**Chip Test Two (CTEST2)**

**Read Only (bit 3 write)**

| 7 | 6 | 5 | 4 | 3 | 2 | | 0 |
|---|---|---|---|---|---|---|---|
| R | SIGP | CIO | CM | PCICIE | R | | |
| x | 0 | x | x | 0 | x | x | x |

**R** **Reserved** **7**

**SIGP** **Signal Process** **6**

This bit is a copy of the SIGP bit in the Interrupt Status Zero (ISTAT0) register (bit 5). The SIGP bit is used to signal a running SCRIPTS instruction. When this register is read, the SIGP bit in the Interrupt Status Zero (ISTAT0) register is cleared.

**CIO** **Configured as I/O** **5**

This bit is defined as the Configuration I/O Enable Status bit. This read only bit indicates if the chip is currently enabled as I/O space.

**CM** **Configured as Memory** **4**

This bit is defined as the configuration memory enable status bit. This read only bit indicates if the chip is currently enabled as memory space.

Note: Bits 4 and 5 may be set if the chip is mapped in both I/O and memory space. Also, bits 4 and 5 may be set if the chip is dual-mapped.

**PCICIE** **PCI Configuration Info Enable** **3**

This bit controls the shadowing of the PCI Base Address Register One (BAR1) (MEMORY), PCI Base Address Register Two (BAR2) (MEMORY), PCI Base Address Register Three (BAR3) (SCRIPTS RAM), PCI Base Address Register Four (BAR4) (SCRIPTS RAM), PCI Device ID, and PCI Revision ID (RID) into the Scratch Register A (SCRATCHA), Memory Move Read Selector (MMRS), Scratch Register B (SCRATCHB), Memory Move Write Selector (MMWS), and SCRIPT Fetch Selector (SFS) registers.

When it is set, MMWS contains bits [63:32] and SCRATCH B contains bits [31:0] of the RAM Base

Address value from the PCI Configuration Base Address Register Three (BAR3) (SCRIPTS RAM) and Base Address Register Four (BAR4) (SCRIPTS RAM). This is the base address for the 8 Kbytes of internal RAM.

Memory Move Read Selector (MMRS) contains bits [63:32] and Scratch Register A (SCRATCHA) contains bits [31:0] of the memory mapped operating register base address. Bits [23:16] of SCRIPT Fetch Selector (SFS) contain the PCI Revision ID (RID) register value and bits [15:0] contain the PCI Device ID register value. When this bit is set, only reads to the registers are affected, writes occur normally.

When this bit is cleared, the SCRATCH A, MMRS, SCRATCH B, MMWS, and SFS registers return to normal operation.

Note: Bit 3 is the only writable bit in this register. All other bits are read only. When modifying this register, all other bits must be written to zero. Do not execute a Read-Modify-Write to this register.

**R** **Reserved** **[2:0]**

## Register: 0x1B

### Chip Test Three (CTEST3)

**Read/Write**

| 7 | | | | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | | | | | FLF | CLF | R | WRIE |
| 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |

**R** **Reserved** **[7:4]**

**FLF** **Flush DMA FIFO** **3**

When this bit is set, data residing in the DMA FIFO is transferred to memory, starting at the address in the DMA Next Address (DNAD) register. The internal DMAWR signal, controlled by the Chip Test Five (CTEST5) register, determines the direction of the transfer. This bit is not self-clearing; clear it once the data is successfully transferred by the SYM53C1010 SCSI function.

Note: Polling of FIFO flags is allowed during flush operations.

**CLF** **Clear DMA FIFO** **2**

When this bit is set, all data pointers for the DMA FIFO are cleared. Any data in the FIFO is lost. After the SYM53C1010 SCSI function successfully clears the appropriate FIFO pointers and registers, this bit automatically clears.

Note: This bit does not clear the data visible at the bottom of the FIFO.

**R** **Reserved** **1**

**WRIE** **Write and Invalidate Enable** **0**

This bit, when set, causes the issuing of Write and Invalidate commands on the PCI bus whenever legal. The Write and Invalidate Enable bit in the PCI Configuration Command register must also be set for the chip to generate Write and Invalidate commands.

## Registers: 0x1C–0x1F

**Temporary (TEMP)**

**Read/Write**

| 31 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TEMP | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**TEMP** **Temporary** **[31:0]**

This 32-bit register stores the Return instruction address pointer from the Call instruction. The address pointer stored in this register is loaded into the DMA SCRIPTS Pointer (DSP) register when a Return instruction is executed. This address points to the next instruction to execute. Do not write to this register while the SYM53C1010 SCSI function is executing SCRIPTS.

During any Memory-to-Memory Move operation, the contents of this register are preserved. The power-up value of this register is indeterminate.

## Register: 0x20
**Reserved**

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| R | | | | | | | |
| x | x | x | x | x | x | x | x |

This register is reserved.

## Register: 0x21
**Chip Test Four (CTEST4)**
**Read/Write**

| 7 | 6 | 5 | 4 | 3 | 2 | | 0 |
|---|---|---|---|---|---|---|---|
| R | FBL3 | R | SRTM | MPEE | FBL[2:0] | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**R** **Reserved** **7**

**FBL3** **FIFO Byte Control 3** **6**
This bit is used with FBL[2:0]. See Bits [2:0] description in this register.

**R** **Reserved** **5**

**SRTM** **Shadow Register Test Mode** **4**
Setting this bit allows access to the shadow registers used by Memory-to-Memory Move operations. When this bit is set, register accesses to the Temporary (TEMP) and Data Structure Address (DSA) registers are directed to the shadow copies STEMP (Shadow TEMP) and SDSA (Shadow DSA). The registers are shadowed to prevent them from being overwritten during a Memory-to-Memory Move operation. The Data Structure Address (DSA) and Temporary (TEMP) registers contain the base address used for Table Indirect calculations, and the address pointer for a call or return instruction, respectively. For more information, refer to Section 4.3, “SCSI Shadow Registers.”

**MPEE** **Master Parity Error Enable** **3**
Setting this bit enables parity checking during master data phases. A parity error during a bus master read is

detected by the SYM53C1010 SCSI function. A parity error during a bus master write is detected by the target, and the SYM53C1010 SCSI function is informed of the error by the PERR/ pin being asserted by the target. When this bit is cleared, the SYM53C1010 SCSI function does not interrupt if a master parity error occurs. This bit is cleared at power-up.

**FBL[2:0]** **FIFO Byte Control** **[2:0]**

| FBL3 | FBL2 | FBL1 | FBL0 | DMA FIFO Byte Lane | Pins |
|---|---|---|---|---|---|
| 0 | x | x | x | Disabled | N/A |
| 1 | 0 | 0 | 0 | 0 | D[7:0] |
| 1 | 0 | 0 | 1 | 1 | D[15:8] |
| 1 | 0 | 1 | 0 | 2 | D[23:16] |
| 1 | 0 | 1 | 1 | 3 | D[31:24] |
| 1 | 1 | 0 | 0 | 4 | D[39:32] |
| 1 | 1 | 0 | 1 | 5 | D[47:40] |
| 1 | 1 | 1 | 0 | 6 | D[53:48] |
| 1 | 1 | 1 | 1 | 7 | D[63:54] |

These bits steer the contents of the Chip Test Six (CTEST6) register to the appropriate byte lane of the 64-bit DMA FIFO. If the FBL3 bit is set, then FBL2 through FBL0 determine which of eight byte lanes can be read or written. When cleared, the byte lane read or written is determined by the current contents of the DMA Next Address (DNAD) and DMA Byte Counter (DBC) registers. Each of the eight bytes that make up the 64-bit DMA FIFO is accessed by writing these bits to the proper value. For normal operation, FBL3 must equal zero.

## Register: 0x22

**Chip Test Five (CTEST5)**

**Read/Write**

| 7 | 6 | 5 | | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ADCK | BBCK | R | | | BL2 | R | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**ADCK** **Clock Address Incrementor** **7**

Setting this bit increments the address pointer contained in the DMA Next Address (DNAD) register. The DNAD register is incremented based on the DNAD contents and the current DMA Byte Counter (DBC) value. This bit automatically clears itself after incrementing the DNAD register.

**BBCK** **Clock Byte Counter** **6**

Setting this bit decrements the byte count contained in the 24-bit DMA Byte Counter (DBC) register. It is decremented based on the DBC contents and the current DMA Next Address (DNAD) value. This bit automatically clears itself after decrementing the DBC register.

**R** **Reserved** **[5:3]**

**BL2** **Burst Length Bit 2** **2**

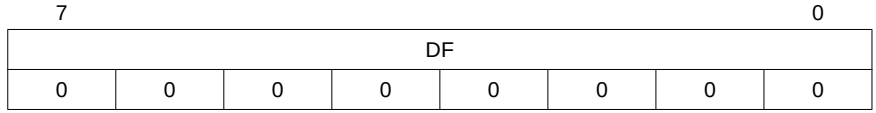
This bit works with bits 6 and 7 (BL[1:0]) in the DMA Mode (DMODE), 0x38 register to determine the burst length. For complete definitions of this field, refer to the descriptions of DMODE bits 6 and 7.

**R** **Reserved** **[1:0]**

## Register: 0x23

**Chip Test Six (CTEST6)**

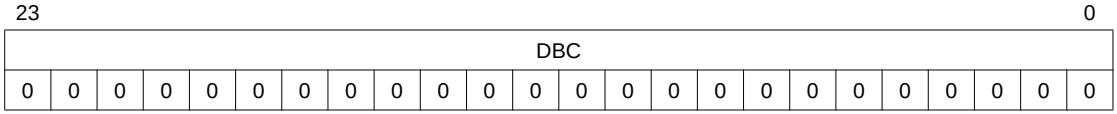
**Read/Write**




**DF** **DMA FIFO** **[7:0]**

Writing to this register writes data to the appropriate byte lane of the DMA FIFO, as determined by the FBL bits in the Chip Test Four (CTEST4) register. Reading this register unloads data from the appropriate byte lane of the DMA FIFO, as determined by the FBL bits in the CTEST4 register. Data written to the FIFO is loaded into the top of the FIFO. Data read out of the FIFO is taken from the bottom. To prevent DMA data from being corrupted, this register should not be accessed before starting or restarting SCRIPTS operations. This register should be the last register read when performing register dumps because of its effects on other registers. Write to this register only when testing the DMA FIFO using the CTEST4 register. Writing to this register while the test mode is not enabled produces unexpected results.

## Registers: 0x24–0x26
### DMA Byte Counter (DBC)
### Read/Write




**DBC** **DMA Byte Counter** **[23:0]**

This 24-bit register determines the number of bytes 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 SYM53C1010 SCSI function. The DBC counter is decremented each time data is transferred on the PCI bus. It is decremented by an amount equal to the number of bytes transferred.

The maximum number of bytes transferred in any one Block Move command is 16,777,215 bytes. The maximum value that can be loaded into the DMA Byte Counter (DBC) register is 0xFFFFFF. If the instruction is a Block Move and a value of 0x000000 is loaded into the DBC register, an illegal instruction interrupt occurs if the SYM53C1010 SCSI function is not in the target mode, Command phase.

The DMA Byte Counter (DBC) register is also used to hold the least significant 24-bits of the first Dword of a SCRIPTS fetch, and to hold the offset value during Table Indirect I/O SCRIPTS. For a complete description see Chapter 5, “SCSI SCRIPTS Instruction Set”. The power-up value of this register is indeterminate.

## Register: 0x27
### DMA Command (DCMD)
### Read/Write

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| DCMD | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**DCMD** **DMA Command** **[7:0]**
This 8-bit register determines the instruction for the SYM53C1010 SCSI function to execute. This register has a different format for each instruction. For a complete description see Chapter 5, “SCSI SCRIPTS Instruction Set”.

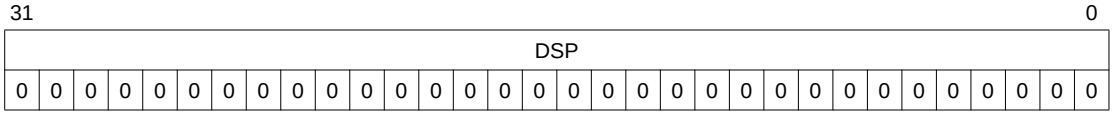
## Registers: 0x28–0x2B
### DMA Next Address (DNAD)
### Read/Write

| 31 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DNAD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**DNAD** **DMA Next Address** **[31:0]**
This 32-bit register contains the general purpose address pointer. At the start of some SCRIPTS operations, its value is copied from the DMA SCRIPTS Pointer Save (DSPS) register. Its value may not be valid except in certain abort conditions. The default value of this register is zero.

## Registers: 0x2C–0x2F

**DMA SCRIPTS Pointer (DSP)**
**Read/Write**



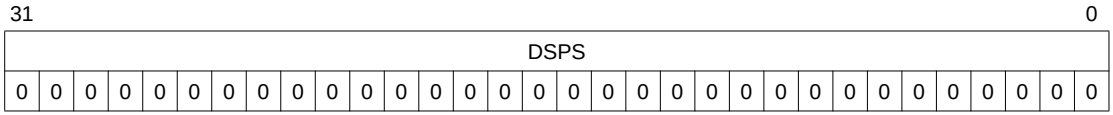

**DSP** — **DMA SCRIPTS Pointer** **[31:0]**

To execute SCSI SCRIPTS, the address of the first SCRIPTS instruction must be written to this register. In normal SCRIPTS operation, once the starting address of the SCRIPT is written to this register, SCRIPTS are automatically fetched and executed until an interrupt condition occurs.

In the single-step mode, there is a single step interrupt after each instruction is executed. The DMA SCRIPTS Pointer (DSP) register does not need to be written with the next address. However, to fetch and execute the next SCRIPTS command, the Start DMA bit (bit 2, DMA Control (DCNTL) register) must be set each time the step interrupt occurs. When writing this register eight bits at a time, writing the upper eight bits begins execution of SCSI SCRIPTS. The default value of this register is zero.

## Registers: 0x30–0x33

**DMA SCRIPTS Pointer Save (DSPS)**
**Read/Write**




**DSPS** — **DMA SCRIPTS Pointer Save** **[31:0]**

This register contains the second Dword of a SCRIPTS instruction. It is overwritten each time a SCRIPTS instruction is fetched. When a SCRIPTS interrupt instruction is executed, this register holds the interrupt vector. The power-up value of this register is indeterminate.

## Registers: 0x34–0x37
### Scratch Register A (SCRATCHA)
### Read/Write

| 31 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SCRATCHA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**SCRATCHA** **Scratch Register A** **[31:0]**

This is a general purpose, user-definable scratch pad register. Apart from CPU access, only register read/write and Memory Moves into the SCRATCH register alter its contents. The power-up value of this register is indeterminate.

A special mode of this register is enabled by setting the PCI Configuration Info Enable bit in the Chip Test Two (CTEST2) register. If this bit is set, bits [31:10] of Scratch Register A (SCRATCHA) return bits [31:10] of the PCI Base Address Register One (BAR1) (MEMORY). Bits [9:0] of SCRATCH A will always return zero in this mode. Writes to the SCRATCHA register are unaffected. Clearing the PCI Configuration Info Enable bit causes the SCRATCH A register to return to normal operation.

## Register: 0x38
### DMA Mode (DMODE)
### Read/Write

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| BL[1:0] | | SIOM | DIOM | ERL | ERMP | BOF | MAN |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**BL[1:0]** **Burst Length** **[7:6]**

These bits control the maximum number of Dwords transferred per bus ownership, regardless of whether the transfers are back-to-back, burst, or a combination of both. This value is also independent of the width (64-bit or 32-bit) of the data transfer on the PCI bus. The SYM53C1010 SCSI function asserts the Bus Request (PCIREQ/) output when the DMA FIFO can accommodate a transfer of at least one burst threshold of

data. Bus Request (PCIREQ/) is also asserted during start-of-transfer and end-of-transfer cleanup and alignment, even if less than a full burst of transfers is performed. The SYM53C1010 SCSI function inserts a “fairness delay” of four CLKs between burst transfers (set in BL[2:0]) during normal operation. The fairness delay is not inserted during PCI retry cycles. This gives the CPU and other bus master devices the opportunity to access the PCI bus between bursts.

| BL2 (CTEST5 bit 2) | BL1 | BL0 | Number of 64-bit Transfers | Number of 32-bit Transfers |
|---|---|---|---|---|
| 0 | 0 | 0 | 2 | 4 |
| 0 | 0 | 1 | 4 | 8 |
| 0 | 1 | 0 | 8 | 16 |
| 0 | 1 | 1 | 16 | 32 |
| 1 | 0 | 0 | 32 | 64 |
| 1 | 0 | 1 | 64 | 128 |
| 1 | 1 | 0 | 64 | 128 |
| 1 | 1 | 1 | Reserved | Reserved |

**SIOM** **Source I/O-Memory Enable** **5**

This bit is defined as an I/O Memory Enable bit for the source address of a Memory Move or Block Move Command. If this bit is set, then the source address is in I/O space; if cleared, the source address is in memory space.

This function is useful for register-to-memory operations using the Memory Move instruction when a SYM53C1010 SCSI function is I/O mapped. Bits 4 and 5 of the Chip Test Two (CTEST2) register are used to determine the configuration status of the SYM53C1010 SCSI function.

**DIOM** **Destination I/O-Memory Enable** **4**

This bit is defined as an I/O Memory Enable bit for the destination address of a Memory Move or Block Move Command. If this bit is set, then the destination address is in I/O space; if cleared, the destination address is in memory space.

This function is useful for memory-to-register operations using the Memory Move instruction when a SYM53C1010 SCSI function is I/O mapped. Bits 4 and 5

of the Chip Test Two (CTEST2) register are used to determine the configuration status of the SYM53C1010 SCSI function.

**ERL** **Enable Read Line** **3**
This bit enables a PCI Read Line command. If this bit is set and the chip is about to execute a read cycle (other than an opcode fetch), the command is 0b1110.

**ERMP** **Enable Read Multiple** **2**
If this bit is set and cache mode is enabled, a Read Multiple command is used on all read cycles when it is legal.

**BOF** **Burst Opcode Fetch Enable** **1**
Setting this bit causes the SYM53C1010 SCSI function to fetch instructions in burst mode. Specifically, the chip bursts in the first two Dwords of all instructions using a single bus ownership. If the instruction is a Memory-to-Memory Move type, the third Dword is accessed in a subsequent bus ownership. If the instruction is an Indirect type, the additional Dword is accessed in a subsequent bus ownership. If the instruction is a Table Indirect Block Move type, the chip accesses the remaining two Dwords in a subsequent bus ownership, thereby fetching the four Dwords required in two bursts of two Dwords each. If prefetch is enabled, this bit has no affect. This bit also has no affect on fetches out of SCRIPTS RAM.

**MAN** **Manual Start Mode** **0**
Setting this bit prevents the SYM53C1010 SCSI function from automatically fetching and executing SCSI SCRIPTS when the DMA SCRIPTS Pointer (DSP) register is written. When this bit is set, the Start DMA bit in the DMA Control (DCNTL) register must be set to begin SCRIPTS execution. Clearing this bit causes the SYM53C1010 SCSI function to automatically begin fetching and executing SCSI SCRIPTS when the DMA SCRIPTS Pointer (DSP) register is written. This bit normally is not used for SCSI SCRIPTS operations.

## Register: 0x39

**DMA Interrupt Enable (DIEN)**
**Read/Write**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| R | MDPE | BF | ABRT | SSI | SIR | R | IID |
| x | 0 | 0 | 0 | 0 | 0 | x | 0 |

This register contains the interrupt mask bits corresponding to the interrupting conditions described in the DMA Status (DSTAT) register. An interrupt is masked by clearing the appropriate mask bit. Masking an interrupt prevents INTA/ (for Function A) or INTB/ (for Function B) from being asserted for the corresponding interrupt, but the status bit is still set in the DMA Status (DSTAT) register. Masking an interrupt does not prevent setting the Interrupt Status Zero (ISTAT0) DIP. All DMA interrupts are considered fatal. Therefore, SCRIPTS halts when this condition occurs, whether or not the interrupt is masked. Setting a mask bit enables the assertion of INTA/, or INTB/, for the corresponding interrupt. A masked nonfatal interrupt does not prevent unmasked or fatal interrupts from getting through; interrupt stacking begins when either the Interrupt Status Zero (ISTAT0) SIP or DIP bit is set.

The INTA/ and INTB/ outputs are latched. Once asserted, they remain asserted until the interrupt is cleared by reading the appropriate status register. Masking an interrupt after the INTA/, or INTB/, output is asserted does not cause deassertion of INTA/ or INTB/. For more information on interrupts, see Chapter 2, “Functional Description”.

| | | |
|---|---|---|
| **R** | **Reserved** | **7** |
| **MDPE** | **Master Data Parity Error** | **6** |
| **BF** | **Bus Fault** | **5** |
| **ABRT** | **Aborted** | **4** |
| **SSI** | **Single Step Interrupt** | **3** |
| **SIR** | **SCRIPTS Interrupt Instruction Received** | **2** |
| **R** | **Reserved** | **1** |
| **IID** | **Illegal Instruction Detected** | **0** |

## Register: 0x3A

**Scratch Byte Register (SBR)**
**Read/Write**

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| SBR | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**SBR** **Scratch Byte Register** **[7:0]**
This is a general purpose register. Apart from CPU access, only register Read/Write and Memory Moves into this register alter its contents. The default value of this register is zero. This register is called the DMA Watchdog Timer on previous SYM53C8XX family products.

## Register: 0x3B

**DMA Control (DCNTL)**
**Read/Write**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CLSE | PFF | PFEN | SSM | IRQM | STD | R | COM |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**CLSE** **Cache Line Size Enable** **7**
Setting this bit enables the SYM53C1010 SCSI function to sense and react to cache line boundaries set up by the DMA Mode (DMODE) or PCI Cache Line Size (CLS) register, whichever contains the smaller value. Clearing this bit disables the cache line size logic.

**PFF** **Prefetch Flush** **6**
Setting this bit causes the prefetch unit to flush its contents. This bit clears after the flush is complete.

**PFEN** **Prefetch Enable** **5**
Setting this bit enables an 8-Dword SCRIPTS instruction prefetch unit. The prefetch unit, when enabled, fetches 8 Dwords of instructions and instruction operands in bursts of 4 or 8 Dwords. Prefetching instructions allows the SYM53C1010 SCSI function to make more efficient use of the system PCI bus, thus improving overall system performance. A flush occurs whenever the PFF bit is set, on all transfer control instructions (when the transfer

conditions are met), on writes to the DMA SCRIPTS Pointer (DSP), on regular MMOV instructions, and when an interrupt is generated. Based on the burst length as determined by the values in the DMA Mode (DMODE) register, the unit automatically determines the maximum burst size that it is capable of performing. If the burst threshold is set to 8 Dwords, the prefetch unit fetches instructions in two bursts of 4 Dwords. If the burst threshold is set to 16 Dwords or greater, the prefetch unit fetches instructions in one burst of 8 Dwords. Burst thresholds of less than 8 Dwords cause the prefetch unit to be disabled. PCI Cache commands (Read Line and Read Multiple) are issued if PCI caching is enabled. Prefetching from SCRIPTS RAM is not supported and is unnecessary due to the speed of the fetches. When fetching from SCRIPTS RAM, the setting of this bit has no effect on the fetch mechanism from SCRIPTS RAM. The prefetch unit does not support 64-bit data instruction fetches across the PCI bus. Prefetches of SCRIPTS instructions are 32-bits in width.

**SSM** **Single-Step Mode** **4**

Setting this bit causes the SYM53C1010 SCSI function to stop after executing each SCRIPTS instruction and to generate a single step interrupt. When this bit is cleared the SYM53C1010 SCSI function does not stop after each instruction. It continues fetching and executing instructions until an interrupt condition occurs. For normal SCSI SCRIPTS operation, keep this bit cleared. To restart the SYM53C1010 SCSI function after it generates a SCRIPTS Step interrupt, read the Interrupt Status Zero (ISTAT0) and DMA Status (DSTAT) registers to recognize and clear the interrupt. Then set the START DMA bit in this register.

**IRQM** **IRQ Mode** **3**

When set, this bit enables a totem pole driver for the INTA/, or INTB/ pin. When cleared, this bit enables an open drain driver for the INTA/, or INTB/, pin with an internal weak pull-up. The bit should remain cleared to retain full PCI compliance.

**STD** **Start DMA Operation** **2**

The SYM53C1010 SCSI function fetches a SCSI SCRIPTS instruction from the address contained in the

DMA SCRIPTS Pointer (DSP) register when this bit is set. This bit is required if the SYM53C1010 SCSI function is in one of the following modes:

- Manual start mode – Bit 0 in the DMA Mode (DMODE) register is set
- Single-step mode – Bit 4 in the DMA Control (DCNTL) register is set

When the SYM53C1010 SCSI function is executing SCRIPTS in manual start mode, the Start DMA bit must be set to start instruction fetches, but need not be set again until an interrupt occurs. When the SYM53C1010 SCSI function is in the single-step mode, set the Start DMA bit to restart execution of SCRIPTS after a single-step interrupt.

**R** **Reserved** **1**

**COM** **SYM53C700 Compatibility** **0**

When the COM bit is cleared, the SYM53C1010 SCSI function behaves in a manner compatible with the SYM53C700; selection/reselection IDs are stored in both the SCSI Selector ID (SSID) and SCSI First Byte Received (SFBR) registers. This bit is not affected by a software reset.

If the COM bit is cleared, do not access this register using SCRIPTS operation as indeterminate operations may occur. This includes SCRIPTS read/write operations and conditional transfer control instructions that initialize the SCSI First Byte Received (SFBR) register.

When the COM bit is set, the ID is stored only in the SCSI Selector ID (SSID) register, protecting the SCSI First Byte Received (SFBR) from being overwritten if a selection/reselection occurs during a DMA register-to-register operation.

## Registers: 0x3C–0x3F
### Adder Sum Output (ADDER)
### Read Only

| 31 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDER | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**ADDER** **Adder Sum Output** **[31:0]**
This register contains the output of the internal adder, and is used primarily for test purposes. The power-up value for this register is indeterminate.

## Register: 0x40
### SCSI Interrupt Enable Zero (SIEN0)
### Read/Write

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| M/A | CMP | SEL | RSL | SGE | UDC | RST | PAR |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

This register contains the interrupt mask bits corresponding to the interrupting conditions described in the SCSI Interrupt Status Zero (SIST0) register. An interrupt is masked by clearing the appropriate mask bit. For more information on interrupts see Chapter 2, “Functional Description”.

**M/A** **SCSI Phase Mismatch - Initiator Mode; SCSI ATN Condition - Target Mode** **7**
In the initiator mode, this bit is set when the SCSI phase asserted by the target and sampled during SREQ/ does not match the expected phase in the SCSI Output Control Latch (SOCL) register. This expected phase is automatically written by SCSI SCRIPTS. In the target mode, this bit is set when the initiator asserts SATN/. See the Disable Halt on Parity Error or SATN/ Condition bit in the SCSI Control One (SCNTL1) register for more information on when this status is actually raised.

**CMP** **Function Complete** **6**
When set, this bit indicates the full arbitration and selection sequence is completed.

**SEL** **Selected** **5**

When set, this bit indicates the SYM53C1010 SCSI function is selected by a SCSI initiator device. For this to occur, set the Enable Response to Selection bit in the SCSI Chip ID (SCID) register.

**RSL** **Reselected** **4**

When set, this bit indicates the SYM53C1010 SCSI function is reselected by a SCSI target device. For this to occur, set the Enable Response to Reselection bit in the SCSI Chip ID (SCID) register.

**SGE** **SCSI Gross Error** **3**

The following conditions are considered SCSI Gross Errors:

- Offset Underflow occurs in target mode when a SACK/ signal is received before the corresponding SREQ/ signal has been sent.
- Offset Overflow occurs in initiator mode when an SREQ/ signal is received and causes the maximum offset, as defined by the MO[5:0] bits in the SCSI Transfer (SXFER) register, to be exceeded.
- In initiator mode, a phase change occurs with an outstanding SREQ/SACK offset.
- Residual Data in SCSI FIFO occurs when a transfer other than Synchronous Data Received is started with data left in the SCSI Synchronous Receive FIFO.
- Multiple CRC Requests occur when, during a synchronous DT transfer, multiple CRC requests are received within the same offset.
- A request for a Pad CRC word is received without the subsequent CRC word requests.
- A phase change occurs without a CRC Request.

Note: Checking for this condition can be disabled by setting the DISCRC bit in the CRC Control Zero (CRCCNTL0) register.

- An illegal Force CRC Request Block Move is executed.
- A SCRIPTS RAM parity error occurs.

Note: The Shadowed SCSI SGE Status 0 register indicates which condition caused an SCE SCSI interrupt. The register is shadowed behind the SIST registers. It can be accessed by setting bit 4, the Shadow Register Test Mode (STRM) bit, in the Chip Test Four (CTEST4) register.

**UDC** **Unexpected Disconnect** **2**

This condition only occurs in the initiator mode. It happens when the target, which the SYM53C1010 SCSI function is connected to, unexpectedly disconnects from the SCSI bus. See the SCSI Disconnect Unexpected bit in the SCSI Control Two (SCNTL2) register for more information on expected versus unexpected disconnects. Any disconnect in the low level mode causes this condition.

**RST** **SCSI Reset Condition** **1**

Indicates assertion of the SRST/ signal by the SYM53C1010 SCSI function or any other SCSI device. This condition is edge-triggered, so multiple interrupts cannot occur because of a single SRST/ pulse.

**PAR** **SCSI Parity/CRC/AIP Error** **0**

This bit indicates the SYM53C1010 SCSI function detected a Parity/CRC/AIP error while receiving or sending SCSI data. See the Disable Halt on Parity/CRC/AIP error or SATN/ Condition bits in the SCSI Control One (SCNTL1) register for more information about when this condition is raised.

## Register: 0x41

### SCSI Interrupt Enable One (SIEN1)

### Read/Write

| 7 | | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| R | | | SBMC | R | STO | GEN | HTH |
| x | x | x | 0 | x | 0 | 0 | 0 |

This register contains the interrupt mask bits corresponding to the interrupting conditions described in the SCSI Interrupt Status One (SIST1) register. An interrupt is masked by clearing the appropriate mask bit. For more information on interrupts refer to Chapter 2, “Functional Description”.

| | | |
|---|---|---|
| **R** | **Reserved** | **[7:5]** |
| **SBMC** | **SCSI Bus Mode Change**<br>Setting this bit allows the SYM53C1010 to generate an interrupt when the DIFFSENS pin detects a change in voltage level that indicates the SCSI bus has changed between SE, LVD, or HVD modes. For example, when this bit is cleared and the SCSI bus changes modes, IRQ/ does not assert and the SIP bit in the Interrupt Status Zero (ISTAT0) register is not set. However, bit 4 in the SCSI Interrupt Status One (SIST1) register is set. Setting this bit allows the interrupt to occur. | **4** |
| **R** | **Reserved** | **3** |
| **STO** | **Selection or Reselection Time-out**<br>This bit is set when the SCSI device which the SYM53C1010 SCSI function is attempting to select or reselect does not respond within the programmed time-out period. See the description of the SCSI Timer Zero (STIME0) register, bits [3:0], for more information on the time-out timer. | **2** |
| **GEN** | **General Purpose Timer Expired**<br>This bit is set when the general purpose timer is expired. The time measured is the time between enabling and disabling of the timer. See the description of the SCSI Timer One (STIME1) register, bits [3:0], for more information on the general purpose timer. | **1** |
| **HTH** | **Handshake-to-Handshake Timer Expired**<br>This bit is set when the handshake-to-handshake timer is expired. The time measured is the SCSI Request-to-Request (target) or Acknowledge-to-Acknowledge (initiator) period. See the description of the SCSI Timer Zero (STIME0) register, bits [7:4], for more information on the handshake-to-handshake timer. | **0** |

## Register: 0x42

### SCSI Interrupt Status Zero (SIST0)

### Read Only

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| M/A | CMP | SEL | RSL | SGE | UDC | RST | PAR |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Reading the SCSI Interrupt Status Zero (SIST0) register returns the status of the various interrupt conditions, whether they are enabled in the SCSI Interrupt Enable Zero (SIEN0) register or not. Each bit set indicates occurrence of the corresponding condition. Reading the SIST0 clears the interrupt status.

Reading this register clears any bits that are set at the time the register is read, but does not necessarily clear the register because additional interrupts may be pending; the SYM53C1010 SCSI functions stack interrupts. SCSI interrupt conditions are individually masked through the SCSI Interrupt Enable Zero (SIEN0) register.

When performing consecutive 8-bit reads of the DMA Status (DSTAT), SCSI Interrupt Status Zero (SIST0), and SCSI Interrupt Status One (SIST1) registers (in any order), insert a delay equivalent to 12 clock periods between the reads to ensure the interrupts clear properly. Also, if reading the registers when both the Interrupt Status Zero (ISTAT0) SIP and DIP bits may not be set, read the SIST0 and SIST1 registers before the DSTAT register to avoid missing a SCSI interrupt. For more information on interrupts refer to Chapter 2, “Functional Description”.

**M/A** — **Initiator Mode: Phase Mismatch; Target Mode: SATN/ Active** — **7**

In the initiator mode, this bit is set if the SCSI phase asserted by the target does not match the instruction. The phase is sampled when SREQ/ is asserted by the target. In the target mode, this bit is set when the SATN/ signal is asserted by the initiator.

**CMP** — **Function Complete** — **6**

This bit is set when an arbitration only or full arbitration sequence is completed.

**SEL** **Selected** **5**

This bit is set when the SYM53C1010 SCSI function is selected by another SCSI device. For the SYM53C1010 SCSI function to respond to selection attempts, the Enable Response to Selection bit must be set in the SCSI Chip ID (SCID) register. The Response ID Zero (RESPID0) and Response ID One (RESPID1) registers must hold the chip's ID.

**RSL** **Reselected** **4**

This bit is set when the SYM53C1010 SCSI function is reselected by another SCSI device. The Enable Response to Reselection bit must be set in the SCSI Chip ID (SCID) register (and the Response ID Zero (RESPID0) and Response ID One (RESPID1) registers must hold the chip's ID) for the SYM53C1010 SCSI function to respond to reselection attempts.

**SGE** **SCSI Gross Error** **3**

This bit is set when the SYM53C1010 SCSI function encounters a SCSI Gross Error condition. The following conditions can result in a SCSI Gross Error:

- Offset Underflow occurs in target mode when a SACK/ signal is received before the corresponding SREQ/ signal has been sent.
- Offset Overflow occurs in initiator mode when an SREQ/ signal is received and causes the maximum offset, as defined by the MO[5:0] bits in the SCSI Transfer (SXFER) register, to be exceeded.
- In initiator mode, a phase change occurs with an outstanding SREQ/SACK offset.
- Residual Data in SCSI FIFO occurs when a transfer other than Synchronous Data Received is started with data left in the SCSI Synchronous Receive FIFO.
- Multiple CRC Requests occur when, during a synchronous DT transfer, multiple CRC requests are received within the same offset.
- A request for a Pad CRC word is received without the subsequent CRC word requests.
- A phase change occurs without a CRC Request.

Note: Checking for this condition can be disabled by setting the DISCRC bit in the CRC Control Zero (CRCCNTL0) register.

- An illegal Force CRC Request Block Move is executed.
- A SCRIPTS RAM parity error occurs.

**UDC** **Unexpected Disconnect** **2**

This bit is set when the SYM53C1010 SCSI function is operating in the initiator mode and the target device unexpectedly disconnects from the SCSI bus. This bit is only valid when the SYM53C1010 SCSI function operates in the initiator mode. When the SCSI function operates in the low level mode, any disconnect causes an interrupt, even a valid SCSI disconnect. This bit is also set if a selection time-out occurs. Since a selection time-out is not considered an expected disconnect, an unexpected disconnect may occur before, at the same time, or stacked after the STO interrupt.

**RST** **SCSI RST/ Received** **1**

This bit is set when the SYM53C1010 SCSI function detects an active SRST/ signal, whether the reset is generated external to the chip or caused by the Assert SRST/ bit in the SCSI Control One (SCNTL1) register. This SCSI reset detection logic is edge-sensitive, so that multiple interrupts are not generated for a single assertion of the SRST/ signal.

**PAR** **Parity/CRC/AIP Error** **0**

This bit indicates the SYM53C1010 SCSI function detected a Parity/CRC/AIP error while receiving or sending SCSI data. See the Disable Halt on Parity/CRC/AIP error or SATN/ Condition bits in the SCSI Control One (SCNTL1) register for more information about when this condition will actually be raised.

## Register: 0x43

**SCSI Interrupt Status One (SIST1)**
**Read Only**

| 7 | | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| R | | | SBMC | R | STO | GEN | HTH |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Reading the SIST1 register returns the status of the various interrupt conditions, whether they are enabled in the SCSI Interrupt Enable One (SIEN1) register or not. Each bit set indicates an occurrence of the corresponding condition. Reading the SIST1 clears the interrupt condition.

**R** **Reserved** **[7:5]**

**SBMC** **SCSI Bus Mode Change** **4**
This bit is set when the DIFFSENS pin detects a change in voltage level indicating the SCSI bus has switched between SE, LVD, or HVD modes.

**R** **Reserved** **3**

**STO** **Selection or Reselection Time-Out** **2**
This bit is set when the SCSI device which the SYM53C1010 SCSI function is attempting to select or reselect does not respond within the programmed time-out period. See the description of the SCSI Timer Zero (STIME0) register, bits [3:0], for more information on the time-out timer.

**GEN** **General Purpose Timer Expired** **1**
This bit is set when the general purpose timer expires. The time measured is the time between enabling and disabling of the timer. See the description of the SCSI Timer One (STIME1) register, bits [3:0], for more information on the general purpose timer.

**HTH** **Handshake-to-Handshake Timer Expired** **0**
This bit is set when the handshake-to-handshake timer expires. The time measured is the SCSI Request-to-Request (target) or Acknowledge-to-Acknowledge (initiator) period. See the description of the SCSI Timer

Zero (STIME0) register, bits [7:4], for more information on the handshake-to-handshake timer.

## Register: 0x44
**Reserved**

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| R | | | | | | | |
| x | x | x | x | x | x | x | x |

This register is reserved.

## Register: 0x45
**SCSI Wide Residue (SWIDE)**
**Read/Write**

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| SWIDE | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**SWIDE** **SCSI Wide Residue** **[7:0]**
After an asynchronous wide SCSI data receive operation, this register contains a residual data byte if the last byte received was never sent across the DMA bus. It represents either the first data byte of a subsequent data transfer, a residue byte which should be cleared when an Ignore Wide Residue message is received, or an overrun data byte. The power-up value of this register is indeterminate.

## Register: 0x46
**Reserved**

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| R | | | | | | | |
| x | x | x | x | x | x | x | x |

This register is reserved.

## Register: 0x47

**General Purpose Pin Control (GPCNTL)**
**Read/Write**

| 7 | 6 | 5 | 4 | | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ME | FE | LEDC | GPIO[4:2] | | | GPIO[1:0] | |
| 0 | 0 | x | 0 | 1 | 1 | 1 | 1 |

This register is used to determine if the pins controlled by the General Purpose (GPREG) register are inputs or outputs. Bits [4:0] in GPCNTL correspond to bits [4:0] in the GPREG register. When the bits are enabled as inputs, an internal pull-down is also enabled.

**ME** — **Master Enable** — **7**

When the ME bit is set, the bus master state of the device is presented on GPIO1. GPIO1 goes low when the part is a bus master. When set, the ME bit is independent of the setting of bit 1 (GPIO1). If the GPIO1 is configured as an input while the ME bit is set, the Master bit will still toggle the GPIO1 pin.

**FE** — **Fetch Enable** — **6**

If the FE bit is set, GPIO0 reflects when an internal opcode fetch is being performed. GPIO0 goes low when an opcode fetch is performed. When set, the FE bit is independent of the setting of bit 0 (GPIO0). If GPIO0 is configured as an input, the Fetch bit still toggles GPIO0.

**LEDC** — **LED_CNTL** — **5**

If the LED_CNTL bit is set GPIO0 reflects the state of the SCSI bus, connected (low) or not connected (GPIO0 high). This occurs if bit 6 (FE) is not set and the chip is not currently performing an EEPROM autodownload. This bit provides a hardware solution for driving an external SCSI activity LED.

**GPIO[4:2]** — **GPIO Enable** — **[4:2]**

The general purpose control corresponds to bits [4:2] in the General Purpose (GPREG) register and to the GPIO4–GPIO2 pins. GPIO4 powers up as a general purpose output. GPIO[3:2] power-up as general purpose inputs.

**GPIO[1:0]** **GPIO Enable** **[1:0]**

These bits are set at power-up causing the GPIO1 and GPIO0 pins to become inputs. Clearing these bits cause GPIO[1:0] to become outputs.

## Register: 0x48

**SCSI Timer Zero (STIME0)**

**Read/Write**

| 7 | | | 4 | 3 | | | 0 |
|---|---|---|---|---|---|---|---|
| HTH[3:0] | | | | SEL[3:0] | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**HTH[3:0]** **Handshake-to-Handshake Timer Period** **[7:4]**

These bits select the handshake-to-handshake time-out period, which is the maximum time between SCSI handshakes (SREQ/ to SREQ/ in target mode; or, SACK/ to SACK/ in initiator mode). When this timing is exceeded, an interrupt is generated and the HTH bit in the SCSI Interrupt Status One (SIST1) register is set. The following table contains time-out periods for the Handshake-to-Handshake Timer, the Selection/ Reselection Timer (bits [3:0]), and the General Purpose Timer (SCSI Timer One (STIME1), bits [3:0]). For a more detailed explanation of interrupts, refer to Chapter 2, “Functional Description”.

| HTH [3:0], SEL [3:0] | Minimum Time-Out |
|---|---|
| 0000 | Disabled |
| 0001 | 125 µs |
| 0010 | 250 µs |
| 0011 | 500 µs |
| 0100 | 1 ms |
| 0101 | 2 ms |
| 0110 | 4 ms |
| 0111 | 8 ms |
| 1000 | 16 ms |
| 1001 | 32 ms |
| 1010 | 64 ms |
| 1011 | 128 ms |
| 1100 | 256 ms |
| 1101 | 512 ms |
| 1110 | 1.024 s |
| 1111 | 2.048 s |

**SEL[3:0]** **Selection Time-Out** **[3:0]**
These bits select the SCSI selection/reselection time-out period. When this timing (plus the 200 µs selection abort time) is exceeded, the STO bit in the SCSI Interrupt Status One (SIST1) register is set. For a more detailed explanation of interrupts, refer to Chapter 2, "Functional Description".

## Register: 0x49
### SCSI Timer One (STIME1)
### Read/Write

| 7 | 6 | 5 | 4 | 3 | | | 0 |
|---|---|---|---|---|---|---|---|
| R | HTHBA | GENSF | HTHSF | GEN[3:0] | | | |
| x | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**R** **Reserved** **7**

**HTHBA** **Handshake-to-Handshake Timer Bus Activity Enable** **6**
Setting this bit causes this timer to begin testing for SCSI REQ/ and ACK/ activity as soon as SBSY/ is asserted, regardless of the agents participating in the transfer.

**GENSF** **General Purpose Timer Scale Factor** **5**

Setting this bit causes this timer to shift by a factor of 16. Refer to the SCSI Timer Zero (STIME0) register description for details.

| HTH[3:0], SEL[3:0], GEN [3:0] | Minimum Time-out | |
|---|---|---|
| | HTHSF = 0, GENSF = 0 | HTHSF = 1, GENSF = 1 |
| 0000 | Disabled | Disabled |
| 0001 | 125 µs | 2 ms |
| 0010 | 250 µs | 4 ms |
| 0011 | 500 µs | 8 ms |
| 0100 | 1 µs | 16 ms |
| 0101 | 2 ms | 32 ms |
| 0110 | 4 ms | 64 ms |
| 0111 | 8 ms | 128 ms |
| 1000 | 16 ms | 256 ms |
| 1001 | 32 ms | 512 ms |
| 1010 | 64 ms | 1 s |
| 1011 | 128 ms | 2 s |
| 1100 | 256 ms | 4.1 s |
| 1101 | 512 ms | 8.2 s |
| 1110 | 1.024 s | 16.4 s |
| 1111 | 2.048 s | 32.8 s |

**HTHSF** **Handshake-to-Handshake Timer Scale Factor** **4**

Setting this bit causes this timer to shift by a factor of 16. Refer to the SCSI Timer Zero (STIME0) register description for details.

**GEN[3:0]** **General Purpose Timer Period** **[3:0]**

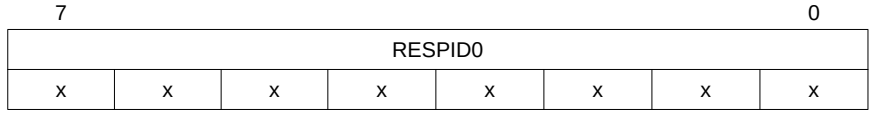
These bits select the period of the general purpose timer. The time measured is the time between enabling and disabling of the timer. When this timing is exceeded, the GEN bit in the SCSI Interrupt Status One (SIST1) register is set. Refer to the table under SCSI Timer Zero (STIME0), bits [3:0], for the available time-out periods.

Note: To reset a timer before it expires and obtain repeatable delays, the time value must be written to zero first, and then written back to the desired value. This is also required when changing from one time value to another.

## Register: 0x4A

**Response ID Zero (RESPID0)**

**Read/Write**



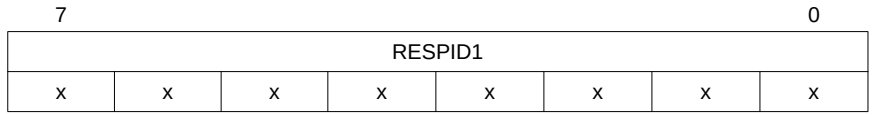

**RESPID0** **Response ID Zero** **[7:0]**

RESPID0 and Response ID One (RESPID1) contain the selection or reselection IDs. These two 8-bit registers contain the SCSI ID that the chip responds to on the SCSI bus. Each bit represents one possible ID; the most significant bit of Response ID One (RESPID1) represents ID 15, and the least significant bit of RESPID0 represents ID 0. The SCSI Chip ID (SCID) register still contains the chip ID used during arbitration. The chip can respond to more than one ID because more than one bit can be set in the RESPID1 and RESPID0 registers. However, the chip can arbitrate with only one ID value in the SCSI Chip ID (SCID) register.

## Register: 0x4B

**Response ID One (RESPID1)**

**Read/Write**




**RESPID1** **Response ID One** **[7:0]**

Response ID Zero (RESPID0) and RESPID1 contain the selection or reselection IDs. These two 8-bit registers contain the SCSI ID that the chip responds to on the SCSI bus. Each bit represents one possible ID; the most significant bit of RESPID1 represents ID 15, and the least significant bit of RESPID0 represents ID 0. The SCSI Chip ID (SCID) register still contains the chip ID used during arbitration. The chip can respond to more than one ID because more than one bit can be set in the RESPID1 and RESPID0 registers. However, the chip can arbitrate with only one ID value in the SCID register.

## Register: 0x4C
### SCSI Test Zero (STEST0)
### Read Only

| 7 | | | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SSAID[3:0] | | | | SLT | ART | SOZ | SOM |
| 0 | 0 | 0 | 0 | 0 | x | 1 | 1 |

**SSAID[3:0]** **SCSI Selected As ID** **[7:4]**

These bits contain the encoded value of the SCSI ID that the SYM53C1010 SCSI function is selected or reselected as during a SCSI Selection or Reselection phase. These bits are read only and contain the encoded value of 0–15 possible IDs that could be used to select the SYM53C1010 SCSI function. During a SCSI Selection or Reselection phase when a valid ID is put on the bus, and the SYM53C1010 SCSI function responds to that ID, the "selected as" ID is written into these bits. These bits are used with Response ID Zero (RESPID0) and Response ID One (RESPID1) registers to allow response to multiple IDs on the bus.

**SLT** **Selection Response Logic Test** **3**

This bit is set when the SYM53C1010 SCSI function is ready to be selected or reselected. This does not take into account the bus settle delay of 400 ns. This bit is used for functional test and fault purposes.

**ART** **Arbitration Priority Encoder Test** **2**

This bit is always set when the SYM53C1010 SCSI function exhibits the highest priority ID asserted on the SCSI bus during arbitration. It is primarily used for chip level testing. It may be used during low level mode operation to determine if the SYM53C1010 SCSI function won arbitration.

**SOZ** **SCSI Synchronous Offset Zero** **1**

This bit indicates that the current synchronous SREQ/, SACK/ offset is zero. This bit is not latched and may change at any time. It is used in low level synchronous SCSI operations. When this bit is set and if the SYM53C1010 is functioning as an initiator, the device is waiting for the target to request data transfers. When this bit is set and if the SYM53C1010 is functioning as a

target, then the initiator has sent the offset number of acknowledges.

**SOM** **SCSI Synchronous Offset Maximum** **0**
This bit indicates that the current synchronous SREQ/, SACK/ offset is the maximum specified by bits [5:0] in the SCSI Transfer (SXFER) register. This bit is not latched and may change at any time. It is used in low level synchronous SCSI operations. If this bit is set and if the SYM53C1010 is functioning as a target, it is waiting for the initiator to acknowledge the data transfers. If the SYM53C1010 SCSI is functioning as an initiator, the target has sent the offset number of requests.

## Register: 0x4D
### SCSI Test One (STEST1)
### Read/Write

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| R | | DOSGE | DISGE | QEN | QSEL | IRM[1:0] | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**R** **Reserved** **[7:6]**

**DOSGE** **Disable Outbound SCSI Gross Errors** **5**
When set, this bit disables all SCSI gross errors related to outbound data transfers.

**DISGE** **Disable Inbound SCSI Gross Errors** **4**
When set, this bit disables all SCSI gross errors related to inbound data transfers.

**QEN** **SCLK Quadrupler Enable** **3**
This bit, when set, powers up the internal clock quadrupler circuit, which quadruples the SCLK 40 MHz clock to the internal 160 MHz SCSI clock required for Ultra2 and Ultra3 SCSI operation. When cleared, this bit powers down the internal quadrupler circuit. Refer to Chapter 2, “Functional Description” for information concerning the operation of the quadrupler.

**QSEL** **SCLK Quadrupler Select** **2**
This bit, when set, selects the output of the internal clock quadrupler as the internal SCSI clock. When cleared, this

bit selects the clock presented on SCLK as the internal SCSI clock. Refer to Chapter 2, "Functional Description" for information concerning the operation of the quadrupler.

**IRM[1:0]** **Interrupt Routing Mode** **[1:0]**

The SYM53C1010 supports four different interrupt routing modes. These modes are described in the following table. Each SCSI core within the chip can be configured independently. Mode 0, the default mode, is compatible with RAID upgrade products.

| Mode | Bits [1:0] | Operation |
|---|---|---|
| 0 | 00 | If the INT_DIR/ input pin is low, interrupts are signaled on ALT_INTx/. Otherwise, interrupts are signaled on both INTx/ and ALT_INTx/. |
| 1 | 01 | Interrupts are only signaled on INTx/, not ALT_INTx/. The INT_DIR/ input pin is ignored. |
| 2 | 10 | Interrupts are only signaled on ALT_INTx/. The INT_DIR/ input pin is ignored. |
| 3 | 11 | Interrupts are signaled on both INTx/ and ALT_INTx/. The INT_DIR input pin is ignored. |

## Register: 0x4E

### SCSI Test Two (STEST2)

### Read/Write

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SCE | ROF | R | | SZM | R | | LOW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**SCE** **SCSI Control Enable** **7**

Setting this bit allows assertion of all SCSI control and data lines through the SCSI Output Control Latch (SOCL) and SCSI Output Data Latch (SODL) registers regardless of whether the SYM53C1010 SCSI function is configured as a target or initiator.

Note: Do not set this bit during normal operation, since it could cause contention on the SCSI bus. It is included for diagnostics purposes only.

**ROF** **Reset SCSI Offset** **6**
Setting this bit clears any outstanding synchronous SREQ/SACK offset. If a SCSI gross error occurs, set this bit. This bit automatically clears itself after resetting the synchronous offset.

**R** **Reserved** **[5:4]**

**SZM** **SCSI High Impedance Mode** **3**
Setting this bit places all the open drain 48 mA SCSI drivers into a high impedance state.

**R** **Reserved** **[2:1]**

**LOW** **SCSI Low Level Mode** **0**
Setting this bit places the SYM53C1010 SCSI function in the low level mode. In this mode, no DMA operations occur and no SCRIPTS execute. Arbitration and selection may be performed by setting the start sequence bit as described in the SCSI Control Zero (SCNTL0) register. SCSI bus transfers are performed by manually asserting and polling SCSI signals. Clearing this bit allows instructions to be executed in the SCSI SCRIPTS mode.

Note: It is not necessary to set this bit for access to the SCSI bit-level registers (SCSI Output Data Latch (SODL), SCSI Bus Control Lines (SBCL), and input registers).

## Register: 0x4F

**SCSI Test Three (STEST3)**

**Read/Write**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TE | R | HSC | DSI | R | TTM | CSF | R |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**TE** **TolerANT Enable** **7**

Setting this bit enables the active negation portion of LSI Logic TolerANT technology. Active negation causes the SCSI Request, Acknowledge, Data, and Parity signals to be actively deasserted, instead of relying on external pull-ups, when the SYM53C1010 SCSI function is driving these signals. Active deassertion of these signals occurs only when the SYM53C1010 SCSI function is in an information transfer phase. When performing synchronous transfers, TolerANT should be enabled to improve setup and deassertion times. Active negation is disabled after reset or when this bit is cleared. For more information on LSI Logic TolerANT technology, see Chapter 1, “Introduction”.

**R** **Reserved** **6**

**HSC** **Halt SCSI Clock** **5**

Setting this bit causes the internal, divided SCSI clock to come to a stop in a glitchless manner. This bit is used for test purposes or to lower $I_{DD}$ during a power-down mode. Refer to Chapter 2, “Functional Description” for operation of the SCSI clock quadrupler.

**DSI** **Disable Single Initiator Response** **4**

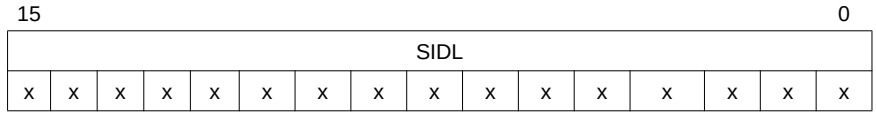
If this bit is set, the SYM53C1010 SCSI function ignores all bus-initiated selection attempts that employ the single initiator option from SCSI-1. In order to select the SYM53C1010 SCSI function while this bit is set, the SYM53C1010 SCSI function’s SCSI ID and the initiator’s SCSI ID must both be asserted. Assert this bit in SCSI-2 systems so that a single bit error on the SCSI bus is not interpreted as a single initiator response.

| | | |
|---|---|---|
| **R** | **Reserved** | **3** |
| **TTM** | **Timer Test Mode**<br>Setting this bit facilitates testing of the selection time-out, general purpose, and handshake-to-handshake timers by greatly reducing all three time-out periods. Setting this bit starts all three timers. If the respective bits in the SCSI Interrupt Enable One (SIEN1) register are asserted, the SYM53C1010 SCSI function generates interrupts at time-out. This bit is intended for internal manufacturing diagnosis and should not be used in normal operation. | **2** |
| **CSF** | **Clear SCSI FIFO**<br>Setting this bit causes the “full flags” for the SCSI FIFO to be cleared. This empties the FIFO. This bit is self-clearing. In addition to the SCSI FIFO pointers, the SIDL, SODL, and SODR full bits in the SCSI Status Zero (SSTAT0) and SCSI Status Two (SSTAT2) are cleared. | **1** |
| **R** | **Reserved** | **0** |

## Registers: 0x50–0x51

### SCSI Input Data Latch (SIDL)

### Read Only




| | | |
|---|---|---|
| **SIDL** | **SCSI Input Data Latch**<br>This register is used primarily for diagnostics testing, programmed I/O operation, or error recovery. Asynchronous Data received from the SCSI bus can be read from this register. When receiving asynchronous SCSI data, the data flows into this register and out to the host FIFO. This register differs from the SCSI Bus Data Lines (SBDL) register; the SCSI Input Data Latch (SIDL) contains latched data and the SCSI Bus Data Lines (SBDL) always contains exactly what is currently on the SCSI data bus. Reading this register causes the SCSI parity bit to be checked, and causes a parity error interrupt if the data is invalid. The power-up values are indeterminate. | **[15:0]** |

## Register: 0x52

**SCSI Test Four (STEST4)**

**Read Only**

| 7 | 6 | 5 | | | | | 0 |
|---|---|---|---|---|---|---|---|
| SMODE[1:0] | | R | | | | | |
| x | x | 0 | 0 | 0 | 0 | 0 | 0 |

**SMODE[1:0] SCSI Mode [7:6]**

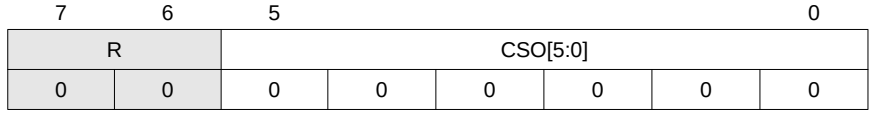
These bits contain the encoded value of the SCSI operating mode that is indicated by the voltage level sensed at the DIFFSENS pin. The incoming SCSI signal goes to a pair of analog comparators that determine the voltage window of the DIFFSENS signal. These voltage windows indicate LVD, SE, or HVD operation. The bit values are defined in the following table. When the HVD mode is detected, all of the SYM53C1010 3-state outputs go to the high impedance state.

| SMODE [1:0] | Operating Mode |
|---|---|
| 00 | Reserved |
| 01 | High Impedance State |
| 10 | SE |
| 11 | LVD SCSI |

**R Reserved [5:0]**

## Register: 0x53

**Current Inbound SCSI Offset (CSO)**
**Read Only**



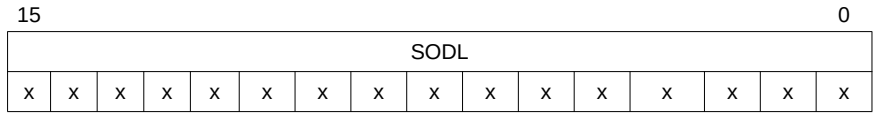
| 7 | 6 | 5 | | | | | 0 |
|---|---|---|---|---|---|---|---|
| R | | CSO[5:0] | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**R** **Reserved** **[7:6]**

**CSO[5:0]** **Current SCSI Offset** **[5:0]**
These bits indicate the SCSI offset for synchronous inbound transfers. This also represents the number of data bytes in the SCSI FIFO in narrow transfer modes and half the number of bytes in wide transfer mode. This does not include any CRC or PAD bytes that may be in the FIFO.

## Registers: 0x54–0x55

**SCSI Output Data Latch (SODL)**
**Read/Write**



| 15 | | | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SODL | | | | | | | | | | | | | | | |
| x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |

**SODL** **SCSI Output Data Latch** **[15:0]**
This register is used primarily for diagnostics testing and programmed I/O operations. Data written to this register is asserted on the SCSI data bus by setting the Assert Data Bus bit in the SCSI Control One (SCNTL1) register. This register is used to send data using programmed I/O. Data flows through this register when sending data in asynchronous mode. It is also used to write to the synchronous data FIFO when testing the chip. The power-up value of this register is indeterminate.

## Register: 0x56

**Chip Control Zero (CCNTL0)**

**Read/Write**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ENPMJ | PMJCTL | ENNDJ | DISFC | R | | DISRC | DPR |
| 0 | 0 | 0 | 0 | x | x | 0 | 0 |

**ENPMJ** **Enable Phase Mismatch Jump** **7**

Upon setting this bit, any phase mismatches do not interrupt but force a jump to an alternate location to handle the phase mismatch. Prior to actually taking the jump, the appropriate remaining byte counts and addresses are calculated to facilitate storage.

In the case of a SCSI send, any data in the part is automatically cleared after being accounted for. In the case of a SCSI receive, all data will be flushed out of the part and accounted for prior to taking the jump. This feature does not cover, however, the byte that may appear in SCSI Wide Residue (SWIDE). This byte must be flushed manually.

This bit also enables the flushing mechanism to flush data during a Data-In phase mismatch in a more efficient manner.

**PMJCTL** **Jump Control** **6**

This bit controls which decision mechanism is used when jumping on phase mismatch. When this bit is cleared the SYM53C1010 will use Phase Mismatch Jump Address One (PMJAD1) when the WSR bit is cleared and Phase Mismatch Jump Address Two (PMJAD2) when the WSR bit is set. When this bit is set, the SYM53C1010 will use Phase Mismatch Jump Address One (PMJAD1) on Data-Out (Data-Out, Command, Message-Out) transfers and Phase Mismatch Jump Address Two (PMJAD2) on Data-In (Data-In, Status, Message-In) transfers. The phase referred to here is the phase encoded in the block move SCRIPTS instruction, not the phase on the SCSI bus that caused the phase mismatch.

**ENNDJ** **Enable Jump On Nondata Phase Mismatches** **5**

This bit controls whether or not a jump is taken during a nondata phase mismatch (Message-In, Message-Out,

Status, or Command). When this bit is cleared, jumps will only be taken on Data-In or Data-Out phases, and a phase mismatch interrupt will be generated for all other phases. When this bit is set, jumps will be taken regardless of the phase in the block move. Note that the phase referred to here is the phase encoded in the block move SCRIPTS instruction, not the phase on the SCSI bus that caused the phase mismatch.

**DISFC** **Disable Auto FIFO Clear** **4**

This bit controls whether or not the FIFO is automatically cleared during a Data-Out phase mismatch. When set, data in the DMA FIFO and in the SCSI Output Data Latch (SODL) and SODR (a hidden buffer register which is not accessible) registers are not cleared after calculations on them are complete. When cleared, the DMA FIFO, SODL, and SODR are automatically cleared. This bit also disables the enhanced flushing mechanism.

**R** **Reserved** **[3:2]**

**DISRC** **Disable Internal SCRIPTS RAM Cycles** **1**

This bit controls whether or not data transfers, for which the source/destination is located in SCRIPTS RAM, generate external PCI cycles.

If cleared, data transfers of this type do NOT generate PCI cycles and stay internal to the chip. If set, data transfers of this type generate PCI cycles. This does not affect SCRIPTS Fetch operations from SCRIPTS RAM, including Table Indirect and Indirect opcode fetches.

**DPR** **Disable Pipe Req** **0**

This bit controls whether or not overlapped arbitration on the PCI bus occurs. Overlapped arbitration is performed by asserting PCI REQ/ for one SCSI function while the other SCSI function is executing a PCI cycle. If set, overlapped arbitration is disabled.

## Register: 0x57

**Chip Control One (CCNTL1)**
**Read/Write**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PULLDIS | PULLEN | DIS64MAS | DIS64SLV | DDAC | 64TIMOD | EN64TIBMV | EN64DBMV |
| 0 | 0 | 0 | 0 | x | x | 0 | 0 |

**PULLDIS** **Pull Disable** **7**
Setting this bit causes all internal pulls to be disabled on all pins. This bit is intended for manufacturing test only and should NOT be set for normal operation. PULLDIS has precedence over PULLEN if both bits are set.

**PULLEN** **Pull Enable** **6**
Setting this bit causes all internal pulls to be enabled on all pins. This bit is intended for manufacturing test only and should NOT be set for normal operation.

**DIS64MAS** **Disable 64-bit Master Operation** **5**
Setting this bit causes the SYM53C1010 to no longer request 64-bit master data transfers. If this bit is set by either SCSI channel, 64-bit data transfers will be disabled for all master transactions.

**DIS64SLV** **Disable 64-bit Slave Cycles** **4**
Setting this bit disables 64-bit slave data transfers to the SCRIPTS RAM. This causes only 32-bit data transfers to occur.

**DDAC** **Disable Dual Address Cycle** **3**
When this bit is set, all 64-bit addressing as a master is disabled. No dual address cycles will be generated by the SYM53C1010.

When this bit is cleared, the SYM53C1010 generates dual address cycles based on the master operation performed and the value of its associated selector register.

**64TIMOD** **64-bit Table Indirect Indexing Mode** **2**
When this bit is cleared, bits [28:24] of the first table entry Dword will select one of 22 possible selectors to be used in a BMOV operation. When this bit is set, bits [31:24] of the first table entry Dword will be copied directly into

DMA Next Address 64 (DNAD64) to provide 40-bit addressing capability. This bit will only function if the EN64TIBMV bit is set.

Index Mode 0 (64TIMOD clear) table entry format:

| Bits [31:29] | Bits [28:24] | Bits [23:0] |
|---|---|---|
| Reserved | Sel Index | Byte Count |
| Source/Destination Address | | |

Index Mode 1 (64TIMOD set) table entry format:

| Bits [31:24] | Bits [23:0] |
|---|---|
| Src/Dest Addr [39:32] | Byte Count |
| Source/Destination Address [31:0] | |

**EN64TIBMV** **Enable 64-bit Table Indirect BMOV** **1**
Setting this bit enables 64-bit addressing for Table Indirect BMOVs using the upper byte (bits [31:24]) of the first Dword of the table entry. When this bit is cleared, Table Indirect BMOVs use the Static Block Move Selector (SBMS) register to obtain the upper 32 bits of the data address.

**EN64DBMV** **Enable 64-bit Direct BMOV** **0**
Setting this bit enables the 64-bit version of a direct BMOV. When this bit is cleared, direct BMOVs use the Static Block Move Selector (SBMS) register to obtain the upper 32 bits of the data address.

## Registers: 0x58–0x59

### SCSI Bus Data Lines (SBDL)

### Read Only

| 15 | | | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SBDL | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**SBDL** **SCSI Bus Data Lines** **[15:0]**
This register contains the SCSI data bus status. Even though the SCSI data bus is active low, these bits are active high. The signal status is not latched and is a true representation of exactly what is on the data bus at the

time the register is read. This register is used when receiving data using programmed I/O. This register can also be used for diagnostics testing or in the low level mode. The power-up value of this register is indeterminate.

If the chip is in wide mode SCSI Control Three (SCNTL3), bit 3 is set) and SCSI Bus Data Lines (SBDL) is read, both byte lanes are checked for parity regardless of phase. When in a nondata phase, this will cause a parity error interrupt to be generated because the upper byte lane parity is invalid.

## Register: 0x5A

**Reserved**

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| R | | | | | | | |
| x | x | x | x | x | x | x | x |

This register is reserved.

## Register: 0x5B

**Chip Control Three (CCNTL3)**

**Read/Write**

| 7 | | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| R | | | ENDSKEW | DSKEW[1:0] | | LVDDL[1:0] | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**R** **Reserved** **[7:5]**

**ENDSKEW** **Enable REQ/ACK to Data Skew Control** **4**
Setting this bit enables the control of the relative skew between the SCSI REQ/ACK signals and the data signals. The actual amount of skew time is controlled by DSKEW[1:0] in this register.

**DSKEW[1:0]** **Data Skew Control** **[3:2]**
These bits control the amount of skew between the SCSI REQ/ACK signal and the SCSI data signals. The skew is affected only if the ENDSKEW bit is set.

Note: These bits are used for Ultra3 SCSI Domain Validation only and should not be set during normal data transfer operations.

**LVDDL[1:0]** **LVD Drive Strength Select** **[1:0]**
These bits control the drive level of the LVD pad drivers.

Note: This feature is for Ultra3 SCSI Domain Validation testing environments only and should not be set during normal data transfer operations.

The table below shows the relative strength increase or decrease based on the LVDDL values.

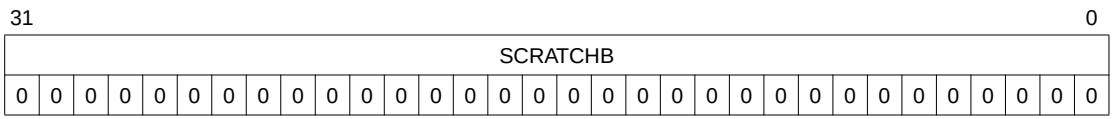
| LVDDL | Drive Level |
|---|---|
| 00 | Nominal |
| 01 | –20% Nominal |
| 10 | +20% Nominal |
| 11 | Reserved |

Note: If one of the LVDDL [1:0] bits are set on either channel, both channels are affected.

## Registers: 0x5C–0x5F

### Scratch Register B (SCRATCHB)

**Read/Write**




**SCRATCHB** **Scratch Register B** **[31:0]**
This is a general purpose user-definable scratch pad register. Apart from CPU access, only register Read/Write and Memory Moves directed at the SCRATCH register will alter its contents. The power-up values are indeterminate. A special mode of this register can be enabled by setting the PCI Configuration Info Enable bit in the Chip Test Two (CTEST2) register. If this bit is set, bits [31:13] of the Scratch Register B (SCRATCHB) register return bits [31:13] of the PCI Base Address Register Three (BAR3) (SCRIPTS RAM). In this mode, bits [12:0] of SCRATCH B will always return zeros. Writes to the SCRATCH B register have no effect.

Resetting the PCI Configuration Info Enable bit causes the SCRATCH B register to return to normal operation.

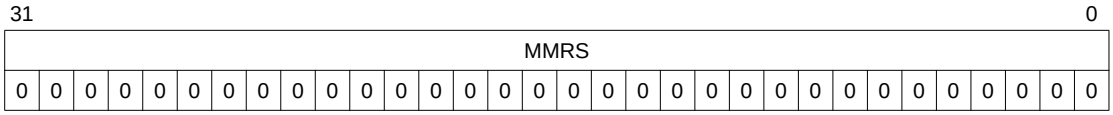
## Registers: 0x60–0x9F

**Scratch Registers C–R (SCRATCHC–SCRATCHR)**
**Read/Write**

These are general purpose user-definable scratch pad registers. Apart from CPU access, only register Read/Write, Memory Moves, and Load/Stores directed at a SCRATCH register alter its contents. The power-up values are indeterminate.

## Registers: 0xA0–0xA3

**Memory Move Read Selector (MMRS)**
**Read/Write**

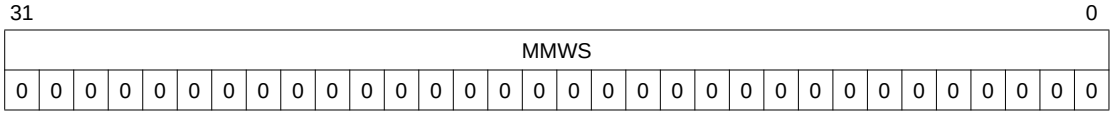



**MMRS** **Memory Move Read Selector** **[31:0]**
This register supplies AD[63:32] during data read operations for Memory-to-Memory Moves and absolute address LOAD operations.

A special mode of this register can be enabled by setting the PCI Configuration Info Enable bit in the Chip Test Two (CTEST2) register. If this bit is set, the Memory Move Read Selector (MMRS) register returns bits [31:0] of the memory mapped operating register, PCI Base Address Register Two (BAR2) (MEMORY), when read. In this mode, writes to the MMRS register have no effect. Clearing the PCI Configuration Info Enable bit causes the MMRS register to return to normal operation.

## Registers: 0xA4–0xA7

**Memory Move Write Selector (MMWS)**
**Read/Write**



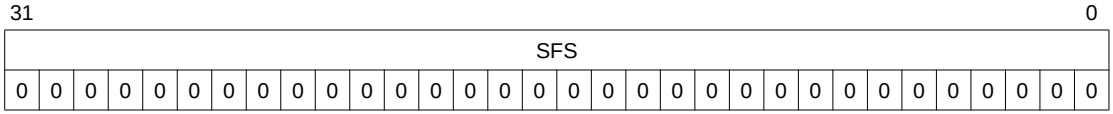

**MMWS** **Memory Move Write Selector** **[31:0]**

This register supplies AD[63:32] during data write operations during Memory-to-Memory Moves and absolute address STORE operations.

A special mode of this register can be enabled by setting the PCI Configuration Info Enable bit in the Chip Test Two (CTEST2) register. If this bit is set, the MMWS register returns bits [31:0] of the SCRIPTS RAM PCI Base Address Register Four (BAR4) (SCRIPTS RAM) in bits [31:0] of the MMWS register when read. In this mode, writes to the MMWS register have no effect. Clearing the PCI Configuration Info Enable bit causes the MMWS register to return to normal operation.

## Registers: 0xA8–0xAB

**SCRIPT Fetch Selector (SFS)**
**Read/Write**




**SFS** **SCRIPT Fetch Selector** **[31:0]**

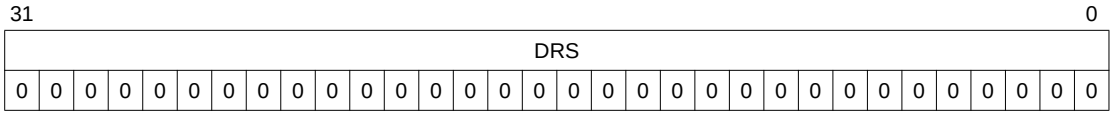
This register supplies AD[63:32] during SCRIPT Fetches and Indirect Fetches (excluding Table Indirect Fetches). This register can be loaded automatically using a 64-bit jump instruction.

A special mode of this register can be enabled by setting the PCI Configuration Info Enable bit in the Chip Test Two (CTEST2) register. If this bit is set, bits [16:23] of this register return the PCI Revision ID (RID) register value and bits [0:15] return the PCI Device ID register value when read.

Writes to the SCRIPT Fetch Selector (SFS) register are unaffected. Clearing the PCI Configuration Info Enable bit causes the SFS register to return to normal operation.

## Registers: 0xAC–0xAF

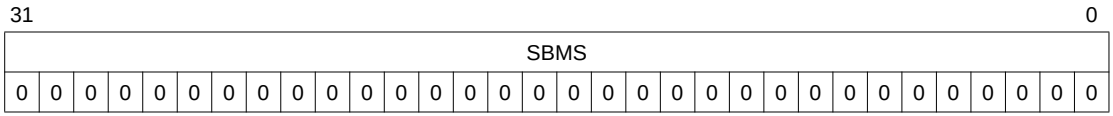
**DSA Relative Selector (DRS)**
**Read/Write**




**DRS** **DSA Relative Selector** **[31:0]**
This register supplies AD[63:32] during Table Indirect Fetches and Load/Store Data Structure Address (DSA) relative operations.

## Registers: 0xB0–0xB3

**Static Block Move Selector (SBMS)**
**Read/Write**




**SBMS** **Static Block Move Selector** **[31:0]**
This register supplies AD[63:32] during block move operations, reads, or writes. This register is static and is not changed when a 64-bit direct BMOV is used.

## Registers: 0xB4–0xB7

**Dynamic Block Move Selector (DBMS)**
**Read/Write**

| 31 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DBMS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**DBMS** **Dynamic Block Move Selector** **[31:0]**
This register supplies AD[63:32] during block move operations, reads, or writes. This register is used only during 64-bit direct BMOV instructions. It is reloaded with the upper 32 bit data address upon execution of 64-bit direct BMOVs.

## Registers: 0xB8–0xBB

**DMA Next Address 64 (DNAD64)**
**Read/Write**

| 31 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DNAD64 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**DNAD64** **DMA Next Address 64** **[31:0]**
This register holds the current selector being used in a host transaction. The appropriate selector is copied to this register prior to beginning the host transaction.

Note: The crossing of selector boundaries in one memory operation is not supported.

## Register: 0xBC

**SCSI Control Four (SCNTL4)**
**Read/Write**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| U3EN | AIPEN | R | | XCLKH_DT | XCLKH_ST | XCLKS_DT | XCLKS_ST |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

This register is automatically loaded when a Table Indirect Select or Reselect SCRIPTS instruction is executed.

**U3EN** **Ultra3 Transfer Enable** **7**
Setting this bit enables Ultra3 transfers. This bit will force all SCSI Block Move SCRIPTS instructions for ST Data-In or ST Data-Out phases to become DT Data-In or DT Data-Out phases.

**AIPEN** **Asynchronous Information Protection Enable** **6**
Setting this bit enables the AIP checking and generation of the upper byte lane of protection information during Command, Status, and Message phases.

**R** **Reserved** **[5:4]**

**XCLKH_DT** **Extra Clock of Data Hold on DT Transfer Edge** **3**
Setting this bit adds a clock of data hold to synchronous DT SCSI transfers on the DT edge. This bit only impacts DT transfers as it affects data hold to the DT edge. Setting this bit reduces the synchronous transfer send rate but does not reduce the transfer rate at which the SYM53C1010 can receive inbound REQs, ACKs or data. Refer to Table 4.4 and Table 4.5 for a summary of available transfer rates and to Figure 4.1 through Figure 4.3 for examples of how the XCLKH bits function.

Note: This bit does not affect CRC timings.

**XCLKH_ST** **Extra Clock of Data Hold on ST Transfer Edge** **2**
Setting this bit adds a clock of data hold to synchronous DT or ST SCSI transfers on the ST edge. This bit impacts both ST and DT transfers as it affects data hold to the ST edge. Setting this bit reduces the synchronous send transfer rate but does not reduce the transfer rate at which the SYM53C1010 can receive inbound REQs, ACKs or data. Refer to Table 4.4 and Table 4.5 for a summary of available transfer rates and to Figure 4.1 through Figure 4.3 for examples of how the XCLKH bits function.

Note: This bit does not affect CRC timings.

**XCLKS_DT** **Extra Clock of Data Setup on DT Transfer Edge** **1**
Setting this bit adds a clock of data setup to synchronous DT SCSI transfers on the DT edge. This bit only impacts DT transfers as it only affects data setup to the DT edge. Setting this bit reduces the synchronous transfer send rate but does not reduce the transfer rate at which the

SYM53C1010 can receive inbound REQs, ACKs or data. Refer to Table 4.4 and Table 4.5 for a summary of available transfer rates and to Figure 4.1 through Figure 4.3 for examples of how the XCLKS bits function.

Note: This bit does not affect CRC timings.

**XCLKS_ST** **Extra Clock of Data Setup on ST Transfer Edge** **0**
Setting this bit adds a clock of data setup to synchronous DT or ST SCSI transfers on the ST edge. This bit impacts both ST and DT transfers as it affects data setup to the ST edge. Setting this bit reduces the synchronous send transfer rate but does not reduce the transfer rate at which the SYM53C1010 can receive inbound REQs, ACKs or data. Refer to Table 4.4 and Table 4.5 for a summary of available transfer rates and to Figure 4.1 through Figure 4.3 for examples of how the XCLKS bits function.

Note: This bit does not affect CRC timings.

**Synchronous Receive Rate Calculation**
The synchronous receive rate, in megatransfers/s, can be calculated using the following formulas:

$$\text{Receive Rate (DT)} = \frac{\text{Input Clock Rate}}{\text{SCF Divisor} \times 2}$$

$$\text{Receive Rate (ST)} = \frac{\text{Input Clock Rate}}{\text{SCF Divisor} \times 4}$$

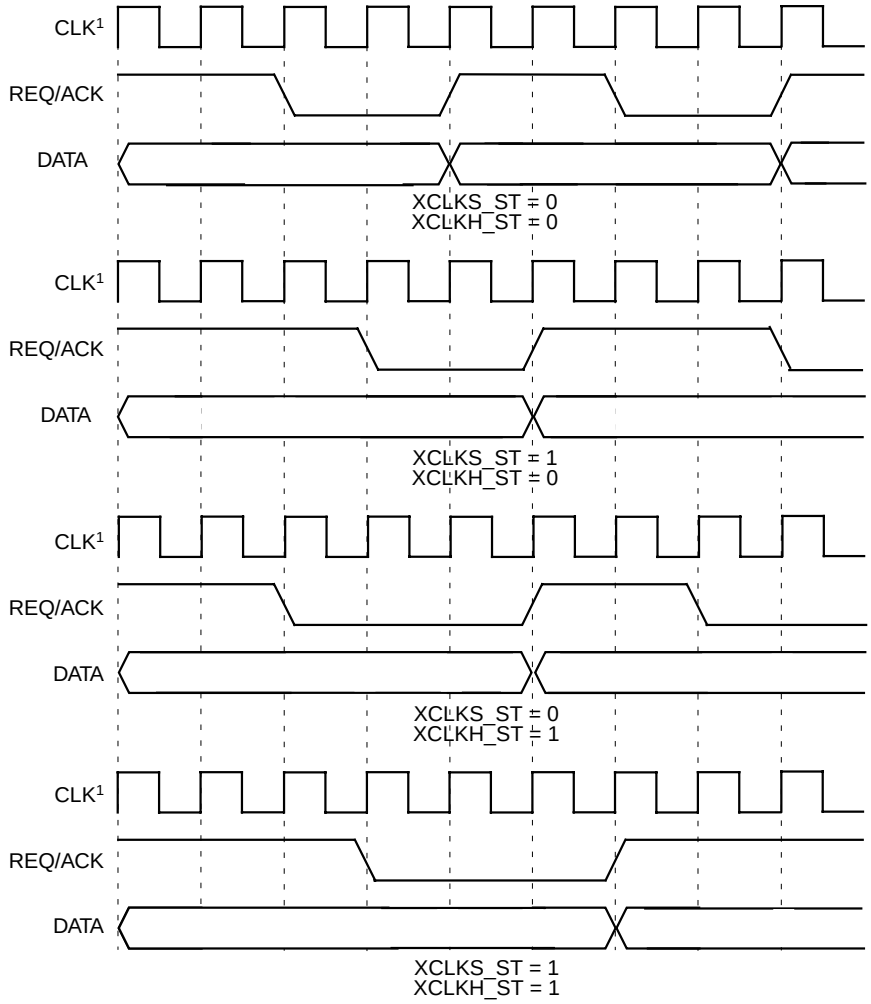
Note: The receive rate is independent of the settings of the XCLKS_DT, XCLKS_ST, XCLKH_DT, and XCLKH_ST bits.

**Synchronous Send Rate Calculation**
The synchronous send rate, in megatransfers/s, can be calculated using the following formula:

$$\text{Send Rate (DT)} = \frac{\text{Input Clock Rate}}{\left(\text{SCFDivisor} \times \left(2 + \frac{\text{XCLKS\_DT} + \text{XCLKS\_ST} + \text{XCLKH\_DT} + \text{XCLKH\_ST}}{2}\right)\right)}$$

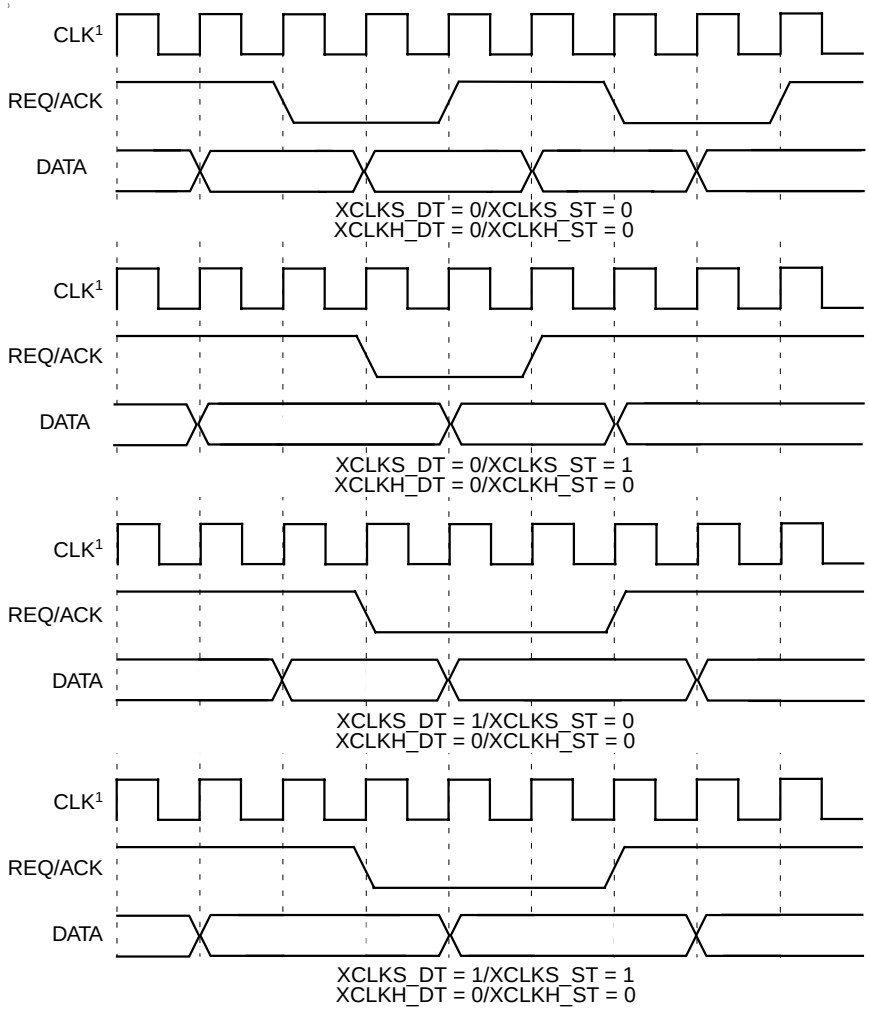
$$\text{Send Rate (ST)} = \frac{\text{Input Clock Rate}}{(\text{SCFDivisor} \times (4 + \text{XCLKS\_ST} + \text{XCLKH\_ST}))}$$

**Figure 4.1 Single Transition Transfer Waveforms**



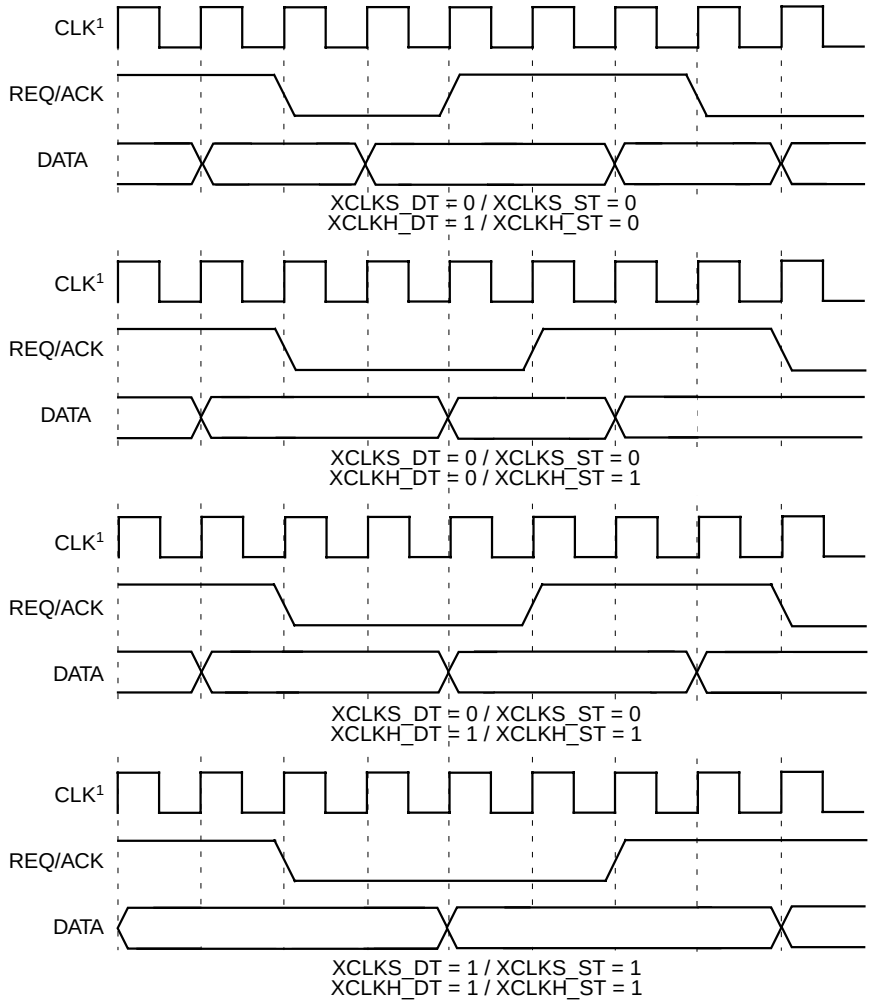

1. CLK = SCLK/SCF Divisor

**Figure 4.2 Double Transition Transfer Waveforms (XCLKS Examples)**




1. CLK = SCLK/SCF Divisor

**Figure 4.3 Double Transition Transfer Waveforms (XCLKH Examples)**




1. CLK = SCLK/SCF Divisor

**Table 4.4 Double Transition Transfer Rates**

| Clock (MHz) | Divisor | Number Xclk's[1] | Base Period (ns) | Receive Rate (Megatransfers/s) | Send Rate (Megatransfers/s) |
|---|---|---|---|---|---|
| 160 | 1 | 0 | 6.25 | 80.00 | 80.00 |
| 160 | 1 | 1 | 6.25 | 80.00 | 64.00 |
| 160 | 1 | 2 | 6.25 | 80.00 | 53.33 |
| 160 | 1 | 3 | 6.25 | 80.00 | 45.71 |
| 160 | 1 | 4 | 6.25 | 80.00 | 40.00 |
| 160 | 1.5 | 0 | 9.38 | 53.33 | 53.33 |
| 160 | 1.5 | 1 | 9.38 | 53.33 | 42.67 |
| 160 | 1.5 | 2 | 9.38 | 53.33 | 35.56 |
| 160 | 1.5 | 3 | 9.38 | 53.33 | 30.48 |
| 160 | 1.5 | 4 | 9.38 | 53.33 | 26.67 |
| 160 | 2 | 0 | 12.50 | 40.00 | 40.00 |
| 160 | 2 | 1 | 12.50 | 40.00 | 32.00 |
| 160 | 2 | 2 | 12.50 | 40.00 | 26.67 |
| 160 | 2 | 3 | 12.50 | 40.00 | 22.86 |
| 160 | 2 | 4 | 12.50 | 40.00 | 20.00 |
| 160 | 3 | 0 | 18.75 | 26.67 | 26.67 |
| 160 | 3 | 1 | 18.75 | 26.67 | 21.33 |
| 160 | 3 | 2 | 18.75 | 26.67 | 17.78 |
| 160 | 3 | 3 | 18.75 | 26.67 | 15.24 |
| 160 | 3 | 4 | 18.75 | 26.67 | 13.33 |
| 160 | 4 | 0 | 25.00 | 20.00 | 20.00 |
| 160 | 4 | 1 | 25.00 | 20.00 | 16.00 |
| 160 | 4 | 2 | 25.00 | 20.00 | 13.33 |
| 160 | 4 | 3 | 25.00 | 20.00 | 11.43 |
| 160 | 4 | 4 | 25.00 | 20.00 | 10.00 |
| 160 | 6 | 0 | 37.50 | 13.33 | 13.33 |
| 160 | 6 | 1 | 37.50 | 13.33 | 10.67 |
| 160 | 6 | 2 | 37.50 | 13.33 | 8.89 |
| 160 | 6 | 3 | 37.50 | 13.33 | 7.62 |
| 160 | 6 | 4 | 37.50 | 13.33 | 6.67 |
| 160 | 8 | 0 | 50.00 | 10.00 | 10.00 |
| 160 | 8 | 1 | 50.00 | 10.00 | 8.00 |
| 160 | 8 | 2 | 50.00 | 10.00 | 6.67 |
| 160 | 8 | 3 | 50.00 | 10.00 | 5.71 |
| 160 | 8 | 4 | 50.00 | 10.00 | 5.00 |
| 40 | 1 | 0 | 25.00 | 20.00 | 20.00 |
| 40 | 1 | 1 | 25.00 | 20.00 | 16.00 |
| 40 | 1 | 2 | 25.00 | 20.00 | 13.33 |
| 40 | 1 | 3 | 25.00 | 20.00 | 11.43 |
| 40 | 1 | 4 | 25.00 | 20.00 | 10.00 |

**Table 4.4 Double Transition Transfer Rates (Cont.)**

| Clock (MHz) | Divisor | Number Xclk's[1] | Base Period (ns) | Receive Rate (Megatransfers/s) | Send Rate (Megatransfers/s) |
|---|---|---|---|---|---|
| 40 | 1.5 | 0 | 37.50 | 13.33 | 13.33 |
| 40 | 1.5 | 1 | 37.50 | 13.33 | 10.67 |
| 40 | 1.5 | 2 | 37.50 | 13.33 | 8.89 |
| 40 | 1.5 | 3 | 37.50 | 13.33 | 7.62 |
| 40 | 1.5 | 4 | 37.50 | 13.33 | 6.67 |
| 40 | 2 | 0 | 50.00 | 10.00 | 10.00 |
| 40 | 2 | 1 | 50.00 | 10.00 | 8.00 |
| 40 | 2 | 2 | 50.00 | 10.00 | 6.67 |
| 40 | 2 | 3 | 50.00 | 10.00 | 5.71 |
| 40 | 2 | 4 | 50.00 | 10.00 | 5.00 |
| 40 | 3 | 0 | 75.00 | 6.67 | 6.67 |
| 40 | 3 | 1 | 75.00 | 6.67 | 5.33 |
| 40 | 3 | 2 | 75.00 | 6.67 | 4.44 |
| 40 | 3 | 3 | 75.00 | 6.67 | 3.81 |
| 40 | 3 | 4 | 75.00 | 6.67 | 3.33 |
| 40 | 4 | 0 | 100.00 | 5.00 | 5.00 |
| 40 | 4 | 1 | 100.00 | 5.00 | 4.00 |
| 40 | 4 | 2 | 100.00 | 5.00 | 3.33 |
| 40 | 4 | 3 | 100.00 | 5.00 | 2.86 |
| 40 | 4 | 4 | 100.00 | 5.00 | 2.50 |
| 40 | 8 | 0 | 200.00 | 2.50 | 2.50 |
| 40 | 8 | 1 | 200.00 | 2.50 | 2.00 |
| 40 | 8 | 2 | 200.00 | 2.50 | 1.67 |
| 40 | 8 | 3 | 200.00 | 2.50 | 1.43 |
| 40 | 8 | 4 | 200.00 | 2.50 | 1.25 |

1. Number Xclk's = XCLKS_DT + XCLKS_ST + XCLKH_DT + XCLKH_ST

**Table 4.5 Single Transition Transfer Rates**

| Clock (MHz) | Divisor | Number Xclk's[1] | Base Period (ns) | Receive Rate (Megatransfers/s) | Send Rate (Megatransfers/s) |
|---|---|---|---|---|---|
| 160 | 1 | 0 | 6.25 | 40.00 | 40.00 |
| 160 | 1 | 1 | 6.25 | 40.00 | 32.00 |
| 160 | 1 | 2 | 6.25 | 40.00 | 26.67 |
| 160 | 1.5 | 0 | 9.38 | 26.67 | 26.67 |
| 160 | 1.5 | 1 | 9.38 | 26.67 | 21.33 |
| 160 | 1.5 | 2 | 9.38 | 26.67 | 17.78 |
| 160 | 2 | 0 | 12.50 | 20.00 | 20.00 |
| 160 | 2 | 1 | 12.50 | 20.00 | 16.00 |
| 160 | 2 | 2 | 12.50 | 20.00 | 13.33 |
| 160 | 3 | 0 | 18.75 | 13.33 | 13.33 |
| 160 | 3 | 1 | 18.75 | 13.33 | 10.67 |
| 160 | 3 | 2 | 18.75 | 13.33 | 8.89 |
| 160 | 4 | 0 | 25.00 | 10.00 | 10.00 |
| 160 | 4 | 1 | 25.00 | 10.00 | 8.00 |
| 160 | 4 | 2 | 25.00 | 10.00 | 6.67 |
| 160 | 6 | 0 | 37.50 | 6.67 | 6.67 |
| 160 | 6 | 1 | 37.50 | 6.67 | 5.33 |
| 160 | 6 | 2 | 37.50 | 6.67 | 4.44 |
| 160 | 8 | 0 | 50.00 | 5.00 | 5.00 |
| 160 | 8 | 1 | 50.00 | 5.00 | 4.00 |
| 160 | 8 | 2 | 50.00 | 5.00 | 3.33 |
| 40 | 1 | 0 | 25.00 | 10.00 | 10.00 |
| 40 | 1 | 1 | 25.00 | 10.00 | 8.00 |
| 40 | 1 | 2 | 25.00 | 10.00 | 6.67 |
| 40 | 1.5 | 0 | 37.50 | 6.67 | 6.67 |
| 40 | 1.5 | 1 | 37.50 | 6.67 | 5.33 |
| 40 | 1.5 | 2 | 37.50 | 6.67 | 4.44 |
| 40 | 2 | 0 | 50.00 | 5.00 | 5.00 |
| 40 | 2 | 1 | 50.00 | 5.00 | 4.00 |
| 40 | 2 | 2 | 50.00 | 5.00 | 3.33 |
| 40 | 3 | 0 | 75.00 | 3.33 | 3.33 |
| 40 | 3 | 1 | 75.00 | 3.33 | 2.67 |
| 40 | 3 | 2 | 75.00 | 3.33 | 2.22 |
| 40 | 4 | 0 | 100.00 | 2.50 | 2.50 |
| 40 | 4 | 1 | 100.00 | 2.50 | 2.00 |
| 40 | 4 | 2 | 100.00 | 2.50 | 1.67 |

**Table 4.5 Single Transition Transfer Rates (Cont.)**

| Clock (MHz) | Divisor | Number Xclk's[1] | Base Period (ns) | Receive Rate (Megatransfers/s) | Send Rate (Megatransfers/s) |
|---|---|---|---|---|---|
| 40 | 6 | 0 | 150.00 | 1.67 | 1.67 |
| 40 | 6 | 1 | 150.00 | 1.67 | 1.33 |
| 40 | 6 | 2 | 150.00 | 1.67 | 1.11 |
| 40 | 8 | 0 | 200.00 | 1.25 | 1.25 |
| 40 | 8 | 1 | 200.00 | 1.25 | 1.00 |
| 40 | 8 | 2 | 200.00 | 1.25 | 0.83 |

1. Number Xclk's = XCLKS_ST + XCLKH_ST

## Register: 0xBD

**Reserved**

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| R | | | | | | | |
| x | x | x | x | x | x | x | x |

This register is reserved.

## Register: 0xBE

**AIP Control Zero (AIPCNTL0)**

**Read/Write**

| 7 | 6 | 5 | 4 | 3 | | | 0 |
|---|---|---|---|---|---|---|---|
| FBAIP | RSQAIP | SEQAIP | | R | | | |
| 0 | 0 | 0 | | 0 | 0 | 0 | 0 |

**FBAIP** **Force Bad AIP value** **7**
Setting this bit causes bad AIP values to be sent over the SCSI bus.

**RSQAIP** **Reset AIP Sequence value** **6**
Setting this bit causes the sequence value used in the calculation of the protection code to reset.

**SEQAIP** **AIP Sequence value** **[5:4]**
These two bits contain the current AIP sequence value. The SEQAIP bits are read only and should only be accessed for diagnostics purposes. The sequence value is automatically reset on every phase change.

**R** **Reserved** **[3:0]**

## Register: 0xBF

**AIP Control One (AIPCNTL1)**
**Read/Write**

| 7 | 6 | 5 | | | | | 0 |
|---|---|---|---|---|---|---|---|
| AIPERR | LAIPERR | AIPV | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**AIPERR** **AIP Error Status** **7**
This bit represents the live value of the error status for the AIP checking logic. This bit may indicate false errors and should not be used except for diagnostics purposes.

**LAIPERR** **Latched AIP Error Status** **6**
This bit represents the latched version of the error status for the AIP checking logic. This bit accurately reflects the fact that an AIP error was detected. This bit will be cleared when the Parity/CRC/AIP error bit is cleared in the SCSI Interrupt Status Zero (SIST0) register.

**AIPV** **AIP Value** **[5:0]**
This value represents the current calculated value of the protection code. This value is based off of the eight SCSI data bits, the three phase signals, the two SCSI Reserved signals, and the Sequence value.

## Registers: 0xC0–0xC3

**Phase Mismatch Jump Address One (PMJAD1)**
**Read/Write**

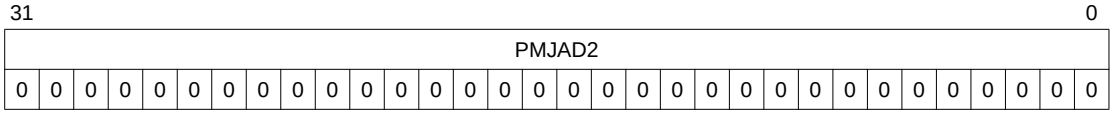
| 31 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PMJAD1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**PMJAD1** **Phase Mismatch Jump Address One** **[31:0]**
This register contains the 32-bit address that is jumped to upon a phase mismatch. Depending upon the state of the PMJCTL bit, this address is either used during an outbound (Data-Out, Command, Message-Out) phase mismatch (PMJCTL = 0) or when the WSR bit is cleared (PMJCTL = 1). This register is loaded with the address of

a SCRIPTS routine that updates the memory data structures of the BMOV that was executing when the phase mismatch occurred.

## Registers: 0xC4–0xC7

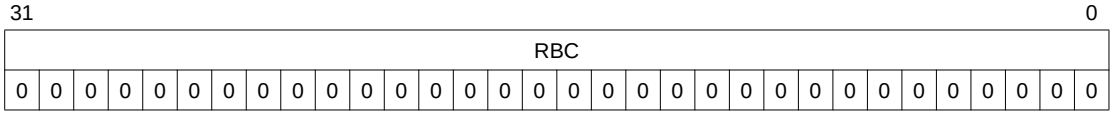
**Phase Mismatch Jump Address Two (PMJAD2)**
**Read/Write**




**PMJAD2** **Phase Mismatch Jump Address Two** **[31:0]**
This register contains the 32-bit address that is jumped to upon a phase mismatch. Depending upon the state of the PMJCTL bit, this address is either used during an inbound (Data-In, Status, Message-In) phase mismatch (PMJCTL = 0) or when the WSR bit is set (PMJCTL = 1). This register is loaded with the address of a SCRIPTS routine that updates the memory data structures of the BMOV that was executing when the phase mismatch occurred.

## Registers: 0xC8–0xCB

**Remaining Byte Count (RBC)**
**Read/Write**




**RBC** **Remaining Byte Count** **[31:0]**
This register contains the byte count that remains for the BMOV that was executing when the phase mismatch occurred. In the case of Direct or Indirect BMOV instructions, the upper byte of this register also contains the opcode of the BMOV that was executing. In the case of a Table Indirect BMOV instruction, the upper byte contains the upper byte of the Table Indirect entry that was fetched.

In the case of a SCSI data receive, this byte count reflects all data received from the SCSI bus, including

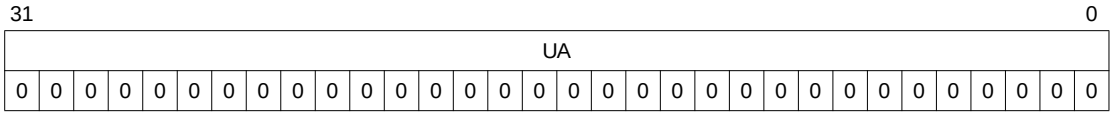
any byte in SCSI Wide Residue (SWIDE). There is no data remaining in the part that must be flushed to memory with the exception of a possible byte in the SWIDE register. That byte must be flushed to memory manually in SCRIPTS.

In the case of a SCSI data send, this byte count reflects all data sent out onto the SCSI bus. Any data left in the part from the phase mismatch is ignored and automatically cleared from the FIFOs.

## Registers: 0xCC–0xCF

### Updated Address (UA)

**Read/Write**




**UA** **Updated Address** **[31:0]**

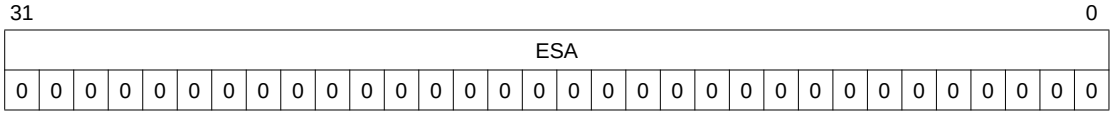
This register contains the updated data address for the BMOV that was executing when the phase mismatch occurred.

In the case of a SCSI data receive, if there is a byte in the SCSI Wide Residue (SWIDE) register then this address points to the location where that byte must be stored. The SWIDE byte must be manually written to memory and this address must be incremented prior to updating any scatter/gather entry.

In the case of a SCSI data receive, if there is not a byte in the SWIDE register then this address is the next location that should be written to when this I/O restarts. No manual flushing will be necessary.

In the case of a SCSI data send, all data sent to the SCSI bus will be accounted for and any data left in the part is ignored and is automatically cleared from the FIFOs.

## Registers: 0xD0–0xD3
### Entry Storage Address (ESA)
### Read/Write




**ESA** **Entry Storage Address** **[31:0]**

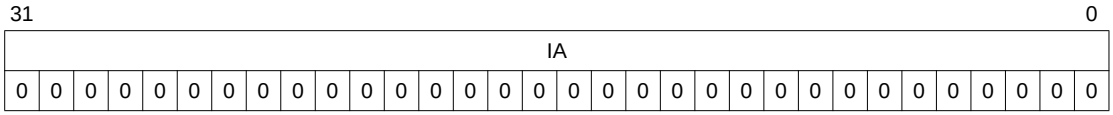
This register's value depends on the type of BMOV being executed. The three types of BMOVs are:

**Direct BMOV:** In the case of a direct BMOV, this register contains the address the BMOV was fetched from when the phase mismatch occurred.

**Indirect BMOV:** In the case of an indirect BMOV, this register contains the address the BMOV was fetched from when the phase mismatch occurred.

**Table Indirect BMOV:** In the case of a Table Indirect BMOV, this register contains the address of the Table Indirect entry being used when the phase mismatch occurred.

## Registers: 0xD4–0xD7
### Instruction Address (IA)
### Read/Write




**IA** **Instruction Address** **[31:0]**

This register always contains the address of the BMOV instruction that was executing when the phase mismatch occurred. This value will always match the value in the Entry Storage Address (ESA) except in the case of a Table Indirect BMOV in which case the ESA will have the address of the Table Indirect entry and this register points to the address of the BMOV instruction.

## Registers: 0xD8–0xDA
### SCSI Byte Count (SBC)
### Read Only

| 23 | | | | | | | | | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SBC | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**SBC** **SCSI Byte Count** **[23:0]**

This register contains the count of the number of bytes transferred to or from the SCSI bus during any given BMOV. This value is used in calculating the information placed into the Remaining Byte Count (RBC) and Updated Address (UA) registers and should not need to be used in normal operations.

There are several conditions for which the byte count does not match the number of bytes transferred. If a BMOV transfers an odd number of bytes across a wide bus, the byte count at the end of the BMOV is one byte greater than the number of bytes sent. This also occurs in an odd byte count wide receive case. Lastly, when a wide send occurs and a chain byte from a previous transfer is present, the byte count does not reflect the chain byte sent across the bus during that BMOV. To determine the correct address to start fetching data from after a phase mismatch, this byte is not counted for this BMOV. It is included in the previous BMOV's byte count.

## Register: 0xDB
### Reserved

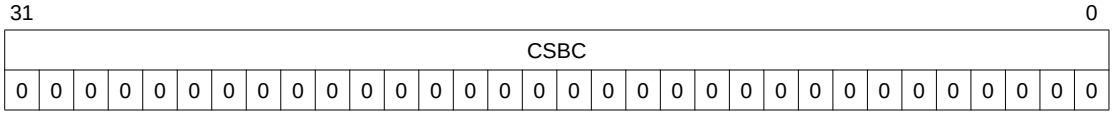
| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| R | | | | | | | |
| x | x | x | x | x | x | x | x |

This register is reserved.

## Registers: 0xDC–0xDF

### Cumulative SCSI Byte Count (CSBC)

### Read/Write



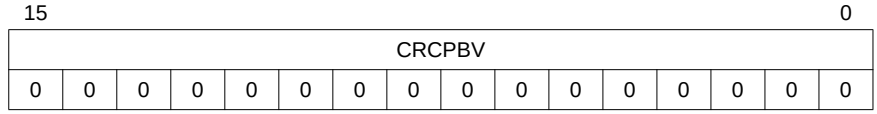

**CSBC** **Cumulative SCSI Byte Count** **[31:0]**

This loadable register contains a cumulative count of the number of bytes transferred across the SCSI bus during data phases. It does not count bytes sent in command, status, Message-In or Message-Out phases. It counts bytes as long as the phase mismatch enable bit (ENPMJ) in the Chip Control Zero (CCNTL0) register is set. Unlike the SCSI Byte Count (SBC), this count is not cleared on each BMOV instruction but continues to count across multiple BMOV instructions. This register can be loaded with any arbitrary start value.

## Registers: 0xE0–0xE1

### CRC Pad Byte Value (CRCPAD)

### Read/Write




**CRCPBV** **CRC Pad Byte Value** **[15:0]**

This register contains the value placed onto the bus for the CRC pad bytes.

## Register: 0xE2

**CRC Control Zero (CRCCNTL0)**
**Read/Write**

| 7 | 6 | 5 | 4 | 3 | | | 0 |
|---|---|---|---|---|---|---|---|
| DCRCC | DCRCPC | RCRCIC | R | CRCRI[3:0] | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**DCRCC** **Disable CRC Checking** **7**
Setting this bit causes the internal logic not to check or report CRC errors during Ultra3 transfers. The SYM53C1010 continues to calculate and send CRCs as requested by the target according to the SPI-3 specification.

**DCRCPC** **Disable CRC Protocol Checking** **6**
Setting this bit causes the internal logic to not check or report CRC protocol errors during Ultra3 transfers. The SYM53C1010 continues to calculate and send CRCs as requested by the target according to the SPI-3 specification but does not set a SGE interrupt if a CRC protocol error occurs. This bit should not be set in normal operation.

**RCRCIC** **Reset CRC Interval Counter (Target Mode Only)** **5**
Setting this bit resets the internal CRC interval counter to zero. This bit is not self-clearing.

**R** **Reserved** **4**

**CRCRI[3:0]** **CRC Request Interval (Target Mode Only)** **[3:0]**
These bits determine when a CRC request is to be sent by the device when operating in the target mode and transferring data in the DT Data-In or DT Data-Out phases. The interval is independent of individual block moves, allowing consistent CRC requests across multiple block moves of varying byte counts as in scatter/gather types of operations. The following table defines the valid CRC request intervals. A setting of zero will disable the automatic CRC request interval and the device will not request a CRC at any time. The SPI-3 specification states that all DT Data-In and DT Data-Out phases must end with a CRC transfer. Thus, to maintain compliance with the SPI-3 specification it is necessary to manually

request a CRC by executing a force CRC Block Move instruction.

| CRCRI | Interval (Bytes) |
|---|---|
| 0x0 | Disabled |
| 0x1 | 128 |
| 0x2 | 256 |
| 0x3 | 512 |
| 0x4 | 1024 |
| 0x5 | 2048 |
| 0x6 | 4096 |
| 0x7 | 8192 |
| 0x8 | 16384 |
| 0x9 | 32768 |
| 0xA | 65536 |
| 0xB | Reserved |
| 0xC | Reserved |
| 0xD | Reserved |
| 0xE | Reserved |
| 0xF | Reserved |

## Register: 0xE3

### CRC Control One (CRCCNTL1)

### Read/Write

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CRCERR | R | ENAS | TSTSD | TSTCHK | TSTADD | CRCDSEL | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**CRCERR** **CRC Error** **7**

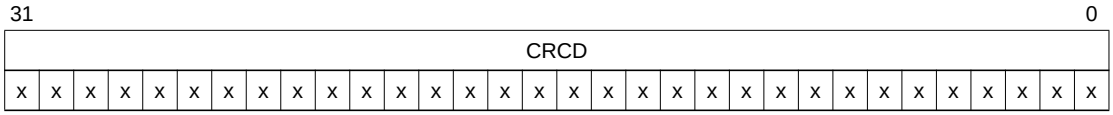
This bit indicates whether or not a CRC error has been detected during a DT Data-In SCSI transfer. This bit is set independent of the DISCRCCHK bit. To clear this condition, either write this bit to a 1 or read the SCSI Interrupt Status Zero (SIST0) and SCSI Interrupt Status One (SIST1) registers. When CRC Checking and the Parity/CRC/AIP Error Interrupt are enabled, this error condition is also indicated as a Parity/CRC/AIP error (bit 0 of the SIST0 register).

| | | |
|---|---|---|
| **R** | **Reserved** | **6** |
| **ENAS** | **Enable CRC Auto Seed**<br>Setting this bit causes the CRC logic to automatically reseed after every CRC check performed during DT Data-In SCSI transfers. When this bit is cleared, the SCSI control logic controls when the reseeding occurs. | **5** |
| **TSTSD** | **Test CRC Seed**<br>Setting this bit causes the CRC logic to immediately reseed itself. This bit should never be set during normal operation as it may cause corrupt CRCs to be generated. | **4** |
| **TSTCHK** | **Test CRC Check**<br>Setting this bit causes the CRC logic to initiate a CRC check. This bit should never be set during normal operation as it results in spurious CRC errors. | **3** |
| **TSTADD** | **Test CRC Accumulate**<br>Setting this bit causes the CRC block to take the value in its input register and add it into the current CRC calculation, resulting in a new output CRC value. This bit should not be set during normal operation as it results in corrupt CRC values. | **2** |
| **CRCDSEL** | **CRC Data Register Selector**<br>These bits control the data that is visible in the CRC Data (CRCD) register. | **[1:0]** |

## Registers: 0xE4–0xE7

**CRC Data (CRCD)**

**Read/Write**




The value in this register is dependent on the setting of the CRCDSEL bits in the CRC Control One (CRCCNTL1) register.

Note: Data written to this register may not be available for immediate read back due to synchronization between the PCI and SCSI clock domains. After a write, wait at least 16 PCI clock cycles before reading this register.

**CRCD** **CRC Data** **[31:0]**

If CRCDSEL = 0b00, this register represents the current CRC value. After sending data during the DT Data-Out phase, this register contains the CRC calculation for that data, if no CRC request occurred during the transfer. In this mode, this register is read only.

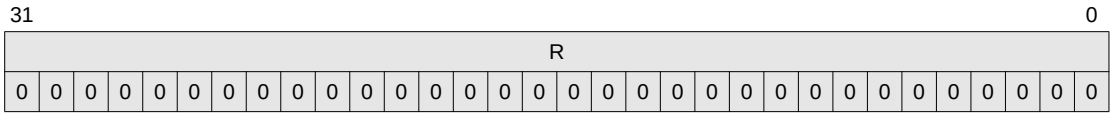
If CRCDSEL = 0b01, this register represents the CRC Input register and contains its current value. It normally contains the SCSI data transferred to or from the SCSI bus during a DT transfer phase. In this mode, this register can be written to in order to manually alter the input data used for CRC calculation. For normal operations, this register should never be written to.

If CRCDSEL = 0b10, this register represents the CRC Accumulator and contains its current value. In this mode, this register can be written to in order to manually modify the value in the accumulator. This register should not be written to during normal operation as corrupt CRC values result.

If CRCDSEL = 0b11, this register contains the saved bad CRC value that was calculated when a CRC error was detected. After a CRC error is detected, this register is not overwritten until the error condition is cleared.

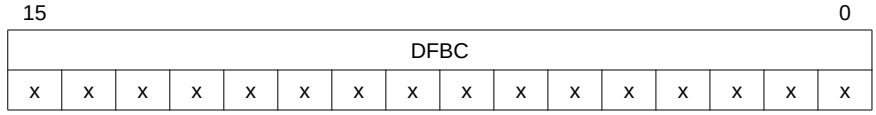
## Registers: 0xE8–0xEF

**Reserved**



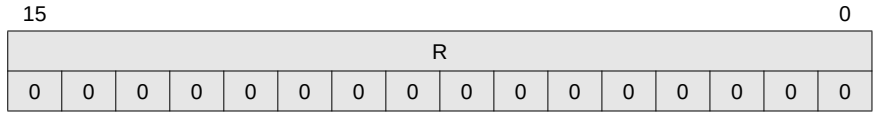

This register is reserved.

## Registers: 0xF0–0xF1
### DMA FIFO Byte Count (DFBC)
### Read Only



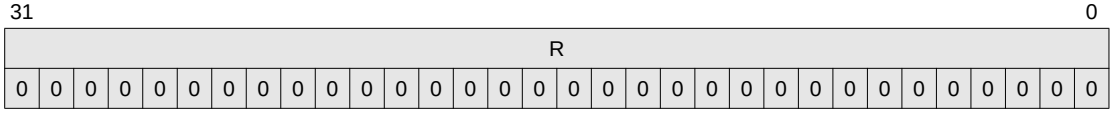

**DFBC** **DMA FIFO Byte Count** **[15:0]**
This 16-bit read only register contains the actual number of bytes contained in the DMA FIFO. This register is not stable while data is actually being transferred. This register can be used during error recovery.

## Registers: 0xF2–0xF3
### Reserved




This register is reserved.

## Registers: 0xF4–0xFF
### Reserved




This register is reserved.

## 4.3 SCSI Shadow Registers

Note: For more information concerning shadow registers, refer to the Chip Test Four (CTEST4), Scratch Register A (SCRATCHA), Scratch Register B (SCRATCHB), Memory Move Read Selector (MMRS), Memory Move Write Selector (MMWS), and SCRIPT Fetch Selector (SFS) register descriptions.

### Registers: 0x34–0x37
**Shadowed Scratch Register A (SCRATCHA)**
**Read/Write**

| 31 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SCRATCHA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**SCRATCHA** **Scratch Register A** **[31:0]**

### Register: 0x42
**Shadowed SCSI SGE Status 0**
**Read/Write**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SRP | DFP | RD | PCO | OO | OU | DO | DU |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

This register contains the individual status bits which cause a SGE SCSI interrupt. These bits correspond to the SGE conditions described in the SCSI Interrupt Status Zero (SIST0) register description. Unlike the other registers in the device, these bits must be set to one to clear the condition. This register is shadowed behind the SIST registers. Setting bit 4 (SRTM) in the Chip Test Four (CTEST4) register, enables access to this register.

**SRP** **SCRIPTS RAM Parity** **7**

**DFP** **DMA FIFO Parity** **6**

**RD** **Residual Data in SCSI FIFO** **5**

| | | |
|---|---|---|
| **PCO** | **Phase Change with Outstanding Offset** | **4** |
| **OO** | **Offset Overflow** | **3** |
| **OU** | **Offset Underflow** | **2** |
| **DO** | **Data Overflow** | **1** |
| **DU** | **Data Underflow** | **0** |

## Register: 0x43

**Shadowed SCSI Interrupt Status One (SIST1)**
**Read Only**

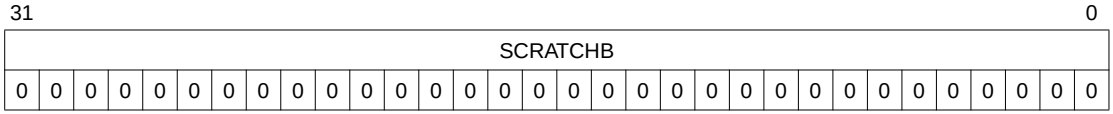
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| R | | PNCRC | FCRC | DTST | NFCRC | MCRC | R |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

This register contains the individual status bits which cause a SGE SCSI interrupt. These bits correspond to the SGE conditions described in the SCSI Interrupt Status Zero (SIST0) register description. Unlike the other registers in the device, these bits must be set to one to clear the condition. This register is shadowed behind the SIST registers. Setting bit 4 (SRTM) in the Chip Test Four (CTEST4) register, enables access to this register.

| | | |
|---|---|---|
| **R** | **Reserved** | **[7:6]** |
| **PNCRC** | **Pad Request with no CRC Request Following** | **5** |
| **FCRC** | **Force CRC** | **4** |
| **DTST** | **Switch from DT to ST Timings During a Transfer** | **3** |
| **NFCRC** | **Phase Change with No Final CRC Request** | **2** |
| **MCRC** | **Multiple CRC Requests with the Same Offset** | **1** |
| **R** | **Reserved** | **0** |

## Registers: 0x5C–0x5F

**Shadowed Scratch Register B (SCRATCHB)**
**Read/Write**




**SCRATCHB Scratch Register B [31:0]**

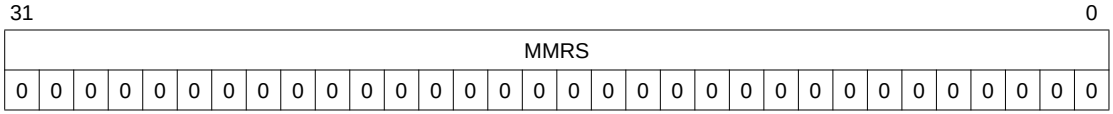
## Registers: 0x60–0x9F

**Shadowed Scratch Registers C–R (SCRATCHC–SCRATCHR)**
**Read/Write**

These are general purpose user-definable shadows of the scratch pad registers.

## Registers: 0xA0–0xA3

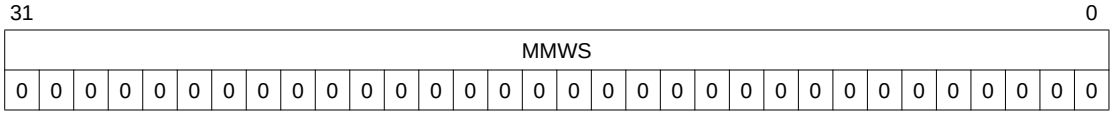
**Shadowed Memory Move Read Selector (MMRS)**
**Read/Write**




**MMRS Shadowed Memory Move Read Selector [31:0]**

## Registers: 0xA4–0xA7

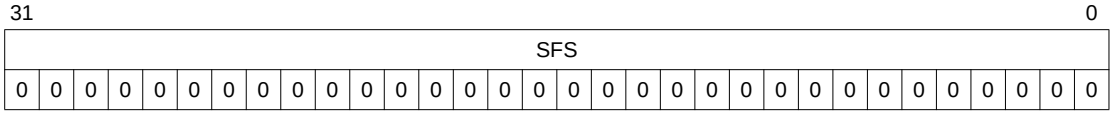
**Shadowed Memory Move Write Selector (MMWS)**
**Read/Write**




**MMWS** **Shadowed Memory Move Write Selector** **[31:0]**

## Registers: 0xA8–0xAB

**Shadowed SCRIPT Fetch Selector (SFS)**
**Read/Write**




**SFS** **Shadowed SCRIPT Fetch Selector** **[31:0]**

# Chapter 5
# SCSI SCRIPTS Instruction Set

After power-up and initialization, the SYM53C1010-33 can operate in the low level register interface mode, or use SCSI SCRIPTS.

With the low level register interface, the user has access to the DMA control logic and the SCSI bus control logic. An external processor has access to the SCSI bus signals and the low level DMA signals, which allow creation of complicated board level test algorithms. The low level interface is useful for backward compatibility with SCSI devices that require certain unique timings or bus sequences to operate properly.

The following sections describe the benefits and use of SCSI SCRIPTS.



## 5.1 SCSI SCRIPTS

To operate in the SCSI SCRIPTS mode, the SYM53C1010-33 requires only a SCRIPTS start address. The start address must be at a Dword (four byte) boundary. This aligns all the following SCRIPTS at a Dword boundary since all SCRIPTS are 8 or 12 bytes long. Instructions are fetched until an interrupt instruction is encountered, or until an unexpected event (such as a hardware error) causes an interrupt to the external processor.

Once an interrupt is generated, the SYM53C1010-33 halts all operations until the interrupt is serviced. Then, the start address of the next SCRIPTS instruction is written to the DMA SCRIPTS Pointer (DSP) register to restart the automatic fetching and execution of instructions.

In the SCSI SCRIPTS mode the SYM53C1010-33 is allowed to make decisions based on the status of the SCSI bus, which frees the microprocessor from servicing the numerous interrupts inherent in I/O operations.

Given the rich set of SCSI-oriented features included in the instruction set, and the ability to re-enter the SCSI algorithm at any point, this high level interface is all that is required for both normal and exception conditions. Switching to the low level mode for error recovery is not required.

The following types of SCRIPTS instructions are implemented in the SYM53C1010-33:

**Block Move** – used to move data between the SCSI bus and memory.

**I/O or Read/Write** – causes the SYM53C1010-33 to trigger common SCSI hardware sequences, or to move registers.

**Transfer Control** – allows SCRIPTS instructions to make decisions based on real time SCSI bus conditions.

**Memory Move** – causes the SYM53C1010-33 to execute block moves between different parts of main memory.

**Load/Store** – provides a more efficient way to move data to/from memory from/to an internal register in the chip without using the Memory Move instruction.

Each instruction consists of two or three 32-bit words. The first 32-bit word is always loaded into the DMA Command (DCMD) and DMA Byte Counter (DBC) registers, the second into the DMA SCRIPTS Pointer Save (DSPS) register. The third word, used only by Memory Move instructions, is loaded into the Temporary (TEMP) shadow register. In an indirect I/O or Move instruction, the first two 32-bit opcode fetches are followed by one or two more 32-bit fetch cycles.

### 5.1.1 Sample Operation

The following example describes execution of a SCRIPTS Block Move instruction.

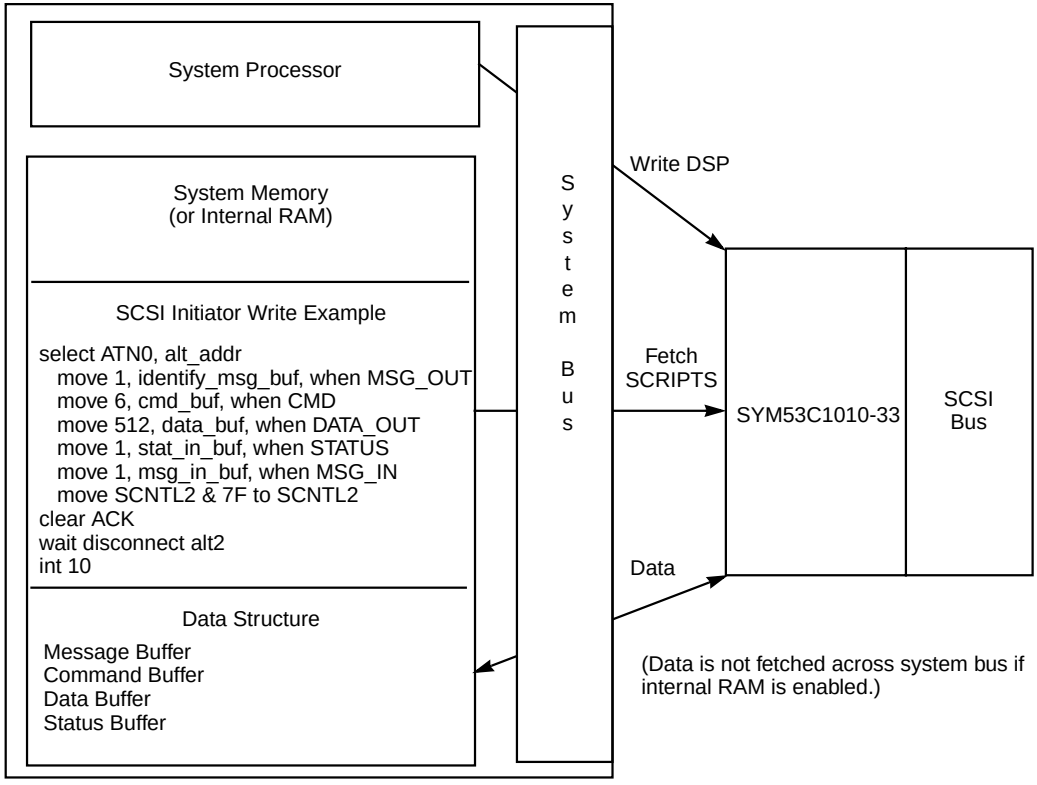
- The host CPU, through programmed I/O, gives the DMA SCRIPTS Pointer (DSP) register (in the Operating register file) the starting address in main memory that points to a SCSI SCRIPTS program for execution.
- Loading the DMA SCRIPTS Pointer (DSP) register causes the SYM53C1010-33 to fetch its first instruction at the address just loaded. This fetch is from main memory or the internal RAM, depending on the address.
- The SYM53C1010-33 typically fetches two Dwords (64 bits) and decodes the high-order byte of the first Dword as a SCRIPTS instruction. If the instruction is a Block Move, the lower three bytes of the first Dword are stored and interpreted as the number of bytes to move. The second Dword is stored and interpreted as the 32-bit beginning address in main memory to which the move is directed.
- For a SCSI send operation, the SYM53C1010-33 waits until there is enough space in the DMA FIFO to transfer a programmable size block of data. For a SCSI receive operation, it waits until enough data is collected in the DMA FIFO for transfer to memory. At this point, the SYM53C1010-33 requests use of the PCI bus again to transfer the data.
- When the SYM53C1010-33 is granted the PCI bus, it executes (as a bus master) a burst transfer (programmable size) of data, decrements the internally stored remaining byte count, increments the address pointer, and then releases the PCI bus. The SYM53C1010-33 stays off the PCI bus until the FIFO can again hold (for a write) or has collected (for a read) enough data to repeat the process.

The process repeats until the internally stored byte count has reached zero. The SYM53C1010-33 releases the PCI bus and then performs another SCRIPTS instruction fetch cycle, using the incremented stored address maintained in the DMA SCRIPTS Pointer (DSP) register. Execution of SCRIPTS instructions continues until an error condition occurs or an interrupt SCRIPTS instruction is received. At this point, the SYM53C1010-33 interrupts the host CPU and waits for further servicing

by the host system. It can execute independent Block Move instructions specifying new byte counts and starting locations in main memory. In this manner, the SYM53C1010-33 performs scatter/gather operations on data without requiring help from the host program, generating a host interrupt, or programming of an external DMA controller.

Figure 5.1 provides an overview of SCRIPTS operation.

**Figure 5.1 SCRIPTS Overview**

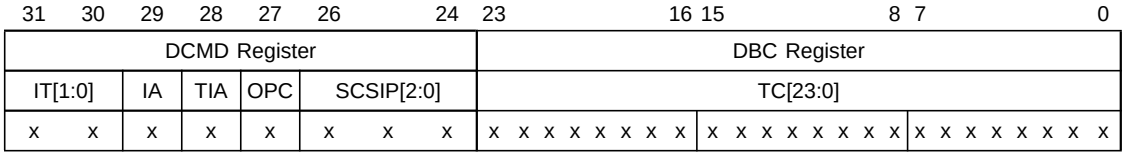


## 5.2 Block Move Instructions

For Block Move instructions, bits 5 and 4 (SIOM and DIOM) in the DMA Mode (DMODE) register determine whether the source/destination address resides in memory or I/O space. When data is moved onto the SCSI bus, SIOM controls whether that data comes from the I/O or memory space. When data is moved off of the SCSI bus, DIOM controls whether that data goes to the I/O or memory space.

### 5.2.1 First Dword

**Figure 5.2 Block Move Instruction - First Dword**




**IT[1:0]** **Instruction Type - Block Move** **[31:30]**
The configuration of these two bits define the SCRIPTS Instruction Type. The Block Move Instruction is bit configuration 0, 0.

**IA** **Indirect Addressing** **29**
When this bit is cleared, user data is moved to or from the 32-bit data start address for the Block Move instruction. The value is loaded into the chip's address register and incremented as data is transferred. The address of the data to move is in the second Dword of this instruction.

When the EN64DBMV bit in Chip Control One (CCNTL1) is set, a third Dword is fetched to provide the upper Dword of a 64-bit address. The upper Dword address is fetched along with the instruction and loaded into the Dynamic Block Move Selector (DBMS) register.

If the EN64DBMV bit is cleared, the upper Dword address is copied from the Static Block Move Selector (SBMS) register.

### Direct Addressing

The byte count and absolute address are as follows:

| Command | Byte Count |
|---|---|
| Lower Dword Address of Data | |
| Upper Dword address of data (EN64DBMV = 1) | |

### Indirect

When set, the 32-bit user data start address for the Block Move is the address of a pointer to the actual data buffer address. The value at the 32-bit start address is loaded into the chip's DMA Next Address (DNAD) register using a third Dword fetch (4-byte transfer across the host computer bus).

Use the fetched byte count, but fetch the data address from the address in the instruction.

If 64-bit addressing is desired, the upper Dword of the address is stored in the Static Block Move Selector (SBMS) register. When the value in SBMS is 0x0, 32-bit addressing is assumed.

| Command | Byte Count |
|---|---|
| Address of Pointer to Data | |

Once the data pointer address is loaded, it is executed as when the chip operates in the direct mode. This indirect feature allows specification of a table of data buffer addresses. Using the SCSI SCRIPTS compiler, the table offset is placed in the SCRIPT at compile time. Then at the actual data transfer time, the offsets are added to the base address of the data address table by the external processor. The logical I/O driver builds a structure of addresses for an I/O rather than treating each address individually.

Note: Using indirect and table indirect addressing simultaneously is not permitted; use only one addressing method at a time.

**TIA** **Table Indirect** **28**
**32-bit Addressing**

When this bit is set, the 24-bit signed value in the start address of the move is treated as a relative displacement from the value in the Data Structure Address (DSA) register. Both the transfer count and the source/ destination address are fetched from this location.

Use the signed integer offset in bits [23:0] of the second four bytes of the instruction, added to the value in the Data Structure Address (DSA) register, to fetch first the byte count and then the data address. The signed value is combined with the data structure base address to generate the physical address used to fetch values from the data structure. Sign-extended values of all ones for negative values are allowed, but bits [31:24] are ignored.

| Command | Not Used |
|---|---|
| Don't Care | Table Offset |

Note: Using indirect and table indirect addressing simultaneously is not permitted; use only one addressing method at a time.

Prior to the start of an I/O, load the Data Structure Address (DSA) register with the base address of the I/O data structure. Any address on a Dword boundary is allowed.

After a Table Indirect opcode is fetched, the Data Structure Address (DSA) is added to the 24-bit signed offset value from the opcode to generate the address of the required data; both positive and negative offsets are allowed. A subsequent fetch from that address brings the data values into the chip.

For a MOVE instruction, the 24-bit byte count is fetched from system memory. Then the 32-bit physical address is brought into the SYM53C1010-33. Execution of the move begins at this point.

SCRIPTS can directly execute operating system I/O data structures, saving time at the beginning of an I/O operation. The I/O data structure can begin on any Dword boundary and may cross system segment boundaries.

There are two restrictions on the placement of pointer data in system memory:

- The eight bytes of data in the MOVE instruction must be contiguous.
- Indirect data fetches are not available during execution of a Memory-to-Memory DMA operation.

| 00 | Byte Count |
|---|---|
| Physical Data Address | |

### 64-bit Addressing

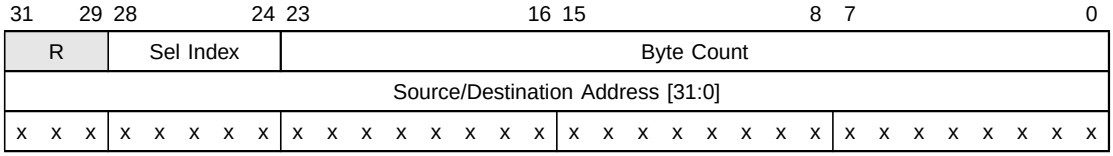
If the Enable 64-bit Table Indirect Block Move (EN64TIBMV) bit is cleared, table indirect block moves remain as two Dword opcodes plus a two Dword table entry. The upper 32 bits of the address are copied from the Static Block Move Selector (SBMS) when performing data transfers during block move operations. The SBMS register must be loaded manually.

If the Enable 64-bit Table Indirect Block Move (EN64TIBMV) bit is set and the 64-bit Table Indirect Index Mode (64TIMOD) bit is cleared, bits [28:24] of the first Dword of the table entry (where the byte count is located) select one of the 16 scratch registers or any of the six 64-bit selector registers as a selector for the upper 32-bit address. Please see the Table Indirect Index mode mapping table for a breakdown of index values and the corresponding registers selected. The selected address is automatically loaded into the DMA Next Address 64 (DNAD64) register.

Note: If EN64TIBMV is set and 64TIMOD is set, bits [31:24] of the first Dword of the table entry (where the byte count is located) are loaded directly into DMA Next Address 64 (DNAD64) to provide a 40-bit address.

The format for the table indirect entries for each mode is shown below. The table for Table Indirect block moves upper 32 bit address locations summarizes the available modes for table indirect block moves.

Index Mode 0 (64TIMOD clear) table entry format:




Index Mode 1 (64TIMOD set) table entry format:

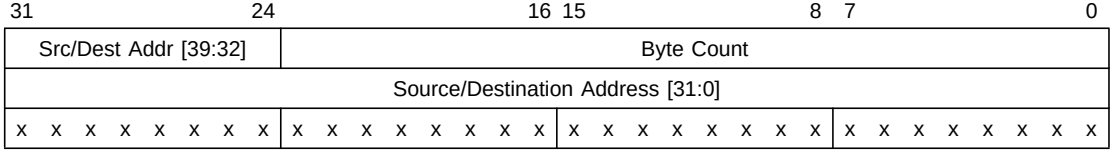



Table Indirect block moves upper 32 bit address locations:

| EN64TIBMV | 64TIMOD | Upper 32-bit Data Address Comes From |
|---|---|---|
| 0 | 0 | SBMS |
| 0 | 1 | SBMS |
| 1 | 0 | ScratchC–J, MMWS, MMRS, SFS, DRS, SBMS, DBMS |
| 1 | 1 | 1st Table Entry Dword bits [31:24] (40-bit addressing only) |

Table Indirect Index mode mapping:

| Index Value | Selector Used |
|---|---|
| 0x00 | Scratch C |
| 0x01 | Scratch D |
| 0x02 | Scratch E |
| 0x03 | Scratch F |
| 0x04 | Scratch G |
| 0x05 | Scratch H |
| 0x06 | Scratch I |
| 0x07 | Scratch J |
| 0x08 | Scratch K |
| 0x09 | Scratch L |
| 0x0A | Scratch M |
| 0x0B | Scratch N |
| 0x0C | Scratch O |
| 0x0D | Scratch P |
| 0x0E | Scratch Q |
| 0x0F | Scratch R |
| 0x10 | MMRS |
| 0x11 | MMWS |
| 0x12 | SFS |
| 0x13 | DRS |
| 0x14 | SBMS |
| 0x15 | SBMS |
| 0x16–0x1F | Illegal (Results in an IID interrupt) |

**OPC** **Opcode** **27**

This 1-bit field defines the instruction to execute as a block move (MOVE).

**Target Mode**

| OPC | Instruction Defined |
|---|---|
| 0 | MOVE/MOVE64 |
| 1 | CHMOV/CHMOV64 |

The SYM53C1010-33 verifies that it is connected to the SCSI bus as a target before executing this instruction.

The SYM53C1010-33 asserts the SCSI phase signals (SMSG/, SC_D/, and SI_O/) as defined by the Phase Field bits in the instruction.

If the instruction is for the command phase, the SYM53C1010-33 receives the first command byte and decodes its SCSI Group Code.

- If the SCSI Group Code is either Group 0, Group 1, Group 2, or Group 5, then the SYM53C1010-33 overwrites the DMA Byte Counter (DBC) register with the length of the Command Descriptor Block: 6, 10, or 12 bytes.
- If the Vendor Unique Enhancement 0 (VUE0) bit (SCSI Control Two (SCNTL2), bit 1) is cleared and the SCSI group code is a vendor unique code, the SYM53C1010-33 overwrites the DMA Byte Counter (DBC) register with the length of the Command Descriptor Block: 6, 10, or 12 bytes. If the VUE0 bit is set, the SYM53C1010-33 receives the number of bytes in the byte count regardless of the group code.
- If any other Group Code is received, the DMA Byte Counter (DBC) register is not modified and the SYM53C1010-33 requests the number of bytes specified in the DMA Byte Counter (DBC) register. If the DBC register contains 0x000000, an illegal instruction interrupt is generated.

The SYM53C1010-33 transfers the number of bytes specified in the DMA Byte Counter (DBC) register starting at the address specified in the DMA Next Address (DNAD) register. If the Opcode bit is set and a data transfer ends on an odd byte boundary, the SYM53C1010-33 stores the last byte in the SCSI Wide Residue (SWIDE) register during a receive operation. This byte is combined with the first byte from the subsequent transfer so that a wide transfer can complete.

If the SATN/ signal is asserted by the initiator or a parity error occurred during the transfer, it is possible to halt the transfer and generate an interrupt. The Disable Halt on Parity Error or ATN bit in the SCSI Control One (SCNTL1) register controls whether the SYM53C1010-33 halts on

these conditions immediately, or waits until completion of the current Move.

**Initiator Mode**

| OPC | Instruction Defined |
|---|---|
| 0 | CHMOV/CHMOV64 |
| 1 | MOVE/MOVE64 |

The SYM53C1010-33 verifies that it is connected to the SCSI bus as an initiator before executing this instruction.

The SYM53C1010-33 waits for an unserviced phase to occur. An unserviced phase is defined as any phase (with SREQ/ asserted) for which the SYM53C1010-33 has not yet transferred data by responding with a SACK/.

The SYM53C1010-33 compares the SCSI phase bits in the DMA Command (DCMD) register with the latched SCSI phase lines stored in the SCSI Status One (SSTAT1) register. These phase lines are latched when SREQ/ is asserted.

If the SCSI phase bits match the value stored in the SCSI Status One (SSTAT1) register, the SYM53C1010-33 transfers the number of bytes specified in the DMA Byte Counter (DBC) register starting at the address pointed to by the DMA Next Address (DNAD) register. If the opcode bit is cleared and a data transfer ends on an odd byte boundary, the SYM53C1010-33 stores the last byte in the SCSI Wide Residue (SWIDE) register during a receive operation, or in the SCSI Output Data Latch (SODL) register during a send operation. This byte is combined with the first byte from the subsequent transfer so that a wide transfer can complete.

If the SCSI phase bits do not match the value stored in the SCSI Status One (SSTAT1) register, the SYM53C1010-33 generates a phase mismatch interrupt and the instruction is not executed.

During a Message-Out phase, after the SYM53C1010-33 has performed a select with Attention (or SATN/ is manually asserted with a Set ATN instruction), the SYM53C1010-33 deasserts SATN/ during the final SREQ/SACK/ handshake.

When the SYM53C1010-33 is performing a block move for Message-In phase, it does not deassert the SACK/ signal for the last SREQ/SACK/ handshake. Clear the SACK/ signal using the Clear SACK I/O instruction.

**SCSIP[2:0]** **SCSI Phase** **[26:24]**

This field defines the desired SCSI information transfer phase. When the SYM53C1010-33 operates in the initiator mode, these bits are compared with the latched SCSI phase bits in the SCSI Status One (SSTAT1) register. When the SYM53C1010-33 operates in the target mode, it asserts the phase defined in this field. The following table describes the possible combinations and the corresponding SCSI phase.

| MSG | C_D | I_O | SCSI Phase |
|---|---|---|---|
| 0 | 0 | 0 | ST Data-Out |
| 0 | 0 | 1 | ST Data-In |
| 0 | 1 | 0 | Command |
| 0 | 1 | 1 | Status |
| 1 | 0 | 0 | DT Data-Out |
| 1 | 0 | 1 | DT Data-In |
| 1 | 1 | 0 | Message-Out |
| 1 | 1 | 1 | Message-In |

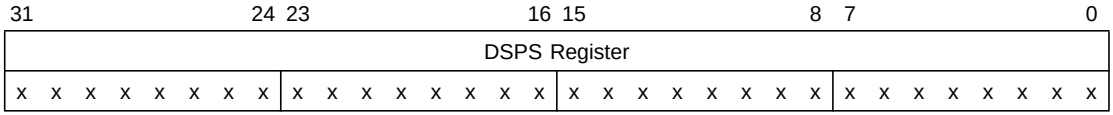
**TC[23:0]** **Transfer Counter** **[23:0]**

This 24-bit field specifies the number of data bytes to be moved between the SYM53C1010-33 and system memory. The field is stored in the DMA Byte Counter (DBC) register. When the SYM53C1010-33 transfers data to/from memory, the DBC register is decremented by the number of bytes transferred. In addition, the DMA Next Address (DNAD) register is incremented by the number of bytes transferred. This process is repeated until the DBC register is decremented to zero. At this time, the SYM53C1010-33 fetches the next instruction.

If bit 28 is set, indicating table indirect addressing, this field is not used. The byte count is instead fetched from a table pointed to by the Data Structure Address (DSA) register.

### 5.2.2 Second Dword

**Figure 5.3 Block Move Instruction - Second Dword**



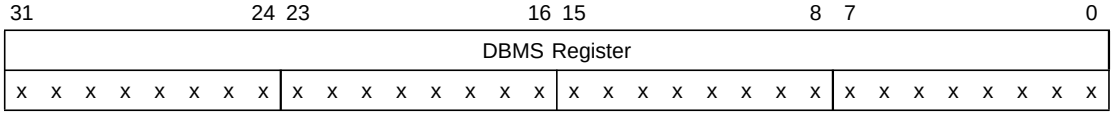

**Start Address** **[31:0]**

This 32-bit field specifies the starting address of the data to move to/from memory. This field is copied to the DMA Next Address (DNAD) register. When the SYM53C1010-33 transfers data to or from memory, the DNAD register is incremented by the number of bytes transferred.

When bit 29 is set, indicating indirect addressing, this address is a pointer to an address in memory that points to the data location. When bit 28 is set, indicating table indirect addressing, the value in this field is an offset into a table pointed to by the Data Structure Address (DSA). The table entry contains byte count and address information.

### 5.2.3 Third Dword

**Figure 5.4 Block Move Instruction - Third Dword**




**Start Address** **[63:32]**

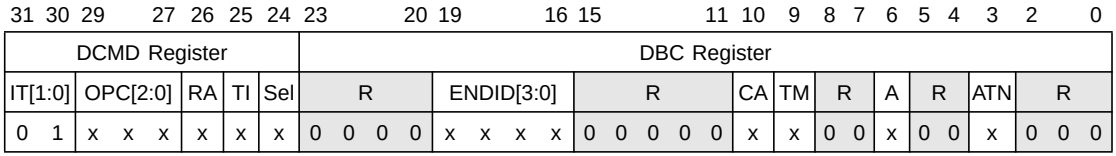
This 32-bit field specifies the upper Dword of a 64-bit starting address of data to move to/from memory. This field is copied to the Dynamic Block Move Selector (DBMS) register. The EN64DBMV bit in the Chip Control One (CCNTL1) register must be set for this Dword to be fetched.

## 5.3 I/O Instructions

This section contains information about the I/O Instruction Register. It is divided into First Dword and Second Dword.

### 5.3.1 First Dword

**Figure 5.5 First 32-bit Word of the I/O Instruction**




**IT[1:0]** **Instruction Type - I/O Instruction** **[31:30]**

**OPC[2:0]** **Opcode** **[29:27]**

The following Opcode bits have different meanings, depending on whether the SYM53C1010-33 is operating in the initiator or target mode. Opcode selections 0b101–0b111 are considered Read/Write instructions, and are described in Section 5.4, “Read/Write Instructions”.

**Target Mode**

| OPC2 | OPC1 | OPC0 | Instruction Defined |
|---|---|---|---|
| 0 | 0 | 0 | Reselect |
| 0 | 0 | 1 | Disconnect |
| 0 | 1 | 0 | Wait Select |
| 0 | 1 | 1 | Set |
| 1 | 0 | 0 | Clear |

**Reselect Instruction**

The SYM53C1010-33 arbitrates for the SCSI bus by asserting the SCSI ID stored in the SCSI Chip ID (SCID) register. If it loses arbitration, it tries again during the next available arbitration cycle without reporting any lost arbitration status.

If the SYM53C1010-33 wins arbitration, it attempts to reselect the SCSI device whose ID is defined in the

destination ID field of the instruction. Once the SYM53C1010-33 wins arbitration, it fetches the next instruction from the address pointed to by the DMA SCRIPTS Pointer (DSP) register. This way the SCRIPTS can move on to the next instruction before the reselection completes. It continues executing SCRIPTS until a SCRIPT that requires a response from the initiator is encountered.

If the SYM53C1010-33 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 DMA Next Address (DNAD) register. Manually set the SYM53C1010-33 to the initiator mode if it is reselected, or to the target mode if it is selected.

**Disconnect Instruction**

The SYM53C1010-33 disconnects from the SCSI bus by deasserting all SCSI signal outputs.

**Wait Select Instruction**

If the SYM53C1010-33 is selected, it fetches the next instruction from the address pointed to by the DMA SCRIPTS Pointer (DSP) register.

If reselected, the SYM53C1010-33 fetches the next instruction from the address pointed to by the 32-bit jump address field stored in the DMA Next Address (DNAD) register. Manually set the SYM53C1010-33 to the initiator mode when it is reselected.

If the CPU sets the SIGP bit in the Interrupt Status Zero (ISTAT0) register, the SYM53C1010-33 aborts the Wait Select instruction and fetches the next instruction from the address pointed to by the 32-bit jump address field stored in the DMA Next Address (DNAD) register.

**Set Instruction**

When the SACK/ or SATN/ bits are set, the corresponding bits in the SCSI Output Control Latch (SOCL) register are set. Do not set SACK/ or SATN/ except for testing purposes. When the target bit is set, the corresponding bit in the SCSI Control Zero (SCNTL0) register is also set. When the carry bit is set, the corresponding bit in the Arithmetic Logic Unit (ALU) is set.

Note: None of the signals are set on the SCSI bus in target mode.

**Clear Instruction**

When the SACK/ or SATN/ bits are cleared, the corresponding bits are cleared in the SCSI Output Control Latch (SOCL) register. Do not set SACK/ or SATN/ except for testing purposes. When the target bit is cleared, the corresponding bit in the SCSI Control Zero (SCNTL0) register is cleared. When the carry bit is cleared, the corresponding bit in the ALU is cleared.

Note: None of the signals are cleared on the SCSI bus in the target mode.

**Initiator Mode**

| OPC2 | OPC1 | OPC0 | Instruction Defined |
|---|---|---|---|
| 0 | 0 | 0 | Select |
| 0 | 0 | 1 | Wait Disconnect |
| 0 | 1 | 0 | Wait Reselect |
| 0 | 1 | 1 | Set |
| 1 | 0 | 0 | Clear |

**Select Instruction**

The SYM53C1010-33 arbitrates for the SCSI bus by asserting the SCSI ID stored in the SCSI Chip ID (SCID) register. If it loses arbitration, it tries again during the next available arbitration cycle without reporting any lost arbitration status.

If the SYM53C1010-33 wins arbitration, it attempts to select the SCSI device whose ID is defined in the destination ID field of the instruction. Once the SYM53C1010-33 wins arbitration, it fetches the next instruction from the address pointed to by the DMA SCRIPTS Pointer (DSP) register. This way the SCRIPTS can move to the next instruction before the selection completes. It continues executing SCRIPTS until a SCRIPT that requires a response from the target is encountered.

If the SYM53C1010-33 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 DMA Next Address (DNAD) register. Manually set the SYM53C1010-33 to the initiator mode if it is reselected, or to the target mode if it is selected.

If the Select with SATN/ field is set, the SATN/ signal is asserted during the selection phase.

**Wait Disconnect Instruction**

The SYM53C1010-33 waits for the target to perform a "legal" disconnect from the SCSI bus. A "legal" disconnect occurs when SBSY/ and SSEL/ are inactive for a minimum of one Bus Free delay (400 ns), after the SYM53C1010-33 receives a Disconnect Message or a Command Complete Message.

**Wait Reselect Instruction**

If the SYM53C1010-33 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 DMA Next Address (DNAD) register. Manually set the SYM53C1010-33 to the target mode when it is selected.

If the SYM53C1010-33 is reselected, it fetches the next instruction from the address pointed to by the DMA SCRIPTS Pointer (DSP) register.

If the CPU sets the SIGP bit in the Interrupt Status Zero (ISTAT0) register, the SYM53C1010-33 aborts the Wait Reselect instruction and fetches the next instruction from the address pointed to by the 32-bit jump address field stored in the DMA Next Address (DNAD) register.

**Set Instruction**

When the SACK/ or SATN/ bits are set, the corresponding bits in the SCSI Output Control Latch (SOCL) register are set. When the target bit is set, the corresponding bit in the SCSI Control Zero (SCNTL0) register is also set. When the carry bit is set, the corresponding bit in the ALU is set.

**Clear Instruction**

When the SACK/ or SATN/ bits are cleared, the corresponding bits are cleared in the SCSI Output Control Latch (SOCL) register. When the target bit is cleared, the corresponding bit in the SCSI Control Zero

(SCNTL0) register is cleared. When the carry bit is cleared, the corresponding bit in the ALU is cleared.

**RA** **Relative Addressing Mode** **26**

When this bit is set, the 24-bit signed value in the DMA Next Address (DNAD) register is used as a relative displacement from the current DMA SCRIPTS Pointer (DSP) address. Use this bit only in conjunction with the Select, Reselect, Wait Select, and Wait Reselect instructions. The Select and Reselect instructions can contain an absolute alternate jump address or a relative transfer address.

**TI** **Table Indirect Mode** **25**

When this bit is set, the 24-bit signed value in the DMA Byte Counter (DBC) register is added to the value in the Data Structure Address (DSA) register, and used as an offset relative to the value in the Data Structure Address (DSA) register. The SCSI Control Three (SCNTL3) value, SCSI ID, synchronous offset and synchronous period are loaded from this address. Prior to the start of an I/O, load the Data Structure Address (DSA) with the base address of the I/O data structure. Any address on a Dword boundary is allowed. After a Table Indirect opcode is fetched, the Data Structure Address (DSA) is added to the 24-bit signed offset value from the opcode to generate the address of the required data. Both positive and negative offsets are allowed. A subsequent fetch from that address brings the data values into the chip.

SCRIPTS can directly execute operating system I/O data structures, saving time at the beginning of an I/O operation. The I/O data structure can begin on any Dword boundary and may cross system segment boundaries. There are two restrictions on the placement of data in system memory:

- The I/O data structure must lie within the 8 Mbytes above or below the base address.
- An I/O command structure must have all four bytes contiguous in system memory, as shown below. The offset/period bits are ordered as in the SCSI Transfer (SXFER) register. The configuration bits are ordered as in the SCSI Control Three (SCNTL3) register.

| Config | ID | Offset/period | 00 |
|---|---|---|---|

Use this bit only in conjunction with the Select, Reselect, Wait Select, and Wait Reselect instructions. It is allowable to set bits 25 and 26 individually or in combination:

| | Bit 25 | Bit 26 |
|---|---|---|
| Direct | 0 | 0 |
| Table Indirect | 0 | 1 |
| Relative | 1 | 0 |
| Table Relative | 1 | 1 |

### Direct

Uses the device ID and physical address in the instruction.

| Command | ID | Not Used | Not Used |
|---|---|---|---|
| Absolute Alternate Address | | | |

### Table Indirect

Uses the physical jump address, but fetches data using the table indirect method.

| Command | Table Offset |
|---|---|
| Absolute Alternate Address | |

### Relative

Uses the device ID in the instruction, but treats the alternate address as a relative jump.

| Command | ID | Not Used | Not Used |
|---|---|---|---|
| Absolute Jump Offset | | | |

### Table Relative

Treats the alternate jump address as a relative jump and fetches the device ID, synchronous offset, and synchronous period indirectly. The value in bits [23:0] of the first four bytes of the SCRIPTS instruction is added

to the data structure base address to form the fetch address.

| Command | Table Offset |
|---|---|
| Alternate Jump Offset | |

**Sel** **Select with ATN/** **24**
This bit specifies whether SATN/ is asserted during the selection phase when the SYM53C1010-33 is executing a Select instruction. When operating in the initiator mode, set this bit for the Select instruction. If this bit is set on any other I/O instruction, an illegal instruction interrupt is generated.

**R** **Reserved** **[23:20]**

**ENDID[3:0]** **Encoded SCSI Destination ID** **[19:16]**
This 4-bit field specifies the destination SCSI ID for an I/O instruction.

**R** **Reserved** **[15:11]**

**CA** **Set/Clear Carry** **10**
This bit is used in conjunction with a Set or Clear instruction to set or clear the Carry bit. Setting this bit with a Set instruction asserts the Carry bit in the ALU. Clearing this bit with a Clear instruction deasserts the Carry bit in the ALU.

**TM** **Set/Clear Target Mode** **9**
This bit is used in conjunction with a Set or Clear instruction to set or clear the target mode. Setting this bit with a Set instruction configures the SYM53C1010-33 as a target device (this sets bit 0 of the SCSI Control Zero (SCNTL0) register). Clearing this bit with a Clear instruction configures the SYM53C1010-33 as an initiator device (this clears bit 0 of the SCNTL0 register).

| | | |
|---|---|---|
| **R** | **Reserved** | **[8:7]** |
| **A** | **Set/Clear SACK/** | **6** |
| **R** | **Reserved** | **[5:4]** |
| **ATN** | **Set/Clear SATN/** | **3** |

These two bits are used in conjunction with a Set or Clear instruction to assert or deassert the corresponding SCSI control signal. Bit 6 controls the SCSI SACK/ signal. Bit 3 controls the SCSI SATN/ signal.

The Set instruction is used to assert SACK/ and/or SATN/ on the SCSI bus. The Clear instruction is used to deassert SACK/ and/or SATN/ on the SCSI bus. The corresponding bit in the SCSI Output Control Latch (SOCL) register will be set or cleared depending on the instruction used.

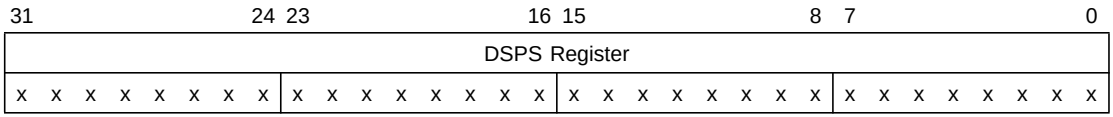
Since SACK/ and SATN/ are initiator signals, they are not asserted on the SCSI bus.

The Set/Clear SCSI ACK/, ATN/ instruction is used after message phase Block Move operations to give the initiator the opportunity to assert attention before acknowledging the last message byte. For example, if the initiator wishes to reject a message, it issues an Assert SCSI ATN instruction before a Clear SCSI ACK instruction.

| | | |
|---|---|---|
| **R** | **Reserved** | **[2:0]** |

## 5.3.2 Second Dword

**Figure 5.6 Second 32-bit Word of the I/O Instruction**




| | | |
|---|---|---|
| **SA** | **Start Address** | **[31:0]** |

This 32-bit field contains the memory address to fetch the next instruction if the selection or reselection fails.

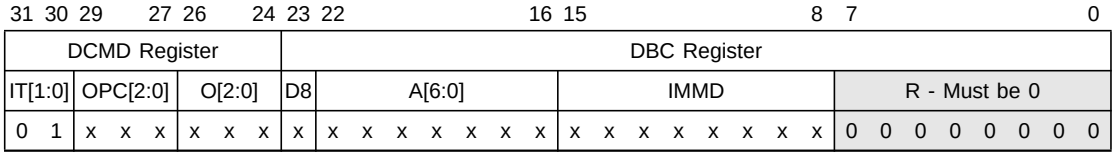
If relative or table relative addressing is used, this value is a 24-bit signed offset relative to the current DMA SCRIPTS Pointer (DSP) register value.

## 5.4 Read/Write Instructions

The Read/Write instruction supports addition, subtraction, and comparison of two separate values within the chip. It performs the desired operation on the specified register and the SCSI First Byte Received (SFBR) register and stores the result back to the specified register or to the SFBR. If the COM bit (DMA Control (DCNTL), bit 0) is cleared, Read/Write instructions cannot be used.

### 5.4.1 First Dword

**Figure 5.7 Read/Write Instruction - First Dword**




**IT[1:0]** **Instruction Type - Read/Write Instruction** **[31:30]**
The Read/Write instruction uses operator bits [26:24] in conjunction with the opcode bits to determine which instruction is currently selected.

**OPC[2:0]** **Opcode** **[29:27]**
These bits determine if the instruction is a Read/Write or an I/O instruction. Opcodes 0b000 through 0b100 are considered I/O instructions.

**O[2:0]** **Operator** **[26:24]**
The Operator bits are used in conjunction with the opcode bits to determine which instruction is currently selected. Refer to Table 5.1 for field definitions.

**D8** **Use data8/SFBR** **23**
When this bit is set, SCSI First Byte Received (SFBR) is used, instead of the data8 value, during a Read-Modify-Write instruction (see Table 5.1). This allows the user to add two register values.

**A[6:0]** **Register Address - A[6:0]** **[22:16]**
It is possible to change register values from SCRIPTS in read-modify-write cycles or move to/from SCSI First Byte

Received (SFBR) cycles. A[6:0] selects an 8-bit source/destination register within the SYM53C1010-33.

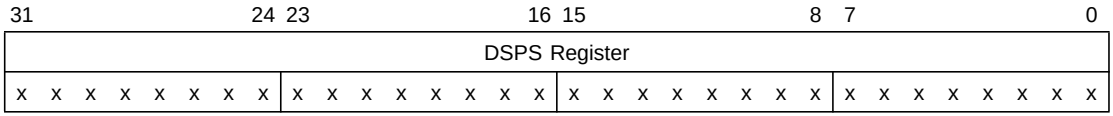
**IMMD** **Immediate Data** **[15:8]**
This 8-bit value is used as a second operand in logical and arithmetic functions.

**A7** **Upper Register Address Line (A7)** **7**
This bit is used to access registers 0x80–0xFF.

**R** **Reserved** **[6:0]**

## 5.4.2 Second Dword

**Figure 5.8 Read/Write Instruction - Second Dword**




**DA** **Destination Address** **[31:0]**
This field contains the 32-bit destination address where the data is to move.

## 5.4.3 Read-Modify-Write Cycles

During these cycles the register is read, the selected operation is performed, and the result is written back to the source register.

The Add operation is used to increment or decrement register values (or memory values if used in conjunction with a Memory-to-Register Move operation) for use as loop counters.

Subtraction is not available when SCSI First Byte Received (SFBR) is used instead of data8 in the instruction syntax. To subtract one value from another when using SFBR, first XOR the value to subtract (subtrahend) with 0xFF, and add 1 to the resulting value. This creates the 2's complement of the subtrahend. The two values are then added to obtain the difference.

### 5.4.4 Move To/From SFBR Cycles

All operations are read-modify-writes. However, two registers are involved, one of which is always the SCSI First Byte Received (SFBR). The possible functions of this instruction are:

- Write one byte (value contained within the SCRIPTS instruction) into any chip register.
- Move to/from the SCSI First Byte Received (SFBR) from/to any other register.
- Alter the value of a register with AND, OR, ADD, XOR, SHIFT LEFT, or SHIFT RIGHT operators.
- After moving values to the SCSI First Byte Received (SFBR), the compare and jump, call, or similar instructions are used to check the value.
- A Move-to-SFBR followed by a Move-from-SFBR is used to perform a register to register move.

**Table 5.1 Read/Write Instructions[1]**

| Operator | Opcode 111<br>Read Modify Write | Opcode 110<br>Move to SFBR | Opcode 101<br>Move from SFBR |
|---|---|---|---|
| 000 | Move data into register. Syntax: “Move data8 to RegA” | Move data into SCSI First Byte Received (SFBR) register. Syntax: “Move data8 to SFBR” | Move data into register. Syntax: “Move data8 to RegA” |
| $001^2$ | Shift register one bit to the left and place the result in the same register. Syntax: “Move RegA SHL RegA” | Shift register one bit to the left and place the result in the SCSI First Byte Received (SFBR) register. Syntax: “Move RegA SHL SFBR” | Shift the SFBR register one bit to the left and place the result in the register. Syntax: “Move SFBR SHL RegA” |
| 010 | OR data with register and place the result in the same register. Syntax: “Move RegA \| data8 to RegA” | OR data with register and place the result in the SCSI First Byte Received (SFBR) register. Syntax: “Move RegA \| data8 to SFBR” | OR data with SFBR and place the result in the register. Syntax: “Move SFBR \| data8 to RegA” |

**Table 5.1 Read/Write Instructions[1] (Cont.)**

| Operator | Opcode 111<br>Read Modify Write | Opcode 110<br>Move to SFBR | Opcode 101<br>Move from SFBR |
|---|---|---|---|
| 011 | XOR data with register and place the result in the same register. Syntax: “Move RegA XOR data8 to RegA” | XOR data with register and place the result in the SCSI First Byte Received (SFBR) register. Syntax: “Move RegA XOR data8 to SFBR” | XOR data with SFBR and place the result in the register. Syntax: “Move SFBR XOR data8 to RegA” |
| 100 | AND data with register and place the result in the same register. Syntax: “Move RegA & data8 to RegA” | AND data with register and place the result in the SCSI First Byte Received (SFBR) register. Syntax: “Move RegA & data8 to SFBR” | AND data with SFBR and place the result in the register. Syntax: “Move SFBR & data8 to RegA” |
| 101[2] | Shift register one bit to the right and place the result in the same register. Syntax: “Move RegA SHR RegA” | Shift register one bit to the right and place the result in the SCSI First Byte Received (SFBR) register. Syntax: “Move RegA SHR SFBR” | Shift the SFBR register one bit to the right and place the result in the register. Syntax: “Move SFBR SHR RegA” |
| 110 | Add data to register without carry and place the result in the same register. Syntax: “Move RegA + data8 to RegA” | Add data to register without carry and place the result in the SCSI First Byte Received (SFBR) register. Syntax: “Move RegA + data8 to SFBR” | Add data to SFBR without carry and place the result in the register. Syntax: “Move SFBR + data8 to RegA” |
| 111 | Add data to register with carry and place the result in the same register. Syntax: “Move RegA + data8 to RegA with carry” | Add data to register with carry and place the result in the SCSI First Byte Received (SFBR) register. Syntax: “Move RegA + data8 to SFBR with carry” | Add data to SFBR with carry and place the result in the register. Syntax: “Move SFBR + data8 to RegA with carry” |

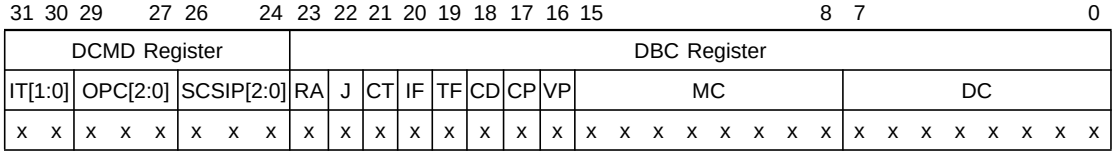
1. Substitute the desired register name or address for “RegA” in the syntax examples. data8 indicates eight bits of data.
   Use SFBR instead of data8 to add two register values.
2. Data is shifted through the Carry bit and the Carry bit is shifted into the data byte.

## 5.5 Transfer Control Instructions

This section describes transfer control instructions for the first Dword, second Dword, and third Dword.

### 5.5.1 First Dword

**Figure 5.9 Transfer Control Instructions - First Dword**




**IT[1:0]** **Instruction Type - Transfer Control Instruction** **[31:30]**

**OPC[2:0]** **Opcode** **[29:27]**

This 3-bit field specifies the type of transfer control instruction to execute. All transfer control instructions can be conditional. They can be dependent on a true/false comparison of the ALU Carry bit or 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 the initiator or target mode.

| OPC2 | OPC1 | OPC0 | Instruction Defined |
|---|---|---|---|
| 0 | 0 | 0 | Jump |
| 0 | 0 | 1 | Call |
| 0 | 1 | 0 | Return |
| 0 | 1 | 1 | Interrupt |
| 1 | x | x | Reserved |

**Jump Instruction**

The SYM53C1010-33 can do a true/false comparison of the ALU carry bit, or compare the phase and/or data as defined by the Phase Compare, Data Compare and True/False bit fields. If the comparisons are true, it loads the DMA SCRIPTS Pointer (DSP) register with the

contents of the DMA SCRIPTS Pointer Save (DSPS) register. The DSP register now contains the address of the next instruction.

If the comparisons are false, the SYM53C1010-33 fetches the next instruction from the address pointed to by the DMA SCRIPTS Pointer (DSP) register, leaving the instruction pointer unchanged.

When the JUMP64 instruction is used, a third Dword is fetched and loaded into the SCRIPT Fetch Selector (SFS) register. Bit 22 indicates whether the jump is to a 32-bit address (0) or a 64-bit address (1). All combinations of jumps are still valid for JUMP64.

**Call Instruction**

The SYM53C1010-33 can do a true/false comparison of the ALU carry bit, or compare the phase and/or data as defined by the Phase Compare, Data Compare, and True/False bit fields. If the comparisons are true, it loads the DMA SCRIPTS Pointer (DSP) register with the contents of the DMA SCRIPTS Pointer Save (DSPS) register and that address value becomes the address of the next instruction.

When the SYM53C1010-33 executes a Call instruction, the instruction pointer contained in the DMA SCRIPTS Pointer (DSP) register is stored in the Temporary (TEMP) register. Since the TEMP register is not a stack and can only hold one Dword, nested call instructions are not allowed.

If the comparisons are false, the SYM53C1010-33 fetches the next instruction from the address pointed to by the DMA SCRIPTS Pointer (DSP) register and the instruction pointer is not modified.

**Return Instruction**

The SYM53C1010-33 can do a true/false comparison of the ALU carry bit, or compare the phase and/or data as defined by the Phase Compare, Data Compare, and True/False bit fields. If the comparisons are true, it loads the DMA SCRIPTS Pointer (DSP) register with the contents of the DMA SCRIPTS Pointer Save (DSPS) register. That address value becomes the address of the next instruction.

When a Return instruction is executed, the value stored in the Temporary (TEMP) register is returned to the DMA SCRIPTS Pointer (DSP) register. The SYM53C1010-33 does not check to see whether the Call instruction has already been executed. It does not generate an interrupt if a Return instruction is executed without previously executing a Call instruction.

If the comparisons are false, the SYM53C1010-33 fetches the next instruction from the address pointed to by the DMA SCRIPTS Pointer (DSP) register and the instruction pointer is not modified.

**Interrupt Instruction**

The SYM53C1010-33 can do a true/false comparison of the ALU carry bit, or 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 SYM53C1010-33 generates an interrupt by asserting the IRQ/ signal.

The 32-bit address field stored in the DMA SCRIPTS Pointer Save (DSPS) register can contain a unique interrupt service vector. When servicing the interrupt, this unique status code allows the ISR to quickly identify the point at which the interrupt occurred.

The SYM53C1010-33 halts and the DMA SCRIPTS Pointer (DSP) register must be written to before starting any further operation.

**Interrupt-on-the-Fly Instruction**

The SYM53C1010-33 can do a true/false comparison of the ALU carry bit or compare the phase and/or data as defined by the Phase Compare, Data Compare, and True/False bit fields. If the comparisons are true, and the Interrupt-on-the-Fly bit Interrupt Status Zero (ISTAT0), bit 2) is set, the SYM53C1010-33 asserts the Interrupt-on-the-Fly bit.

**SCSIP[2:0]** **SCSI Phase** **[26:24]**

This 3-bit field corresponds to the three SCSI bus phase signals that are compared with the phase lines latched when SREQ/ is asserted. Comparisons can be performed to determine the SCSI phase actually being driven on the SCSI bus. The following table describes the possible

combinations and their corresponding SCSI phase. These bits are only valid when the SYM53C1010-33 is operating in the initiator mode. Clear these bits when the SYM53C1010-33 is operating in the target mode.

| MSG | C/D | I/O | SCSI Phase |
|---|---|---|---|
| 0 | 0 | 0 | ST Data-Out |
| 0 | 0 | 1 | ST Data-In |
| 0 | 1 | 0 | Command |
| 0 | 1 | 1 | Status |
| 1 | 0 | 0 | DT Data-Out |
| 1 | 0 | 1 | DT Data-In |
| 1 | 1 | 0 | Message-Out |
| 1 | 1 | 1 | Message-In |

**RA** **Relative Addressing Mode** **23**

When this bit is set, the 24-bit signed value in the DMA SCRIPTS Pointer Save (DSPS) register is used as a relative offset from the current DMA SCRIPTS Pointer (DSP) address, which is pointing to the next instruction and not the one currently executing. The relative mode does not apply to Return and Interrupt SCRIPTS.

**Jump/Call an Absolute Address**

Start execution at the new absolute address.

| Command | Condition Codes |
|---|---|
| Absolute Alternate Address | |

**Jump/Call a Relative Address**

Start execution at the current address plus (or minus) the relative offset.

| Command | Condition Codes |
|---|---|
| Don't Care | Alternate Jump Offset |

The SCRIPTS program counter is a 32-bit value pointing to the SCRIPTS currently under execution by the SYM53C1010-33. The next address is formed by adding the 32-bit program counter to the 24-bit signed value of the last 24 bits of the Jump or Call instruction. Because

it is signed (2's complement), the jump can be forward or backward.

A relative transfer can be to any address within a 16 Mbyte segment. The program counter is combined with the 24-bit signed offset (using addition or subtraction) to form the new execution address.

SCRIPTS programs may contain a mixture of direct jumps and relative jumps to provide maximum versatility when writing SCRIPTS. For example, major sections of code can be accessed with far calls using the 32-bit physical address, then local labels can be called using relative transfers. If a SCRIPT is written using only relative transfers it does not require any run time alteration of physical addresses, and can be stored in and executed from a PROM.

**J** **32/64 Bit Jump** **22**
When this bit is cleared, the jump address is 32 bits wide. When this bit is set, the jump address is 64 bits wide.

**CT** **Carry Test** **21**
When this bit is set, decisions based on the ALU carry bit can be made. True/False comparisons are legal, but Data Compare and Phase Compare are illegal.

**IF** **Interrupt-on-the-Fly** **20**
When this bit is set, the interrupt instruction does not halt the SCRIPTS processor. Once the interrupt occurs, the Interrupt-on-the-Fly bit (Interrupt Status Zero (ISTAT0), bit 2) is asserted.

**TF** **Jump If True/False** **19**
This bit determines whether the SYM53C1010-33 branches when a comparison is true or when a comparison is false. This bit applies to phase compares, data compares, and carry tests. If both the Phase Compare and Data Compare bits are set, then both compares must be true to branch on a true condition. Both compares must be false to branch on a false condition.

| Bit 19 | Result of Compare | Action |
|---|---|---|
| 0 | False | Jump Taken |
| 0 | True | No Jump |
| 1 | False | No Jump |
| 1 | True | Jump Taken |

**CD** **Compare Data** **18**

When this bit is set, the first byte received from the SCSI data bus (contained in the SCSI First Byte Received (SFBR) register) is compared with the Data to be Compared Field in the Transfer Control instruction. The Wait for Valid Phase bit controls when this compare occurs. The Jump if True/False bit determines the condition (true or false) to branch on.

**CP** **Compare Phase** **17**

When the SYM53C1010-33 is in the initiator mode, this bit controls phase compare operations. When this bit is set, the SCSI phase signals (latched by SREQ/) are compared to the Phase Field in the Transfer Control instruction. If they match, the comparison is true. The Wait for Valid Phase bit controls when the compare occurs. When the SYM53C1010-33 is operating in the target mode and this bit is set it tests for an active SCSI SATN/ signal.

**VP** **Wait For Valid Phase** **16**

If the Wait for Valid Phase bit is set, the SYM53C1010-33 waits for a previously unserviced phase before comparing the SCSI phase and data.

If the Wait for Valid Phase bit is cleared, the SYM53C1010-33 compares the SCSI phase and data immediately.

**MC** **Data Compare Mask** **[15:8]**

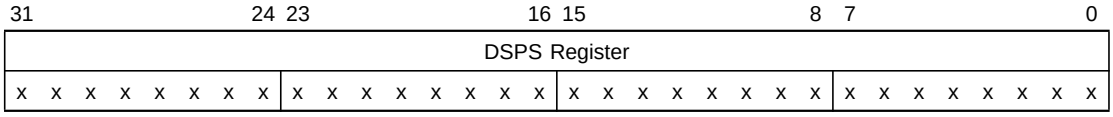
The Data Compare Mask allows a SCRIPT to test certain bits within a data byte. During the data compare, if any mask bits are set, the corresponding bit in the SCSI First Byte Received (SFBR) data byte is ignored. For instance, a mask of 0b01111111 and data compare value of 0b1XXXXXXX allows the SCRIPTS processor to

determine whether or not the high-order bit is set while ignoring the remaining bits.

**DC** **Data Compare Value** **[7:0]**
This 8-bit field is the data compared against the SCSI First Byte Received (SFBR) register. These bits are used in conjunction with the Data Compare Mask Field to test for a particular data value. If the COM bit (DMA Control (DCNTL), bit 0) is cleared, the value in the SFBR register may not be stable. In this case, do not use instructions using this data compare value.

## 5.5.2 Second Dword

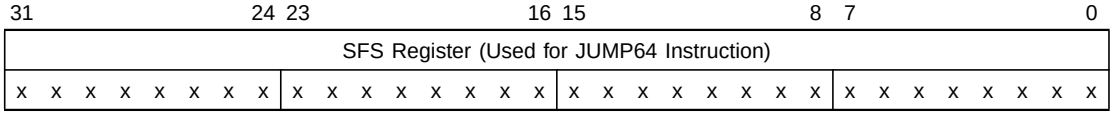
**Figure 5.10 Transfer Control Instructions - Second Dword**




**Jump Address** **[31:0]**
This 32-bit field contains the address of the next instruction to fetch when a jump is taken. Once the SYM53C1010-33 fetches the instruction from the address pointed to by these 32 bits, this address is incremented by 4, loaded into the DMA SCRIPTS Pointer (DSP) register and becomes the current instruction pointer.

## 5.5.3 Third Dword

**Figure 5.11 Transfer Control Instructions - Third Dword**




**JUMP64 Address** **[31:0]**
This 32-bit field contains the upper Dword of a 64-bit address of the next instruction to fetch when a JUMP64 is taken.

## 5.6 Memory Move Instructions

For Memory Move instructions, bits 5 and 4 (SIOM and DIOM) in the DMA Mode (DMODE) register determine whether the source or destination addresses reside in memory or I/O space. By setting these bits appropriately, data may be moved within memory space, within I/O space, or between the two address spaces.

The Memory Move instruction is used to copy the specified number of bytes from the source address to the destination address.

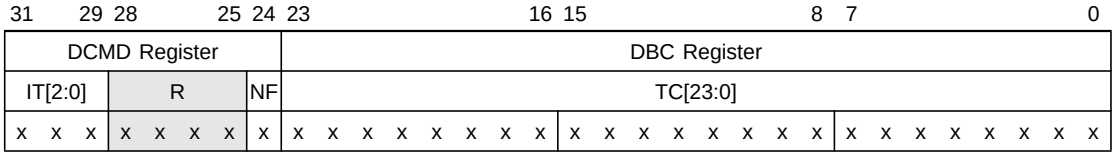
For memory moves where the data read is from the 64-bit address space, the upper Dword of the address resides in the Memory Move Read Selector (MMRS) register. For memory moves where the data is written to the 64-bit address space, the upper Dword of the address resides in the Memory Move Write Selector (MMWS) register.

Allowing the SYM53C1010-33 to perform memory moves frees the system processor for other tasks and moves data at higher speeds than available from current DMA controllers. Up to 16 Mbytes may be transferred with one instruction. There are two restrictions:

- Both the source and destination addresses must start with the same address alignment (A[1:0]) must be the same). If the source and destination are not aligned, then an illegal instruction interrupt occurs. For the PCI Cache Line Size (CLS) register setting to take effect, the source and destination must be the same distance from a cache line boundary.
- Indirect addresses are not allowed. A burst of data is fetched from the source address, put into the DMA FIFO and then written out to the destination address. The move continues until the byte count decrements to zero, then another SCRIPT is fetched from system memory.

The DMA SCRIPTS Pointer Save (DSPS) and Data Structure Address (DSA) registers are additional holding registers used during the Memory Move. However, the contents of the Data Structure Address (DSA) register are preserved.

**Figure 5.12 Memory Move Instructions - First Dword**




**IT[2:0]** **Instruction Type - Memory Move** **[31:29]**

**R** **Reserved** **[28:25]**
These bits are reserved and must be zero. If any of these bits are set, an illegal instruction interrupt occurs.

**NF** **No Flush** **24**
When this bit is set, the SYM53C1010-33 performs a Memory Move without flushing the prefetch unit. When this bit is cleared, the Memory Move instruction automatically flushes the prefetch unit. Use the No Flush option if the source and destination are not within four instructions of the current Memory Move instruction.

Note: This bit has no effect unless the Prefetch Enable bit in the DMA Control (DCNTL) register is set. For information on SCRIPTS instruction prefetching, see Chapter 2, "Functional Description".

**TC[23:0]** **Transfer Count** **[23:0]**
The number of bytes to transfer is stored in the lower 24 bits of the first instruction word.

## 5.6.1 Read/Write System Memory from a SCRIPT

By using the Memory Move instruction, single or multiple register values are transferred to or from system memory.

Because the SYM53C1010-33 responds to addresses as defined in the Base Address Register Zero (BAR0) (I/O) or Base Address Register One (BAR1) (MEMORY) registers, the device can be accessed during a Memory Move operation if the source or destination address decodes to within the chip's register space. If this occurs, the register indicated by the lower seven bits of the address is taken as the data source or destination. In this way, register values are saved to system memory and

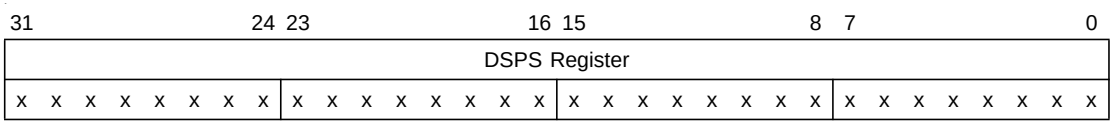
later restored, and SCRIPTS can make decisions based on data values in system memory.

The SCSI First Byte Received (SFBR) is not writable using the CPU, and therefore not by a Memory Move. However, it can be loaded using SCRIPTS Read/Write operations. To load the SFBR with a byte stored in system memory, first move the byte to an intermediate SYM53C1010-33 register (for example, a SCRATCH register), and then to the SCSI First Byte Received (SFBR).

The same address alignment restrictions apply to register access operations as to normal memory-to-memory transfers.

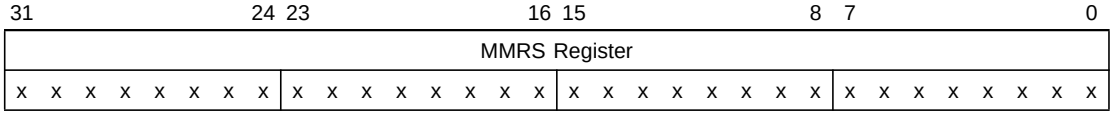
## 5.6.2 Second Dword

**Figure 5.13 Memory Move Instructions - Second Dword**



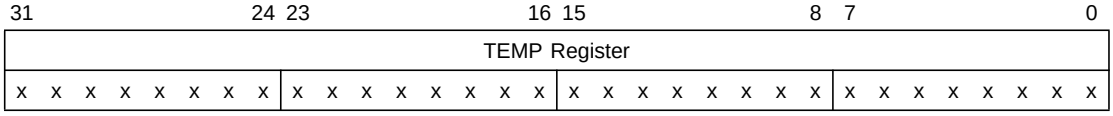

**DSPS Register** **[31:0]**
These bits contain the source address of the Memory Move.

If the source address is in the 64-bit address space, the bits will be contained in the Memory Move Read Selector (MMRS) register.



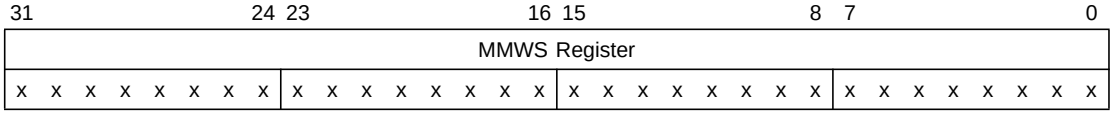

## 5.6.3 Third Dword

**Figure 5.14 Memory Move Instructions - Third Dword**



**TEMP Register** **[31:0]**
These bits contain the destination address for the Memory Move.

If the destination address is in the 64-bit address space, the bits will be contained in the Memory Move Write Selector (MMWS) register.




## 5.7 Load and Store Instructions

The Load and Store instructions provide a more efficient way to move data from/to memory to/from an internal register in the chip without using the normal memory move instruction.

The load and store instructions are represented by two-Dword opcodes. The first Dword contains the DMA Command (DCMD) and DMA Byte Counter (DBC) register values. The second Dword contains the DMA SCRIPTS Pointer Save (DSPS) value. This is either the actual memory location of where to load/store, or the offset from the Data Structure Address (DSA), depending on the value of bit 28 (DSA Relative).

For load operations where the data is read from the 64-bit address space, the upper Dword of address resides in the Memory Move Read Selector (MMRS) register. For store operations where the data is written to the 64-bit address space, the upper Dword of address resides in the Memory Move Write Selector (MMWS) register.

A maximum of 4 bytes may be moved with these instructions. The register address and memory address must have the same byte alignment, and the count set such that it does not cross Dword boundaries. The memory address may not map back to the chip, excluding RAM and ROM. If it does, a PCI read/write cycle occurs (the data does not actually transfer to/from the chip), and the chip issues an interrupt (Illegal Instruction Detected) immediately following.

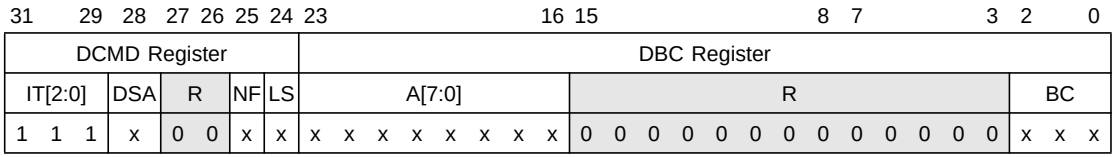
| Bits A1, A0 | Number of Bytes Allowed to Load/Store |
|---|---|
| 00 | One, two, three or four |
| 01 | One, two, or three |
| 10 | One or two |
| 11 | One |

The SIOM and DIOM bits in the DMA Mode (DMODE) register determine whether the destination or source address of the instruction is in Memory space or I/O space, as illustrated in the following table. The Load/Store utilizes the PCI commands for I/O read and I/O write to access the I/O space.

| Bit | Source | Destination |
|---|---|---|
| SIOM (Load) | Memory | Register |
| DIOM (Store) | Register | Memory |

## 5.7.1 First Dword

**Figure 5.15 Load and Store Instructions - First Dword**




**IT[2:0]** **Instruction Type** **[31:29]**
These bits should be 0b111, indicating the Load and Store instruction.

**DSA** **DSA Relative** **28**
When this bit is cleared, the value in the DMA SCRIPTS Pointer Save (DSPS) is the actual 32-bit memory address used to perform the Load/Store to/from. When this bit is set, the chip determines the memory address to perform the Load/Store to/from by adding the 24-bit signed offset value in the DMA SCRIPTS Pointer Save (DSPS) to the Data Structure Address (DSA).

**R** **Reserved** **[27:26]**

**NF** **No Flush (Store instruction only)** **25**
When this bit is set, the SYM53C1010-33 performs a Store without flushing the prefetch unit. When this bit is cleared, the Store instruction automatically flushes the prefetch unit. Use No Flush if the source and destination are not within four instructions of the current Store instruction. This bit has no effect on the Load instruction.

Note: This bit has no effect unless the Prefetch Enable bit in the DMA Control (DCNTL) register is set. For information on SCRIPTS instruction prefetching, see Chapter 2, "Functional Description".

**LS** **Load/Store** **24**
When this bit is set, the instruction is a Load. When cleared, it is a Store.

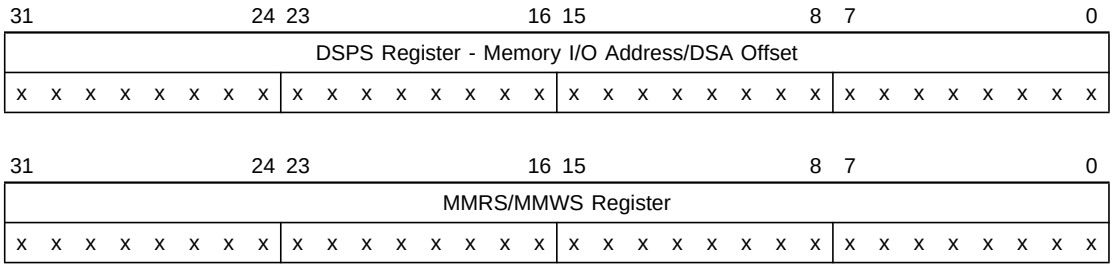
**A[7:0]** **Register Address** **[23:16]**
A[7:0] selects the register to load/store to/from within the SYM53C1010-33.

**R** **Reserved** **[15:3]**

**BC** **Byte Count** **[2:0]**
This value is the number of bytes to load/store.

## 5.7.2 Second Dword

**Figure 5.16 Load and Store Instructions - Second Dword**




**Memory I/O Address/DSA Offset** **[31:0]**
This is the actual memory location of where to load/store, or the offset from the Data Structure Address (DSA) register value.

# Chapter 6
# Specifications

This chapter specifies the SYM53C1010-33 electrical and mechanical characteristics. It is divided into the following sections:

- Section 6.1, “DC Characteristics”
- Section 6.2, “TolerANT Technology Electrical Characteristics”
- Section 6.3, “AC Characteristics”
- Section 6.4, “PCI and External Memory Interface Timing Diagrams”
- Section 6.5, “SCSI Timing Diagrams”

## 6.1 DC Characteristics

**Table 6.1 Absolute Maximum Stress Ratings**

| Symbol | Parameter | Min | Max[1] | Unit | Test Conditions |
|---|---|---|---|---|---|
| $T_{STG}$ | Storage temperature | –55 | 150 | °C | – |
| $V_{DD}$ | Supply voltage | –0.5 | 4.5 | V | – |
| $V_{IN}$ | Input voltage | –0.3 | 5.55 | V | – |
| $V_{IN-PCI}$ | Input voltage PCI pins | –.5 | 11.0 | V | – |
| $I_{LP}$[2] | Latch-up current | ±150 | – | mA | – |
| T | Lead temperature | – | 250 | °C | < 10 seconds |
| ESD[3] | Electrostatic discharge | – | 2 K | V | MIL-STD 883C, Method 3015.7 |

1. Stresses beyond those listed above may cause permanent damage to the device. These are stress ratings only; functional operation of the device at these or any other conditions beyond those indicated in the Operating Conditions section of the manual is not implied.
2. –2 V < VPIN < 8 V.
3. SCSI pins only.

### Table 6.2 Operating Conditions

| Symbol | Parameter | Min | Max[1] | Unit | Test Conditions |
|---|---|---|---|---|---|
| $V_{DD}$ | Supply voltage | 3.13 | 3.47 | V | – |
| $I_{DD}$ | Supply current (dynamic)[2] | – | 800 | mA | – |
| $I_{DD\text{-}SCSI}$ | LVD Mode Supply Current (dynamic) | – | 1 | A | – |
| $T_A$ | Operating free air | 0 | 70 | °C | – |
| $\theta_{JA}$ | Thermal resistance (junction to ambient air) | – | 20 | °C/W | – |

1. Conditions that exceed the operating limits may cause the device to function incorrectly.
2. Core and analog supply only.

### Table 6.3 LVD Driver SCSI Signals[1]—SD[15:0], SDP[1:0], SREQ/, SACK/, SMSG/, SIO/, SCD/, SATN/, SBSY/, SSEL/, SRST/

| Symbol | Parameter | Min | Max | Units | Test Conditions |
|---|---|---|---|---|---|
| $I_{O\text{-}ASSERT}+$ | Source (+) current | –9.6 | –14.4 | mA | Asserted state |
| $I_{O\text{-}ASSERT}-$ | Sink (–) current | 9.6 | 14.4 | mA | Asserted state |
| $I_{O\text{-}NEG}+$ | Source (+) current | –6.4 | –9.6 | mA | Negated state |
| $I_{O\text{-}NEG}-$ | Sink (–) current | 6.4 | 9.6 | mA | Negated state |
| $I_{OZ}$ | 3-state leakage | – | 20 | µA | – |

1. $V_{CM}$ = 0.7–1.8 V (Common Mode, nominal ~1.2 V), $R_L$ = 0–110 Ω, $R_{bias}$ = 10 kΩ.

**Figure 6.1 LVD Driver**

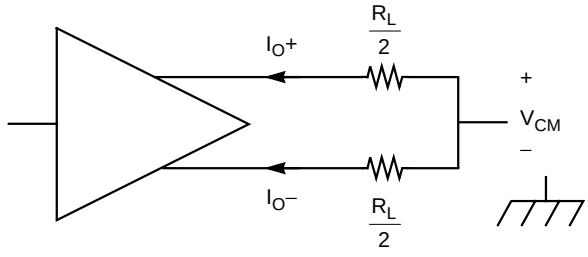


**Table 6.4 LVD Receiver SCSI Signals[1]—SD[15:0], SDP[1:0], SREQ/, SACK/, SMSG/, SIO/, SCD/, SATN/, SBSY/, SSEL/, SRST/**

| Symbol | Parameter | Min | Max | Units | Test Condition |
|---|---|---|---|---|---|
| $V_I$ | LVD receiver voltage asserting | 60 | – | mV | Differential voltage |
| $V_I$ | LVD receiver voltage negating | –60 | – | mV | Differential voltage |

1. $V_{CM}$ = 0.7–1.8 V (Common Mode Voltage, nominal ~1.2 V.)

**Figure 6.2 LVD Receiver**

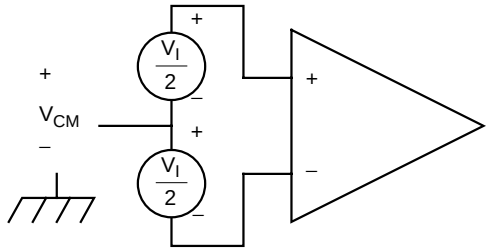



**Table 6.5 A and B DIFFSENS SCSI Signals**

| Symbol | Parameter | Min | Max | Unit | Test Conditions |
|---|---|---|---|---|---|
| $V_{IH}$ | HVD sense voltage | 2.4 | 5.05 | V | Note 1 |
| $V_S$ | LVD sense voltage | 0.7 | 1.9 | V | Note 1 |
| $V_{IL}$ | SE sense voltage | $V_{SS}$ –0.35 | 0.5 | V | Note 1 |
| $I_{OZ}$ | 3-state leakage | –10 | 10 | µA | 0 $V_{DD}$ = 3 Max |

1. Functional test specified $V_{IH}/V_{IL}$ for each mode.

**Table 6.6 Input Capacitance**

| Symbol | Parameter | Min | Max | Unit | Test Conditions |
|---|---|---|---|---|---|
| $C_I$ | Input capacitance of input pads | – | 7 | pF | Guaranteed by design |
| $C_{IO}$ | Input capacitance of I/O pads | – | 15 | pF | Guaranteed by design |
| $C_{PCI}$ | Input capacitance of PCI pads | – | 8 | pF | Guaranteed by design |
| $C_{LVD}$ | Input capacitance of LVD pads | – | 8 | pF | 6.5 pf Pad<br>1.5 pf Package |

**Table 6.7 8 mA Bidirectional Signals—GPIO0_FETCH/, GPIO1_MASTER/, GPIO2, GPIO3, GPIO4[1]**

| Symbol | Parameter | Min | Max | Unit | Test Conditions |
|---|---|---|---|---|---|
| $V_{IH}$ | Input high voltage | 2.0 | 5.55 | V | – |
| $V_{IL}$ | Input low voltage | –0.3 | 0.8 | V | – |
| $V_{OH}$ | Output high voltage | 2.4 | $V_{DD}$ | V | –8 mA |
| $V_{OL}$ | Output low voltage | $V_{SS}$ | 0.4 | V | 8 mA |
| $I_{OZ}$ | 3-state leakage | –10 | 10 | µA | – |
| $I_{PULL}$ | Pull up current | 25 | – | µA | – |

1. For channels A and B (except MAD[7:0]).

**Table 6.8 4 mA Bidirectional Signals—MAD[7:0]**

| Symbol | Parameter | Min | Max | Unit | Test Conditions |
|---|---|---|---|---|---|
| $V_{IH}$ | Input high voltage | 2.0 | 5.55 | V | – |
| $V_{IL}$ | Input low voltage | –0.3 | 0.8 | V | – |
| $V_{OH}$ | Output high voltage | 2.4 | $V_{DD}$ | V | –4 mA |
| $V_{OL}$ | Output low voltage | $V_{SS}$ | 0.4 | V | 4 mA |
| $I_{OZ}$ | 3-state leakage | –10 | 10 | µA | – |
| $I_{PULL}$ | Pull down current | 25 | – | µA | – |

**Table 6.9 4 mA Output Signals—MAS/[1:0], MCE/, MOE/_TESTOUT[1], MWE/, TDO**

| Symbol | Parameter | Min | Max | Unit | Test Conditions |
|---|---|---|---|---|---|
| $V_{OH}$ | Output high voltage | 2.4 | $V_{DD}$ | V | –4 mA |
| $V_{OL}$ | Output low voltage | $V_{SS}$ | 0.4 | V | 4 mA |
| $I_{OZ}$ | 3-state leakage | –10 | 10 | µA | – |

1. MOE/_TESTOUT is not tested for 3-state leakage. It cannot be 3-stated.

**Table 6.10 8 mA PCI Bidirectional Signals—AD[63:0], C_BE[7:0]/, FRAME/, IRDY/, TRDY/, DEVSEL/, STOP/, PERR/, PAR, PAR64, REQ64/, ACK64/**

| Symbol | Parameters | Min | Max | Unit | Test Conditions |
|---|---|---|---|---|---|
| $V_{IH}$ | Input high voltage | 0.5 $V_{DD}$ | $V_{DD}$ +0.5 | V | – |
| $V_{IL}$ | Input low voltage | –0.5 | 0.3 $V_{DD}$ | V | – |
| $V_{OH}$ | Output high voltage | 0.9 $V_{DD}$ | $V_{DD}$ | V | –8 mA |
| $V_{OL}$ | Output low voltage | $V_{SS}$ | 0.1 $V_{DD}$ | V | 8 mA |
| $I_{OZ}$ | 3-state leakage | –10 | 10 | µA | – |
| $I_{PULL\text{-}DOWN}$[1] | Pull down current | 25 | – | µA | – |

1. Pull-down text does not apply to AD[31:0] and C_BE[3:0]/.

**Table 6.11 Input Signals—CLK, GNT/, IDSEL, INT_DIR, RST/, SCLK, TCK, TDI, TEST_HSC, TEST_RST/, TEST_PD, TMS**

| Symbol | Parameters | Min | Max | Unit | Test Conditions |
|---|---|---|---|---|---|
| $V_{IH}$ | Input high voltage | 2.0 | 5.55 | V | – |
| $V_{IL}$ | Input low voltage | –0.3 | 0.8 | V | – |
| $I_{IN}$ | 3-state leakage | –10 | 10 | µA | – |
| $I_{PULL\text{-}UP}$[1] | Pull current | 25 | – | µA | – |

1. Pull-up text does not apply to CLK, GNT/, IDSEL, RST/, and SCLK.

**Table 6.12 8 mA Output Signals—INTA, INTB, ALT_INTA, ALT_INTB, REQ/, SERR/**

| Symbol | Parameters | Min | Max | Unit | Test Conditions |
|---|---|---|---|---|---|
| $V_{OH}$ | Output high voltage | 0.9 $V_{DD}$ | $V_{DD}$ | V | –8 mA |
| $V_{OL}$ | Output low voltage | $V_{DD}$ | 0.1 $V_{DD}$ | V | 8 mA |
| $I_{OZ}$ | 3-state leakage | –10 | 10 | µA | – |
| $I_{PULL-UP}$[1] | Pull current | 25 | – | µA | – |

1. Pull-up text does not apply o REQ/, SERR/.

## 6.2 TolerANT Technology Electrical Characteristics

**Table 6.13 TolerANT Technology Electrical Characteristics for SE SCSI Signals[1]**

| Symbol | Parameter | Min | Max | Units | Test Conditions |
|---|---|---|---|---|---|
| $V_{OH}$[2] | Output high voltage | 2.2 | 3.7 | V | $I_{OH}$ = 7 mA |
| $V_{OL}$ | Output low voltage | 0.0 | 0.5 | V | $I_{OL}$ = 48 mA |
| $V_{IH}$ | Input high voltage | 1.9 | – | V | – |
| $V_{IL}$ | Input low voltage | – | 1.0 | V | Referenced to $V_{SS}$ |
| $V_{IK}$ | Input clamp voltage | –0.66 | –0.77 | V | $V_{PP}$ = 4.75 V; $I_1$ = –20 mA |
| $V_{TH}$ | Threshold, HIGH to LOW | 1.15 | 1.25 | V | – |
| $V_{TL}$ | Threshold, LOW to HIGH | 1.55 | 1.65 | V | – |
| $V_{TH}$-$V_{TL}$ | Hysteresis | 300 | 500 | mV | – |
| $I_{OH}$[2] | Output high current | 0 | 7 | mA | $V_{OH}$ = 2.4 V |
| $I_{OL}$ | Output low current | 48 | – | mA | $V_{OL}$ = 0.5 V |
| $I_{OSH}$[2] | Short-circuit output high current | 48 | – | mA | Short to $V_{DD}$[3] |
| $I_{OSL}$ | Short-circuit output low current | 22 | – | mA | Short to $V_{SS}$ |
| $I_{LH}$ | Input high leakage | – | 20 | µA | –0.5<$V_{DD}$<5.25 $V_{PIN}$ = 2.7 V |
| $I_{LL}$ | Input low leakage | – | 20 | µA | –0.5<$V_{DD}$<5.25 $V_{PIN}$ = 0.5 V |
| $R_I$ | Input resistance | 20 | – | MΩ | SCSI pins |
| $C_P$ | Capacitance per pin | – | 8 | pF | PQFP |
| $t_R$[2] | Rise time, 10% to 90% | 6.7 | 14.7 | ns | Figure 6.3 |
| $t_F$ | Fall time, 90% to 10% | 5.7 | 17.2 | ns | Figure 6.3 |
| $dV_H/dt$ | Slew rate LOW to HIGH | 100 | 470 | mV/ns | Figure 6.3 |
| $dV_L/dt$ | Slew rate HIGH to LOW | 110 | 440 | mV/ns | Figure 6.3 |

**Table 6.13 TolerANT Technology Electrical Characteristics for SE SCSI Signals[1]**

| Symbol | Parameter | Min | Max | Units | Test Conditions |
|---|---|---|---|---|---|
| ESD | Electrostatic discharge | 2 | – | kV | MIL-STD-883C; 3015-7 |
| | Latch-up | 100 | – | mA | – |
| | Filter delay | 20 | 30 | ns | Figure 6.4 |
| | Ultra filter delay | 10 | 15 | ns | Figure 6.4 |
| | Ultra2 filter delay | 5 | 8 | ns | Figure 6.4 |
| | Extended filter delay | 40 | 60 | ns | Figure 6.4 |

1. These values are guaranteed by periodic characterization; they are not 100% tested on every device.
2. Active negation outputs only: Data, Parity, SREQ/, SACK/. (Minus Pins) SCSI mode only.
3. Single pin only; irreversible damage may occur if sustained for one second.

**Figure 6.3 Rise and Fall Time Test Condition**

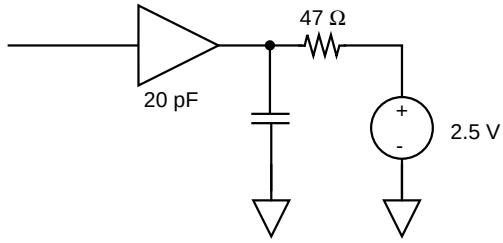
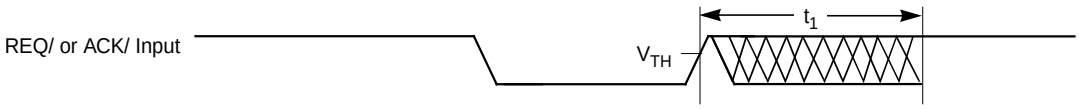



**Figure 6.4 SCSI Input Filtering**




Note: $t_1$ is the input filtering period.

**Figure 6.5 Hysteresis of SCSI Receivers**

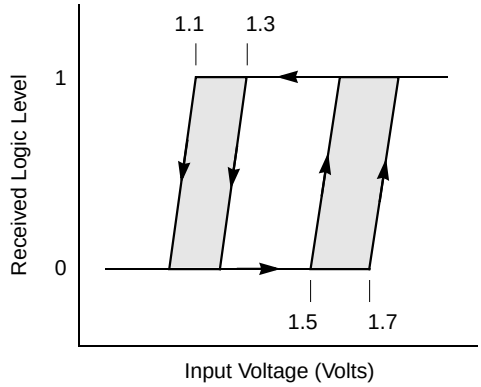



**Figure 6.6 Input Current as a Function of Input Voltage**

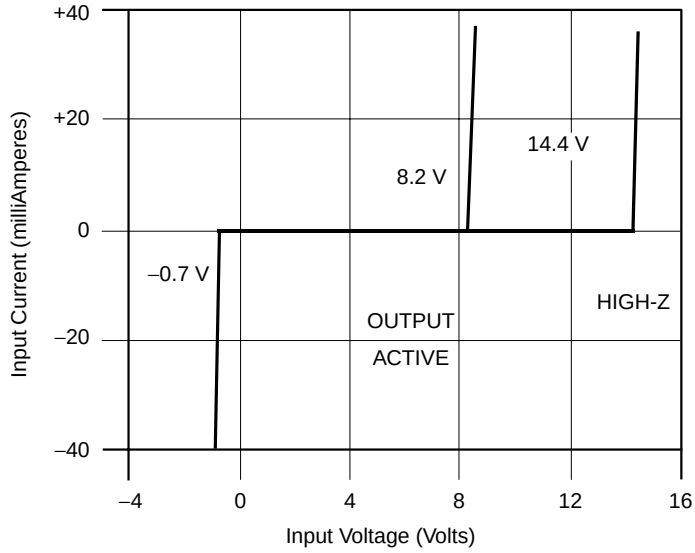
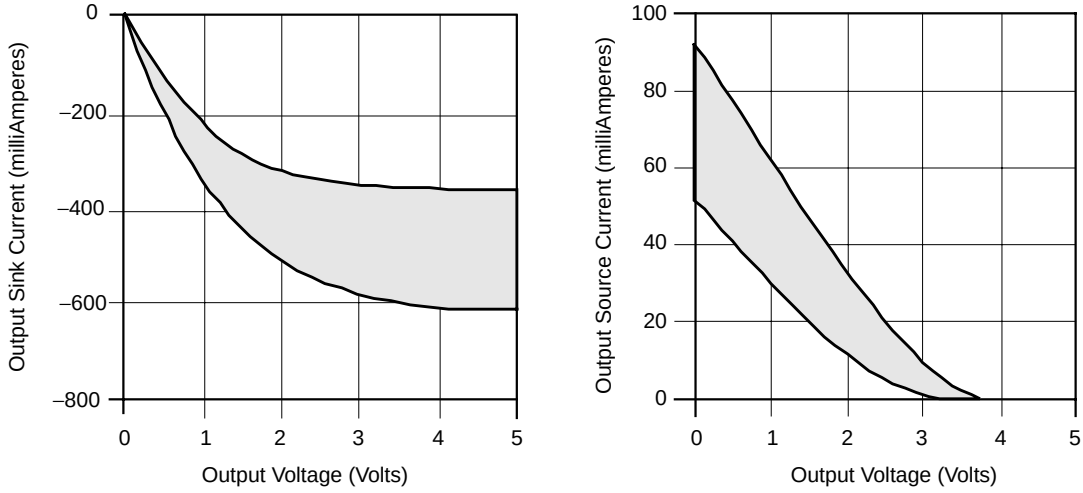


**Figure 6.7 Output Current as a Function of Output Voltage**



## 6.3 AC Characteristics

The AC characteristics described in this section apply over the entire range of operating conditions (refer to Section 6.1, “DC Characteristics”). Chip timing is based on simulation at worst case voltage, temperature, and processing. Timing was developed with a load capacitance of 50 pF.

**Table 6.14 External Clock**

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $t_1$ | PCI Bus clock period | 30 | DC | ns |
| | SCSI clock period[1] | 25 | 25 | ns |
| $t_2$ | PCI CLK LOW time[2] | 11 | – | ns |
| | SCLK LOW time[2] | 10 | 15 | ns |
| $t_3$ | PCI CLK HIGH time[2] | 11 | – | ns |
| | SCLK HIGH time[2] | 10 | 15 | ns |
| $t_4$ | PCI CLK slew rate | 1 | 4 | V/ns |
| | PCI Clock Spread Spectrum Modulation Frequency | 30 | 33 | kHz |
| | PCI Clock Spread Spectrum Frequency Spread | –1 | 9 | % |

1. Timing is for an external 40 MHz clock. A quadrupled 40 MHz clock is required for Ultra3 SCSI operation.
2. Duty cycle not to exceed 60/40.

**Figure 6.8 External Clock**

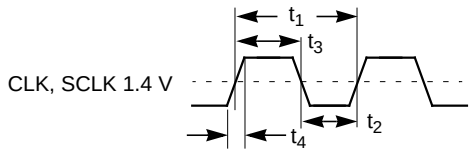


**Table 6.15 Reset Input**

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $t_1$ | Reset pulse width | 10 | – | $t_{CLK}$ |
| $t_2$ | Reset deasserted setup to CLK HIGH | 0 | – | ns |
| $t_3$ | MAD setup time to CLK HIGH (for configuring the MAD bus only) | 20 | – | ns |
| $t_4$ | MAD hold time from CLK HIGH (for configuring the MAD bus only) | 20 | – | ns |

**Figure 6.9 Reset Input**

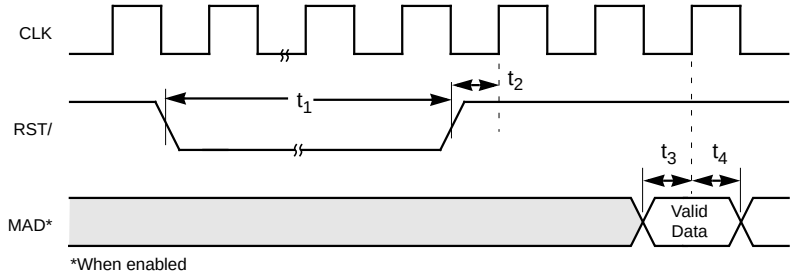
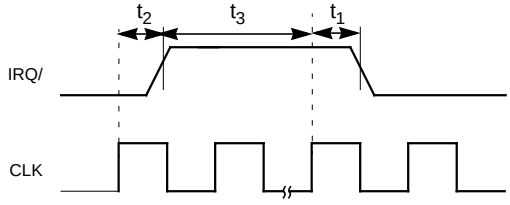



**Table 6.16 Interrupt Output**

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $t_1$ | CLK HIGH to IRQ/ LOW | 2 | 11 | ns |
| $t_2$ | CLK HIGH to IRQ/ HIGH | 2 | 11 | ns |
| $t_3$ | IRQ/ deassertion time | 3 | – | CLK |

**Figure 6.10 Interrupt Output**



## 6.4 PCI and External Memory Interface Timing Diagrams

Figure 6.11 through Figure 6.34 represent signal activity when the SYM53C1010-33 accesses the PCI bus. This section includes timing diagrams for access to three groups of memory configurations. The first group applies to Target Timing. The second group applies to Initiator Timing. The third group applies to External Memory Timing.

> Note: Multiple byte accesses to the external memory bus increase the read or write cycle by 11 clocks for each additional byte.

Timing diagrams included in this section are:

- Target Timing
  - PCI Configuration Register Read
  - PCI Configuration Register Write
  - 32-Bit Operating Registers/SCRIPTS RAM Read
  - 64-Bit Operating Register/SCRIPTS RAM Read
  - 32-Bit Operating Register/SCRIPTS RAM Write
  - 64-Bit Operating Register/SCRIPTS RAM Write
- Initiator Timing
  - Nonburst Opcode Fetch, 32-Bit Address and Data
  - Burst Opcode Fetch, 32-Bit Address and Data
  - Back-to-Back Read, 32-Bit Address and Data
  - Back-to-Back Write, 32-Bit Address and Data
  - Burst Read, 32-Bit Address and Data
  - Burst Read, 64-Bit Address and Data
  - Burst Write, 32-Bit Address and Data
  - Burst Write, 64-Bit Address and Data
- External Memory Timing
  - External Memory Read
  - External Memory Write

- Normal/Fast Memory (≥ 128 Kbytes) Single Byte Access Read Cycle
- Normal/Fast Memory (≥ 128 Kbytes) Singe Byte Access Write Cycle
- Slow Memory (≥ 128 Kbytes) Read Cycle
- Slow Memory (≥ 128 Kbytes) Write Cycle
- ≤ 64 Kbytes ROM Read Cycle
- ≤ 64 Kbytes ROM Write Cycle

## 6.4.1 Target Timing

**Table 6.17 PCI Configuration Register Read**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_1$ | Shared signal input setup time | 7 | – | ns |
| $t_2$ | Shared signal input hold time | 0 | – | ns |
| $t_3$ | CLK to shared signal output valid | 2 | 11 | ns |

**Figure 6.11 PCI Configuration Register Read**

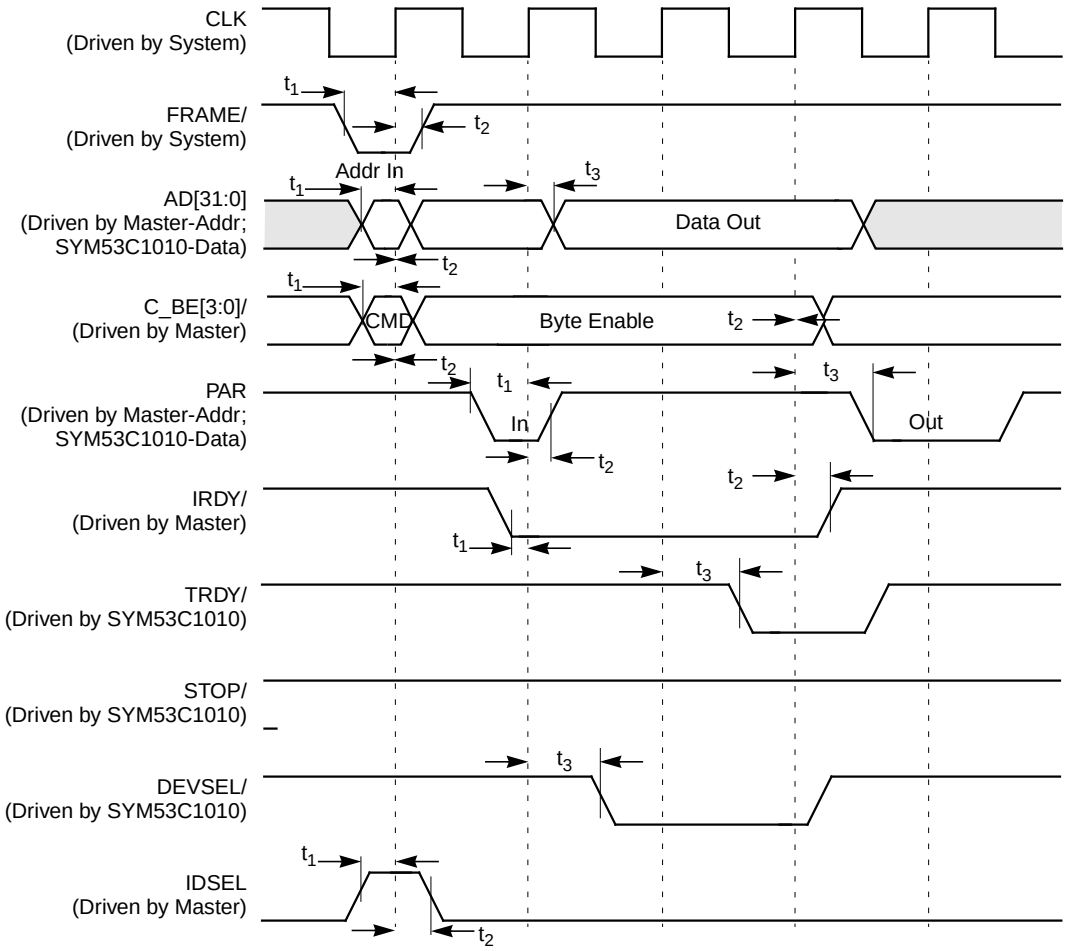


**Table 6.18 PCI Configuration Register Write**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_1$ | Shared signal input setup time | 7 | – | ns |
| $t_2$ | Shared signal input hold time | 0 | – | ns |
| $t_3$ | CLK to shared signal output valid | 2 | 11 | ns |

**Figure 6.12 PCI Configuration Register Write**

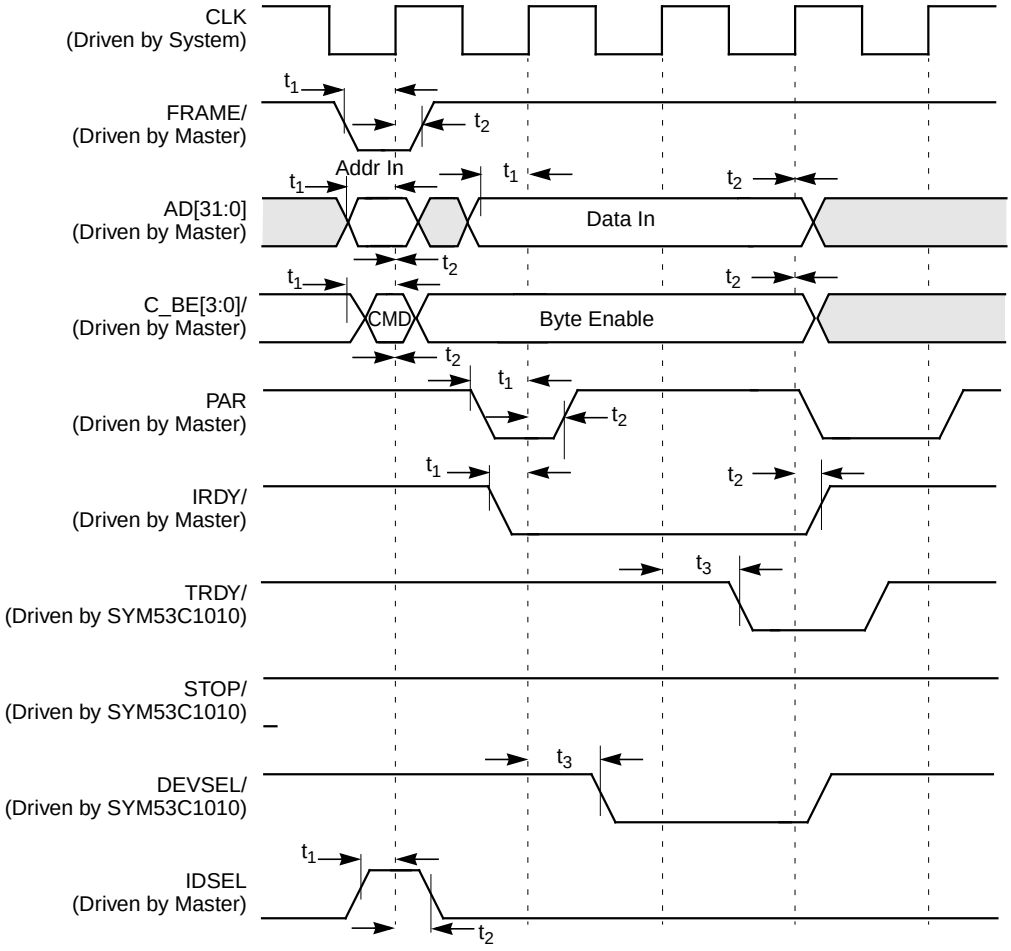


**Table 6.19 32-Bit Operating Register/SCRIPTS RAM Read**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_1$ | Shared signal input setup time | 7 | – | ns |
| $t_2$ | Shared signal input hold time | 0 | – | ns |
| $t_3$ | CLK to shared signal output valid | 2 | 11 | ns |

**Figure 6.13 32-Bit Operating Registers/SCRIPTS RAM Read**

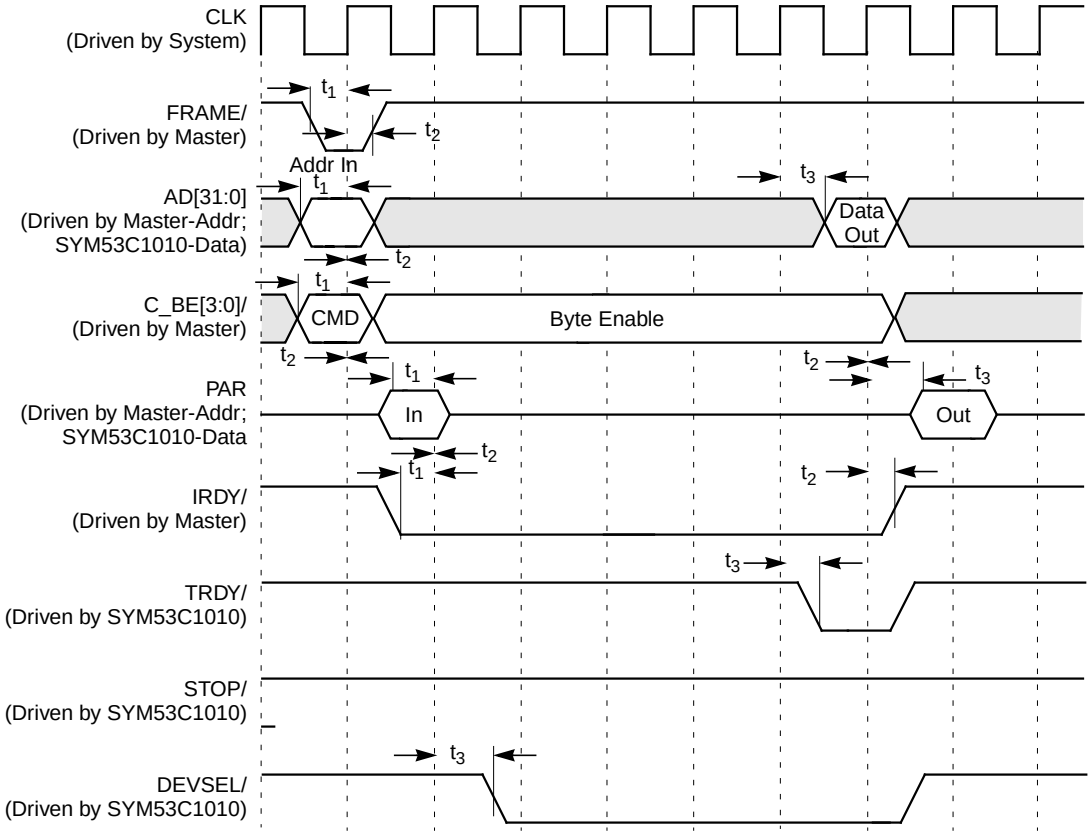


**Table 6.20 64-Bit Operating Register/SCRIPTS RAM Read**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_1$ | Shared signal input setup time | 7 | – | ns |
| $t_2$ | Shared signal input hold time | 0 | – | ns |
| $t_3$ | CLK to shared signal output valid | 2 | 11 | ns |

**Figure 6.14 64-Bit Operating Register/SCRIPTS RAM Read**

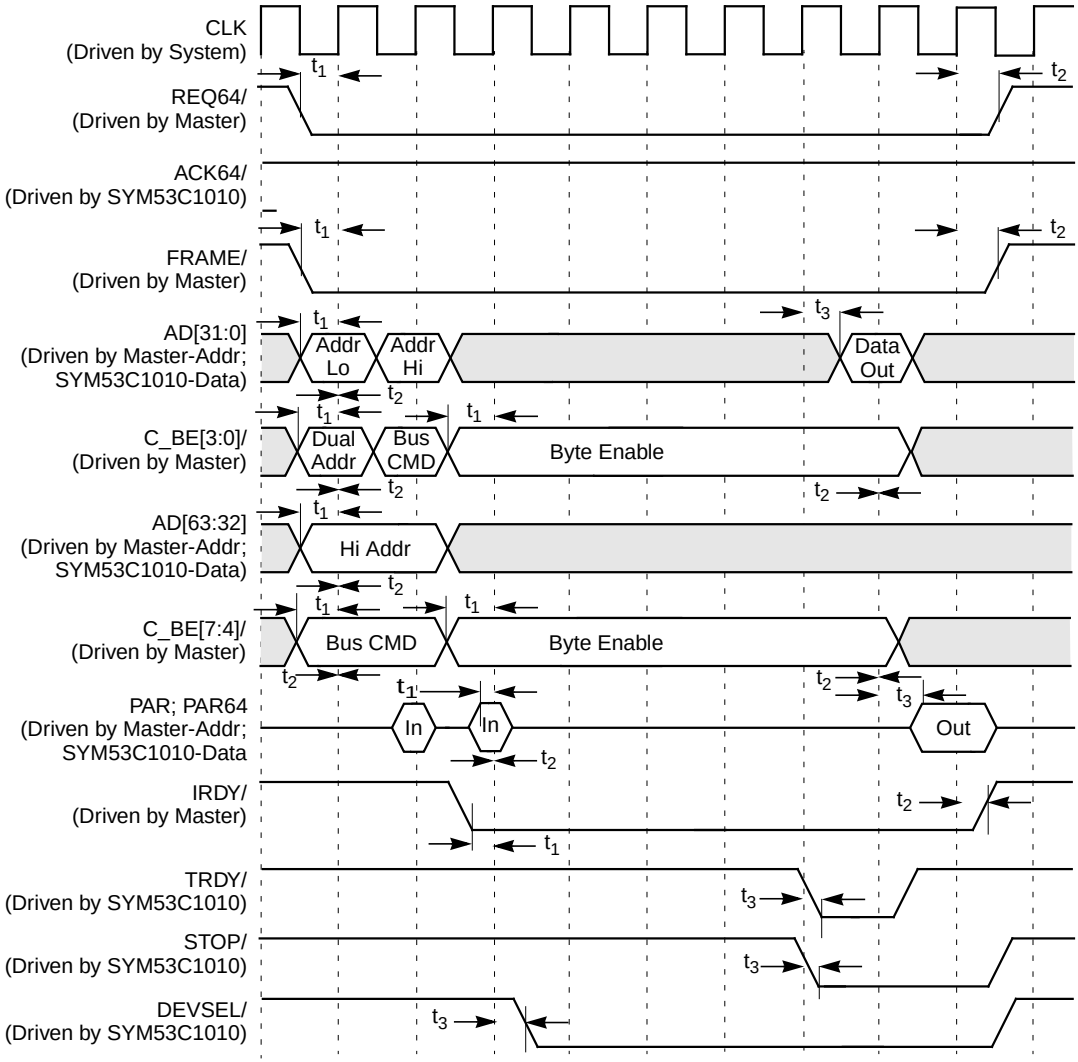


**Table 6.21 32-Bit Operating Register/SCRIPTS RAM Write**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_1$ | Shared signal input setup time | 7 | – | ns |
| $t_2$ | Shared signal input hold time | 0 | – | ns |
| $t_3$ | CLK to shared signal output valid | 2 | 11 | ns |

**Figure 6.15 32-Bit Operating Register/SCRIPTS RAM Write**

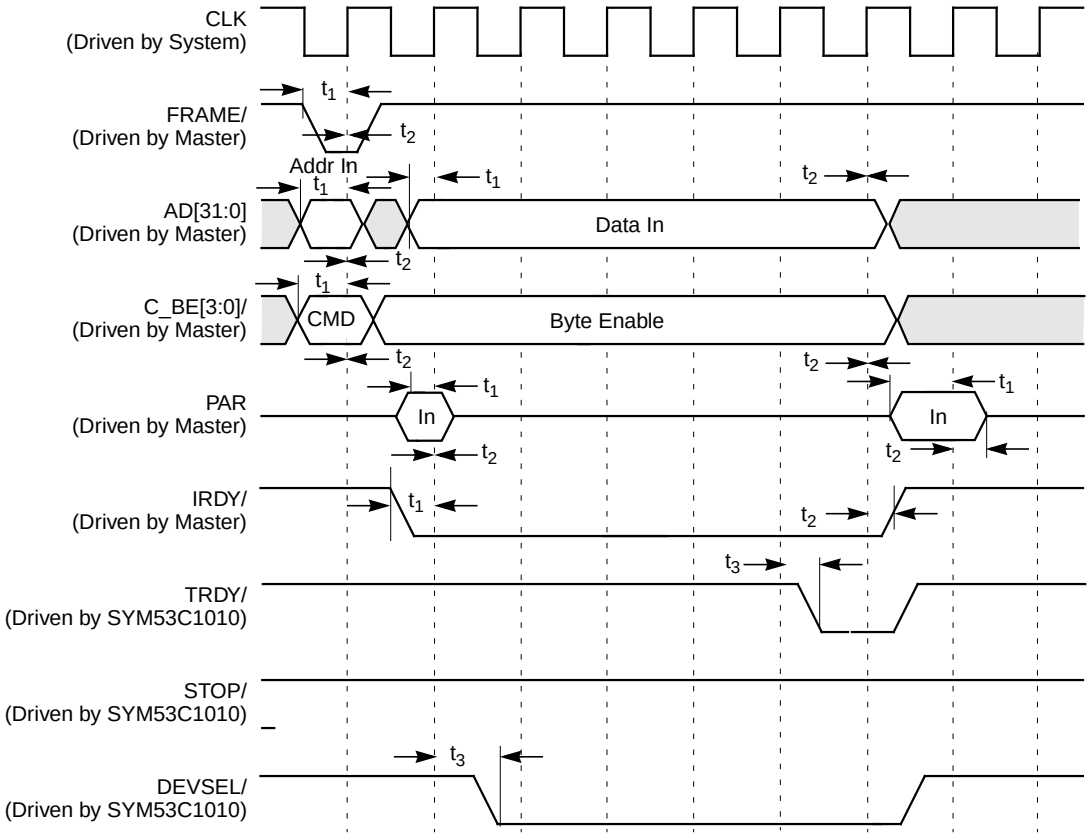
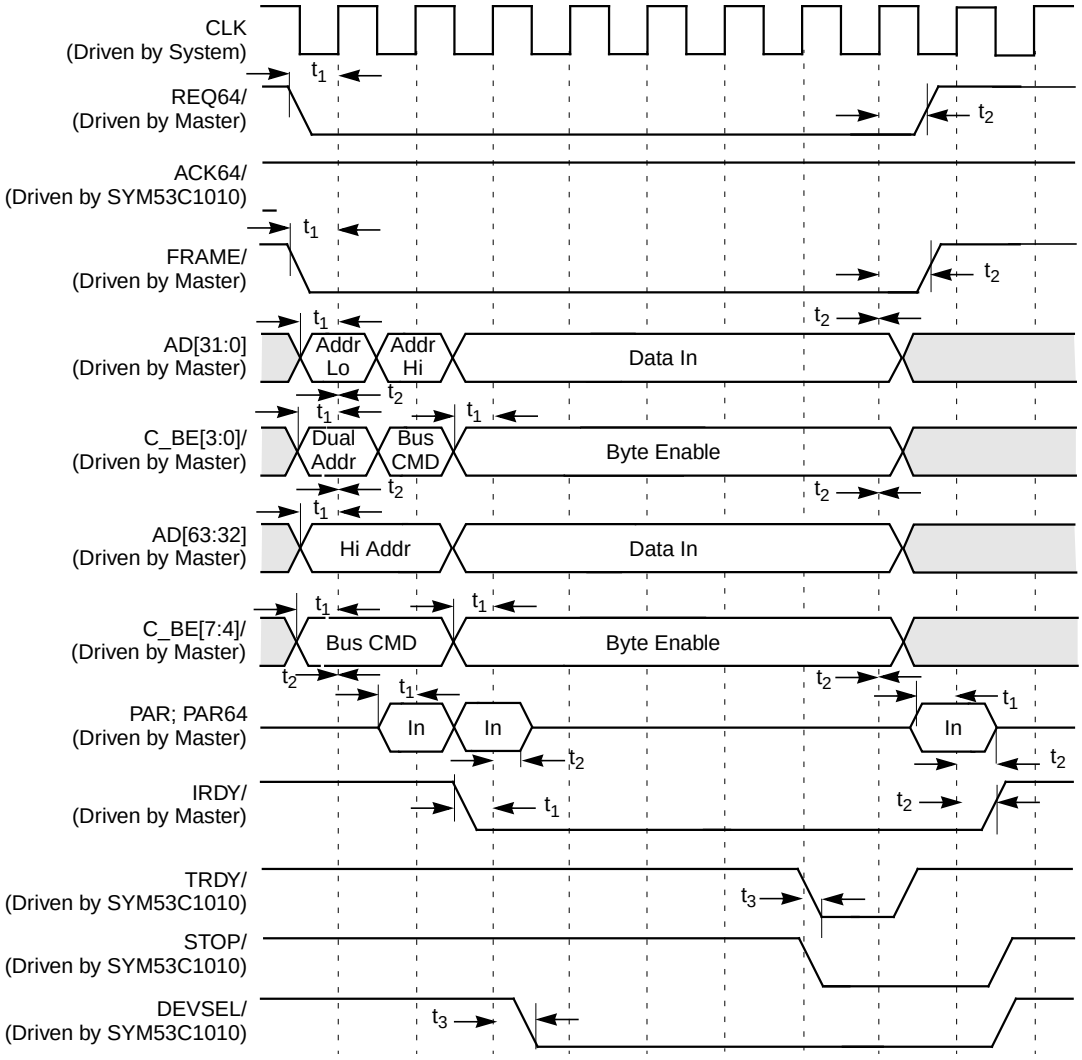


**Table 6.22 64-Bit Operating Register/SCRIPTS RAM Write**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_1$ | Shared signal input setup time | 7 | – | ns |
| $t_2$ | Shared signal input hold time | 0 | – | ns |
| $t_3$ | CLK to shared signal output valid | 2 | 11 | ns |

**Figure 6.16 64-Bit Operating Register/SCRIPTS RAM Write**



## 6.4.2 Initiator Timing

**Table 6.23 Nonburst Opcode Fetch, 32-Bit Address and Data**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_1$ | Shared signal input setup time | 7 | – | ns |
| $t_2$ | Shared signal input hold time | 0 | – | ns |
| $t_3$ | CLK to shared signal output valid | 2 | 11 | ns |
| $t_4$ | Side signal input setup time | 10 | – | ns |
| $t_5$ | Side signal input hold time | 0 | – | ns |
| $t_6$ | CLK to side signal output valid | 2 | 12 | ns |

**Figure 6.17 Nonburst Opcode Fetch, 32-Bit Address and Data**

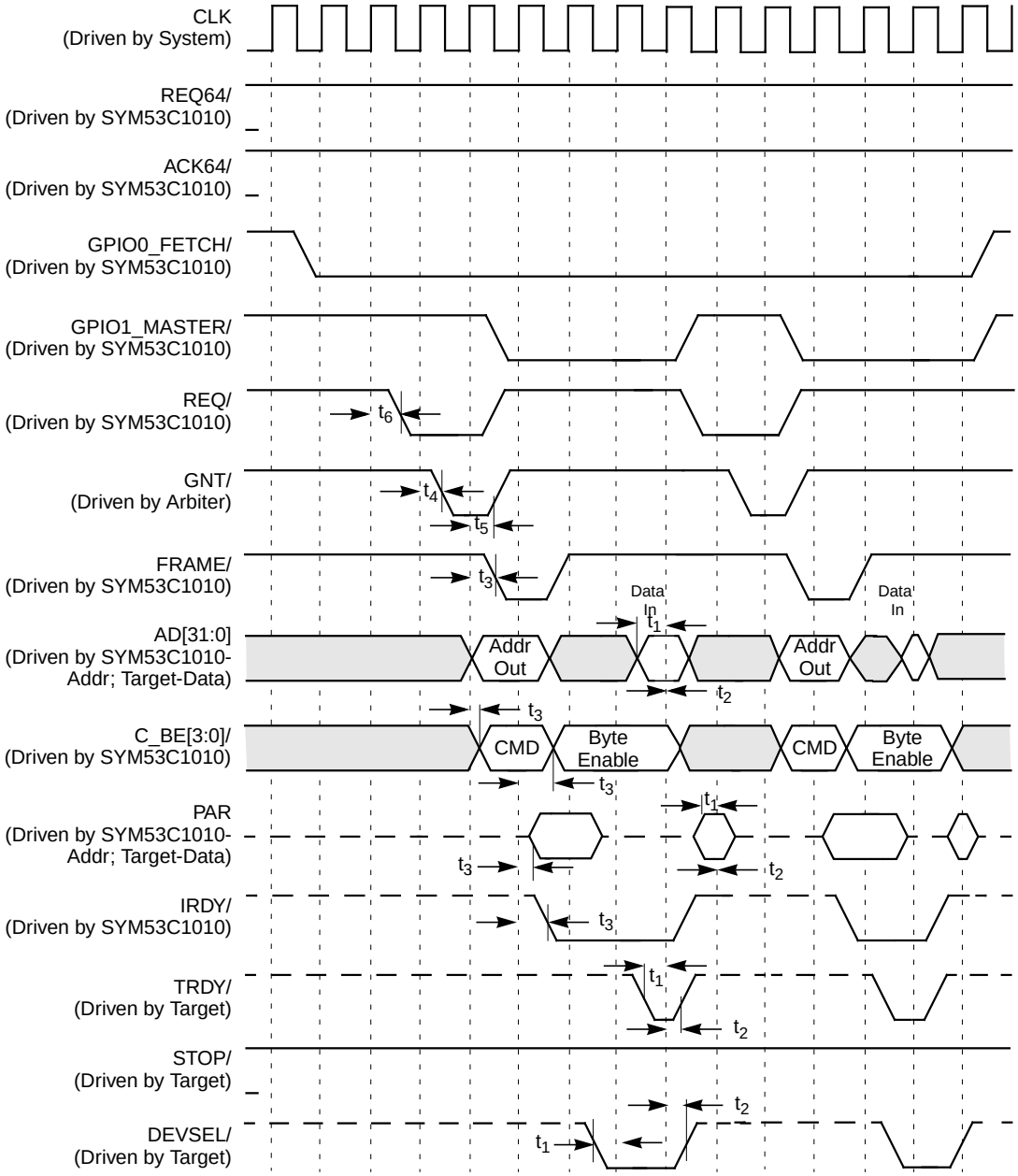


**Table 6.24 Burst Opcode Fetch, 32-Bit Address and Data**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_1$ | Shared signal input setup time | 7 | – | ns |
| $t_2$ | Shared signal input hold time | 0 | – | ns |
| $t_3$ | CLK to shared signal output valid | 2 | 11 | ns |
| $t_4$ | Side signal input setup time | 10 | – | ns |
| $t_5$ | Side signal input hold time | 0 | – | ns |
| $t_6$ | CLK to side signal output valid | 2 | 12 | ns |

**Figure 6.18 Burst Opcode Fetch, 32-Bit Address and Data**

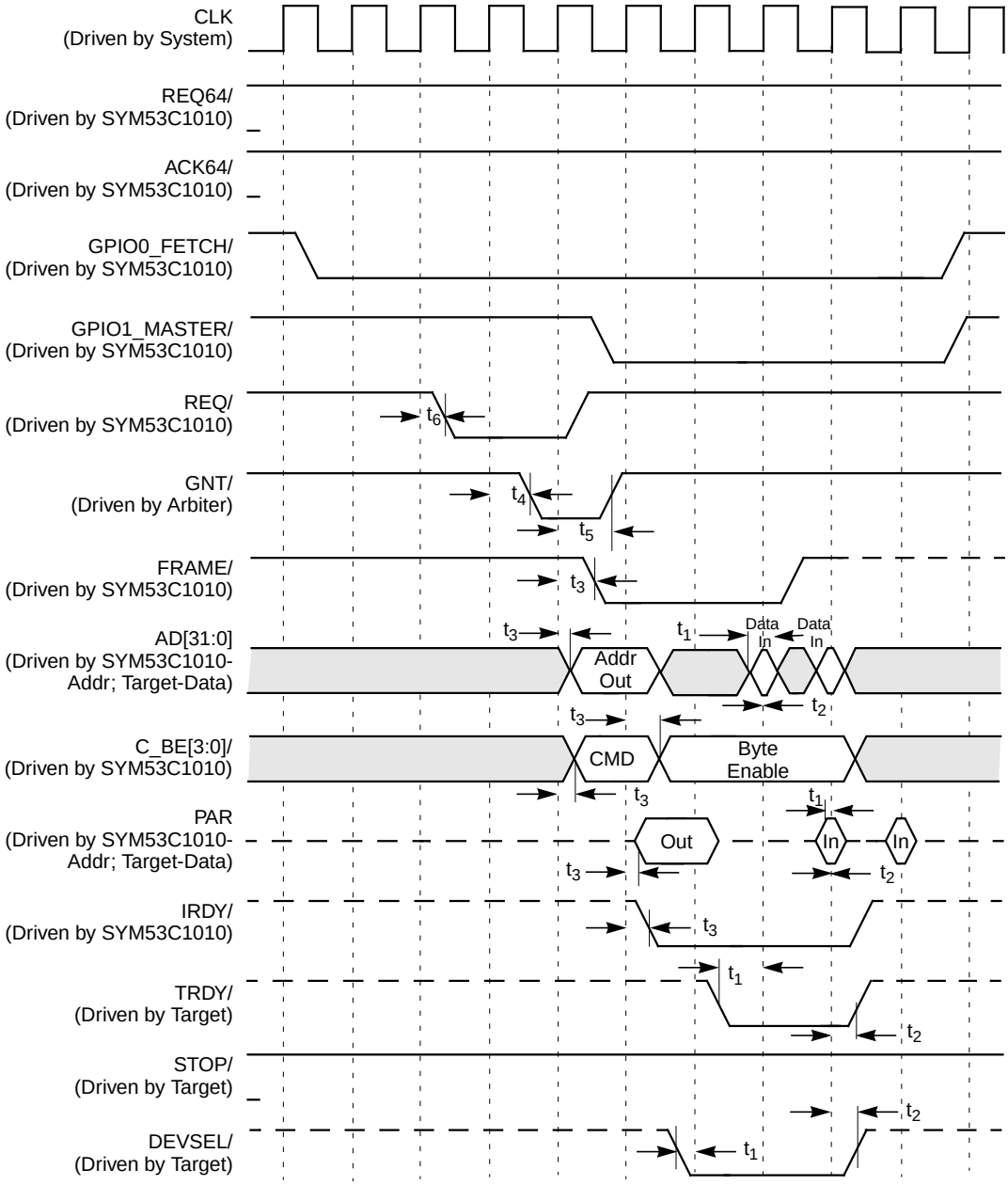


**Table 6.25 Back-to-Back Read, 32-Bit Address and Data**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_1$ | Shared signal input setup time | 7 | – | ns |
| $t_2$ | Shared signal input hold time | 0 | – | ns |
| $t_3$ | CLK to shared signal output valid | 2 | 11 | ns |
| $t_4$ | Side signal input setup time | 10 | – | ns |
| $t_5$ | Side signal input hold time | 0 | – | ns |
| $t_6$ | CLK to side signal output valid | 2 | 12 | ns |

**Figure 6.19 Back-to-Back Read, 32-Bit Address and Data**

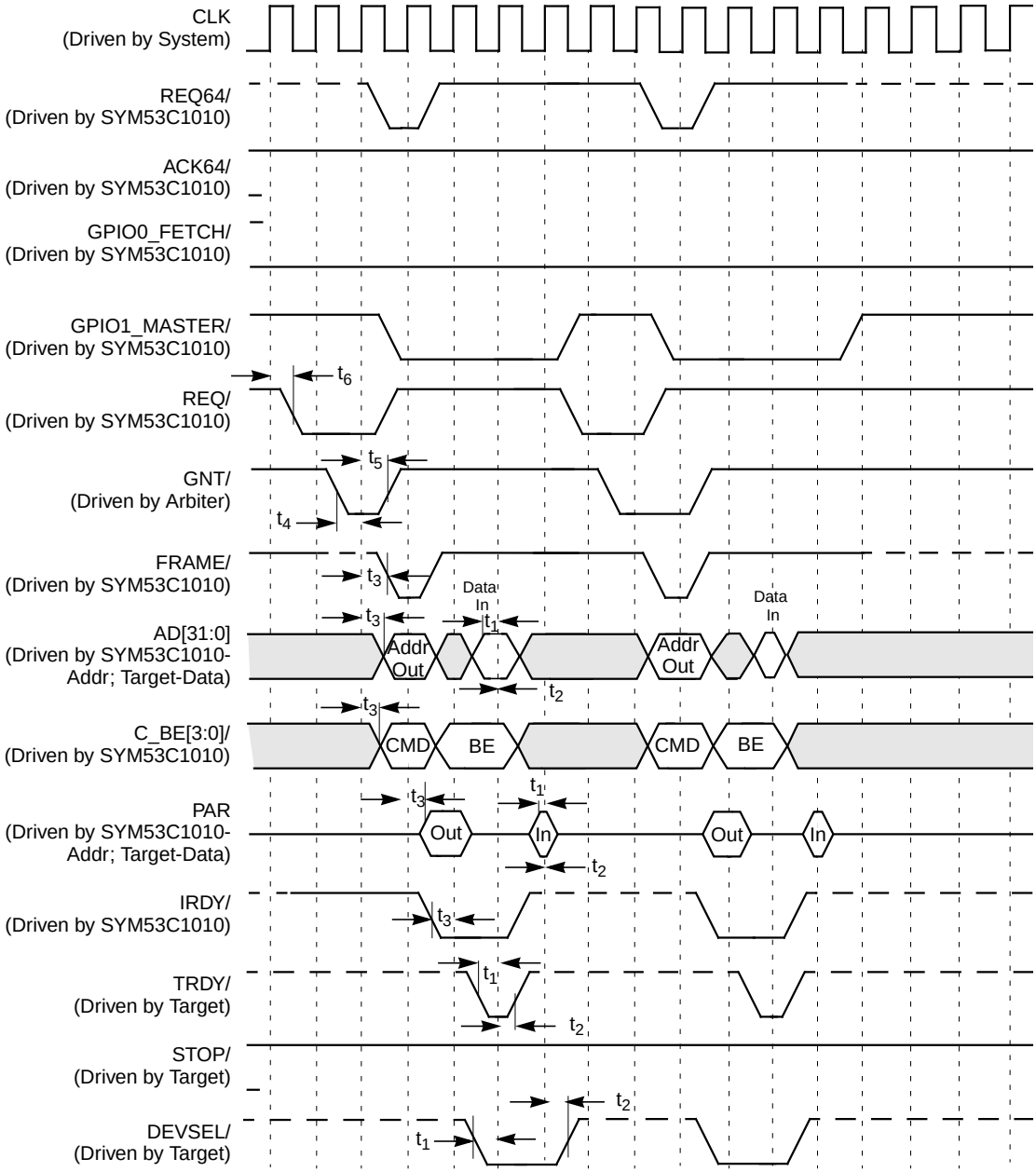


**Table 6.26 Back-to-Back Write, 32-Bit Address and Data**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_1$ | Shared signal input setup time | 7 | – | ns |
| $t_2$ | Shared signal input hold time | 0 | – | ns |
| $t_3$ | CLK to shared signal output valid | 2 | 11 | ns |
| $t_4$ | Side signal input setup time | 10 | – | ns |
| $t_5$ | Side signal input hold time | 0 | – | ns |
| $t_6$ | CLK to side signal output valid | 2 | 12 | ns |

**Figure 6.20 Back-to-Back Write, 32-Bit Address and Data**

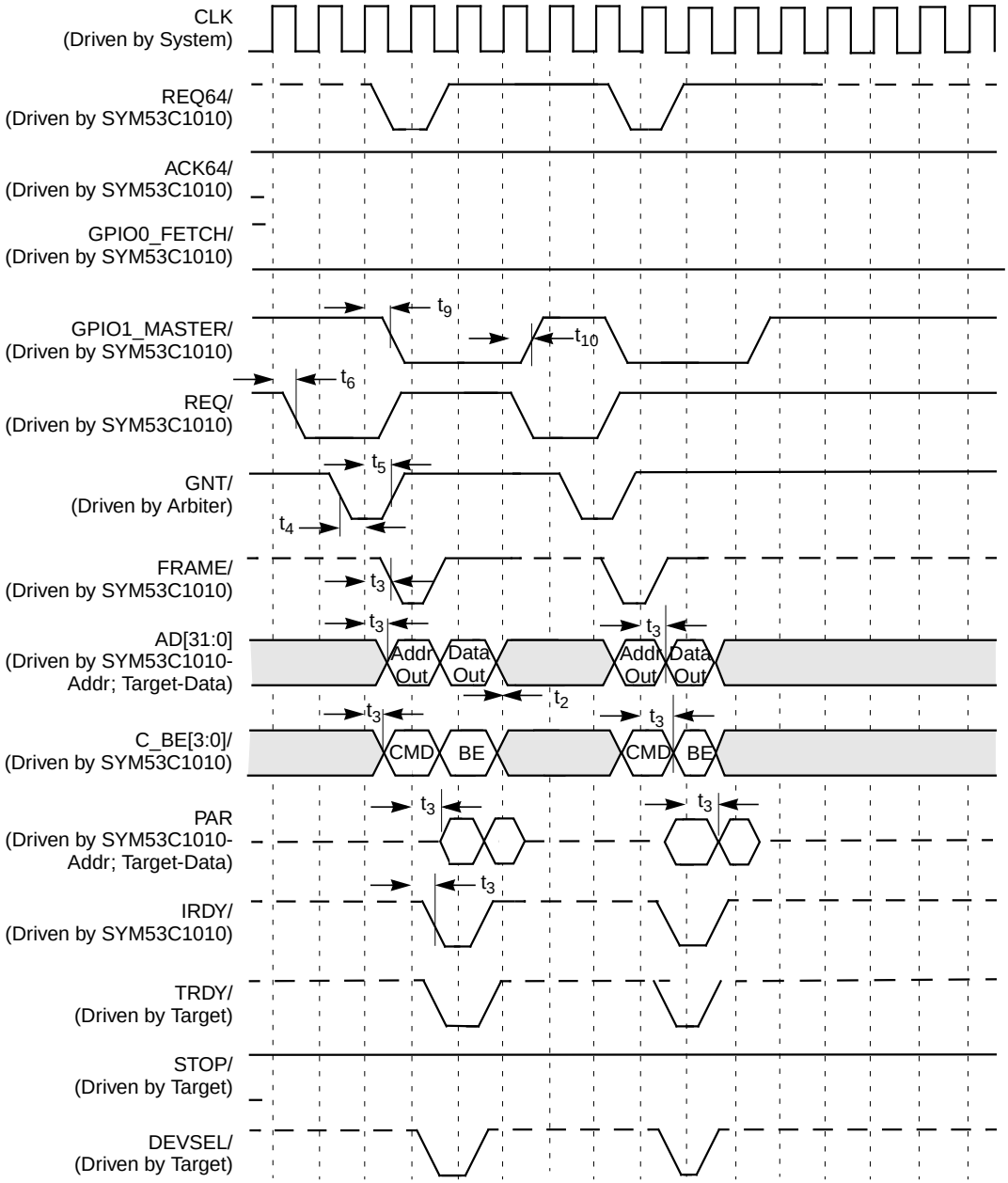


**Table 6.27 Burst Read, 32-Bit Address and Data**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_1$ | Shared signal input setup time | 7 | – | ns |
| $t_2$ | Shared signal input hold time | 0 | – | ns |
| $t_3$ | CLK to shared signal output valid | 2 | 11 | ns |

**Figure 6.21 Burst Read, 32-Bit Address and Data**

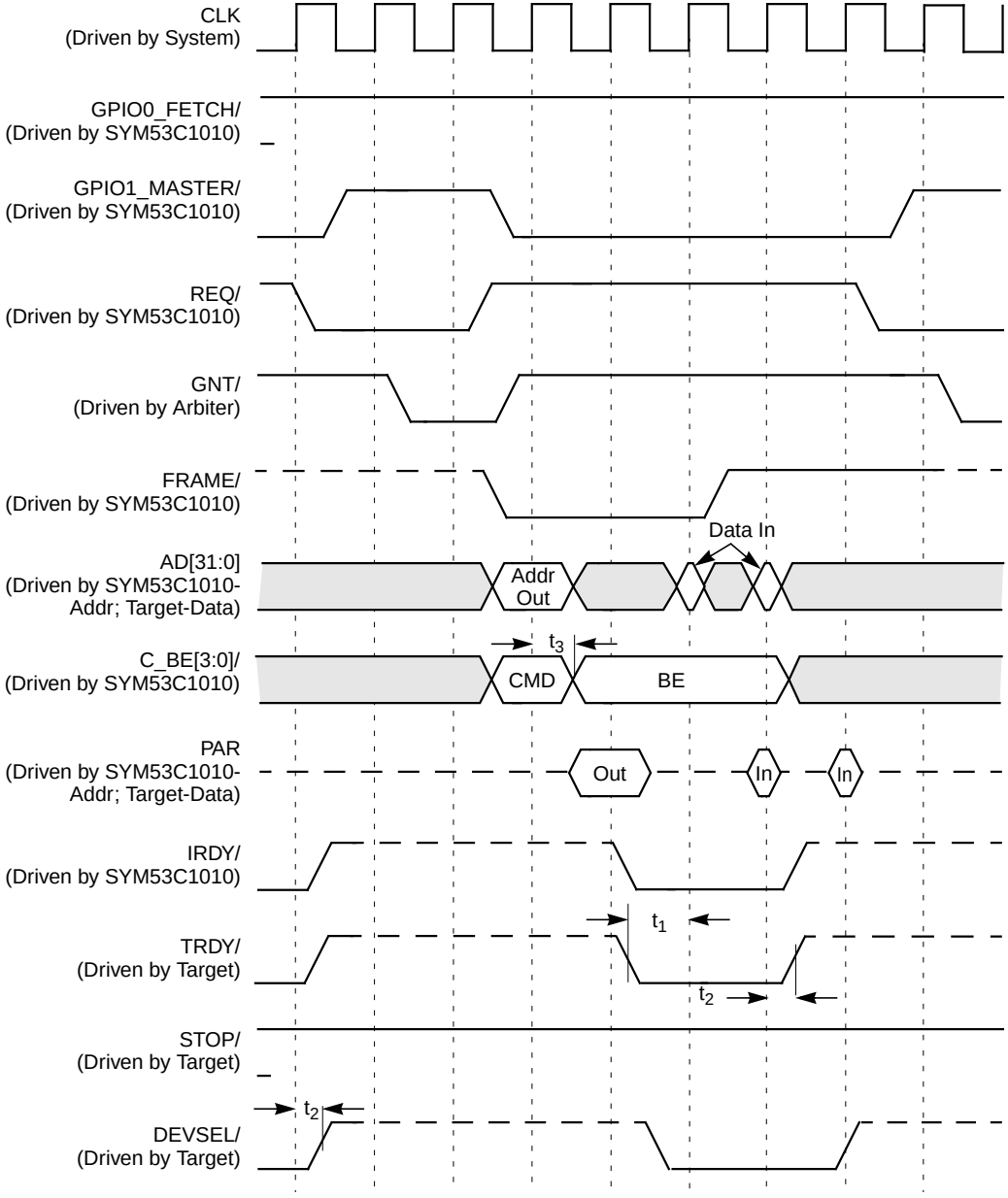


**Table 6.28 Burst Read, 64-Bit Address and Data**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_1$ | Shared signal input setup time | 7 | – | ns |
| $t_2$ | Shared signal input hold time | 0 | – | ns |
| $t_3$ | CLK to shared signal output valid | 2 | 11 | ns |

**Figure 6.22 Burst Read, 64-Bit Address and Data**

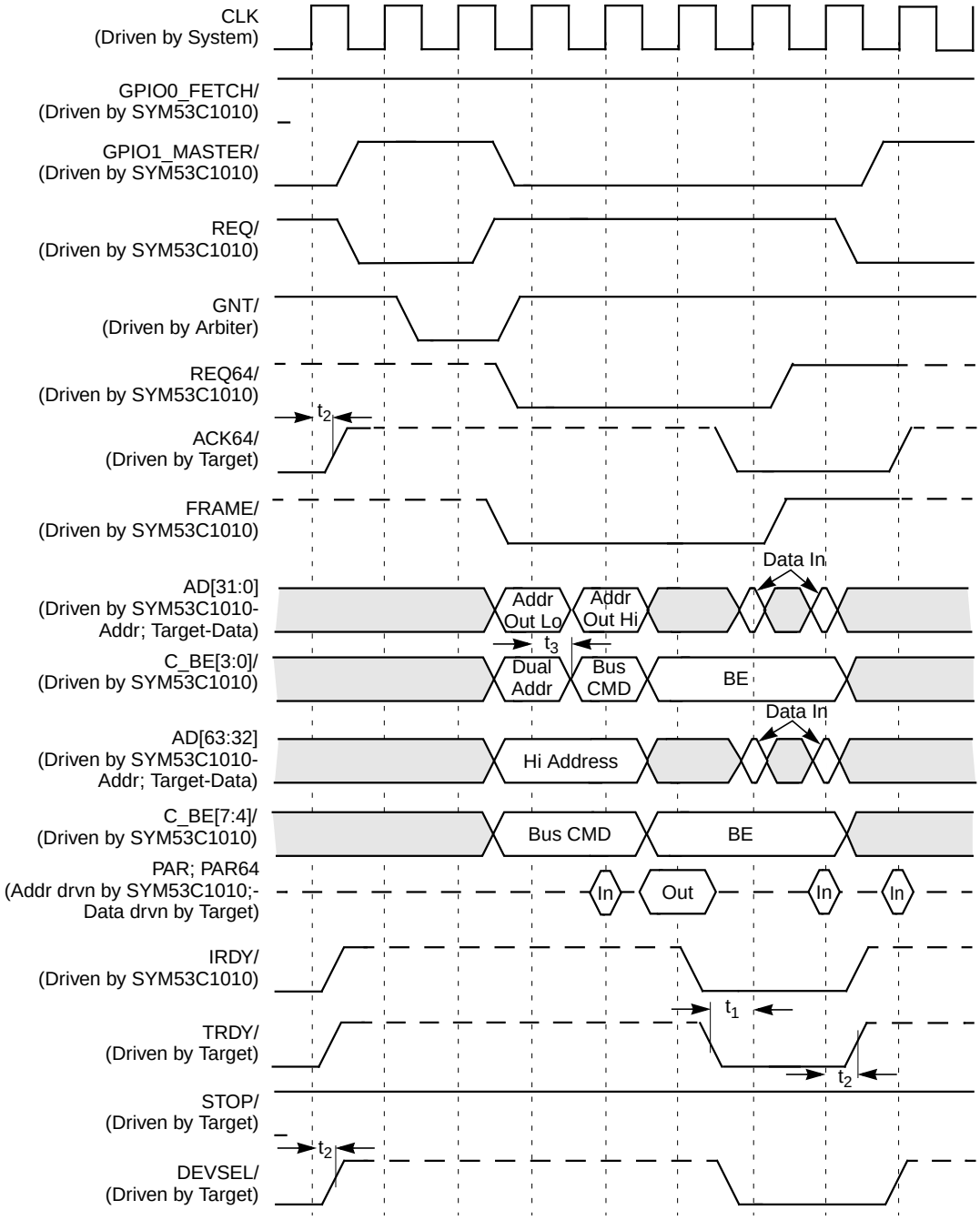


**Table 6.29 Burst Write, 32-Bit Address and Data**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_1$ | Shared signal input setup time | 7 | – | ns |
| $t_2$ | Shared signal input hold time | 0 | – | ns |
| $t_3$ | CLK to shared signal output valid | 2 | 11 | ns |

**Figure 6.23 Burst Write, 32-Bit Address and Data**

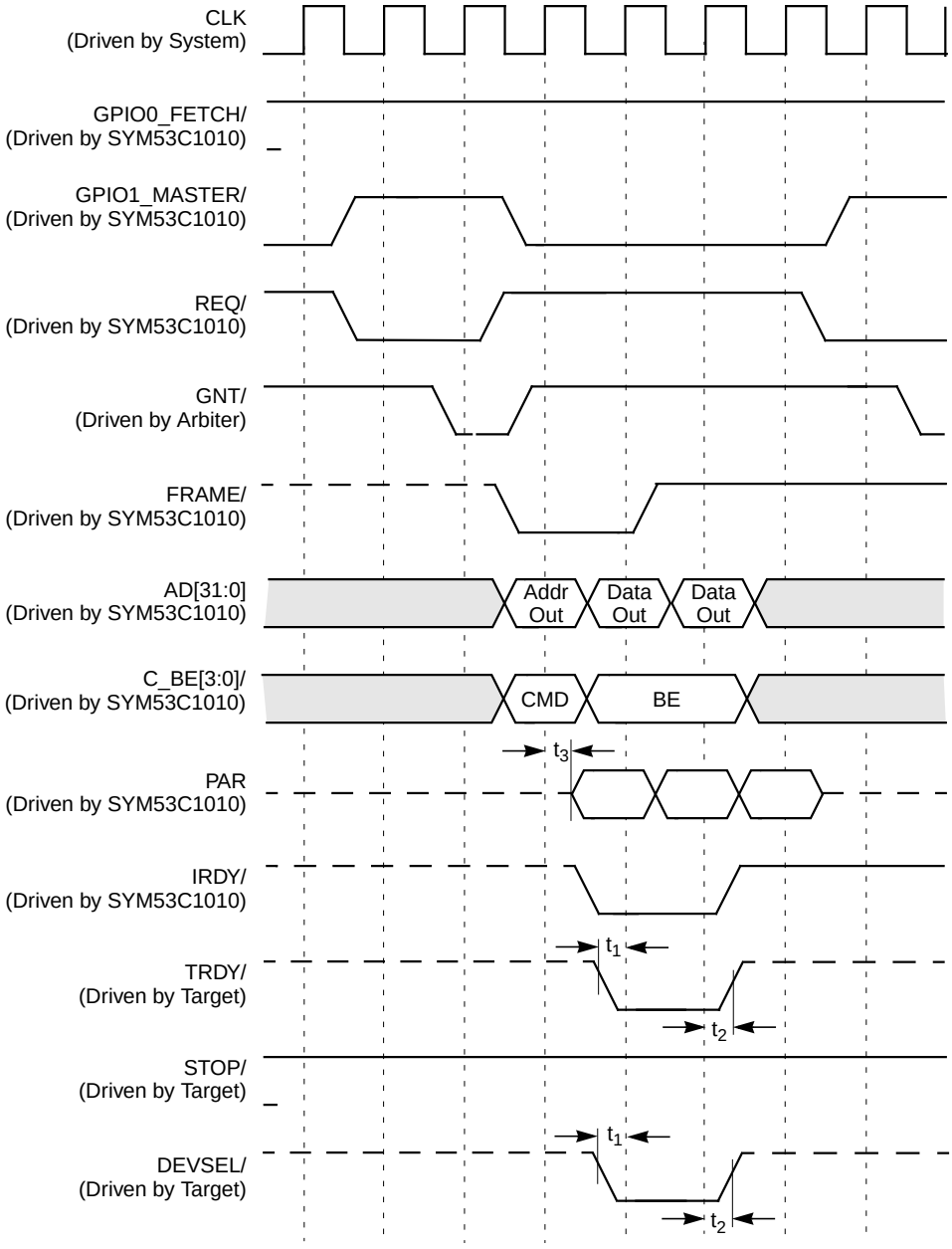
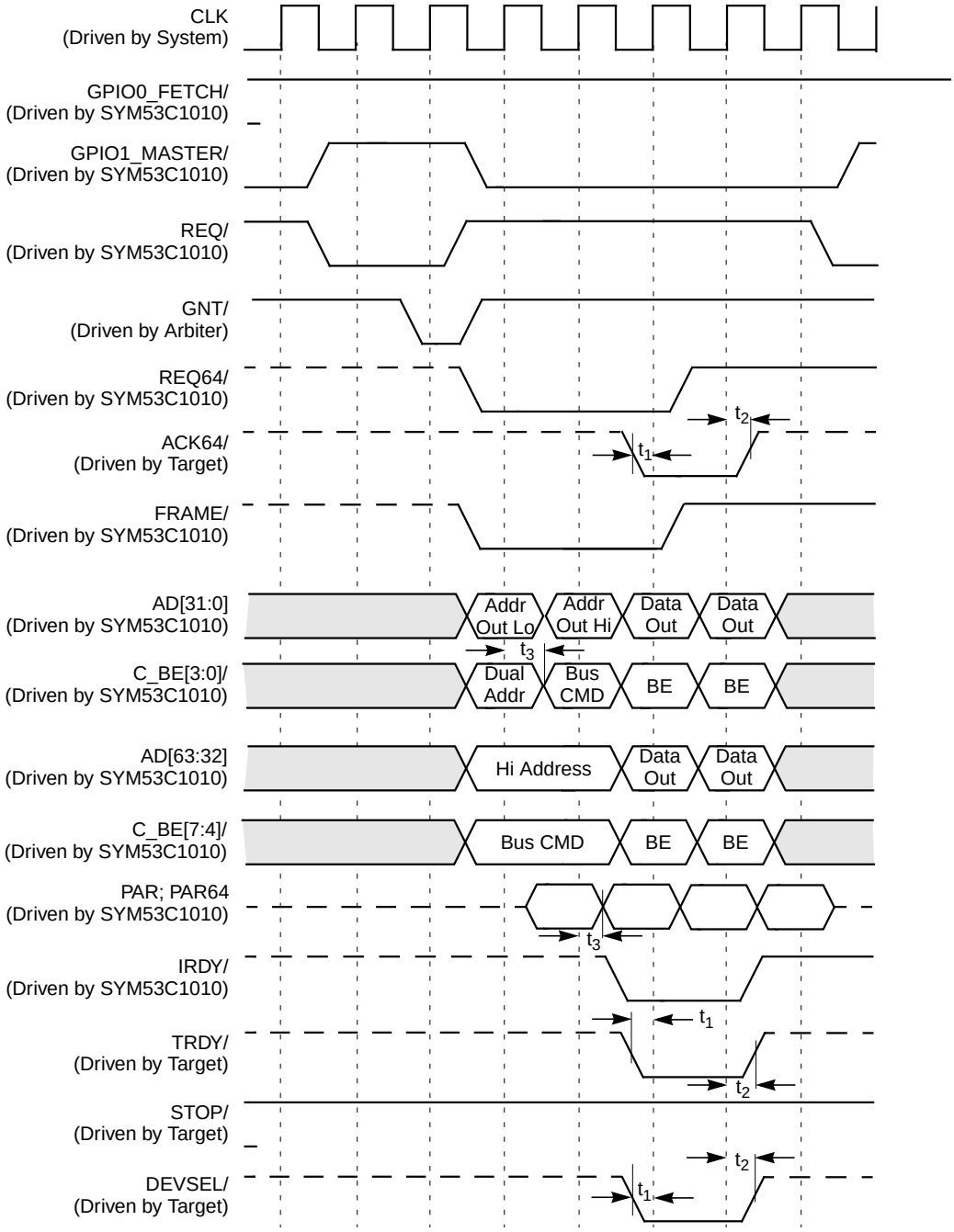


**Table 6.30 Burst Write, 64-Bit Address and Data**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_1$ | Shared signal input setup time | 7 | – | ns |
| $t_2$ | Shared signal input hold time | 0 | – | ns |
| $t_3$ | CLK to shared signal output valid | 2 | 11 | ns |

**Figure 6.24 Burst Write, 64-Bit Address and Data**



## 6.4.3 External Memory Timing

**Table 6.31 External Memory Read**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_1$ | Shared signal input setup time | 7 | – | ns |
| $t_2$ | Shared signal input hold time | 0 | – | ns |
| $t_3$ | CLK to shared signal output valid | 2 | 11 | ns |
| $t_{11}$ | Address setup to MAS/ HIGH | 25 | – | ns |
| $t_{12}$ | Address hold from MAS/ HIGH | 15 | – | ns |
| $t_{13}$ | MAS/ pulse width | 25 | – | ns |
| $t_{14}$ | MCE/ LOW to data clocked in | 150 | – | ns |
| $t_{15}$ | Address valid to data clocked in | 205 | – | ns |
| $t_{16}$ | MOE/ LOW to data clocked in | 100 | – | ns |
| $t_{17}$ | Data hold from address, MOE/, MCE/ change | 0 | – | ns |
| $t_{19}$ | Data setup to CLK HIGH | 5 | – | ns |

**Figure 6.25 External Memory Read**

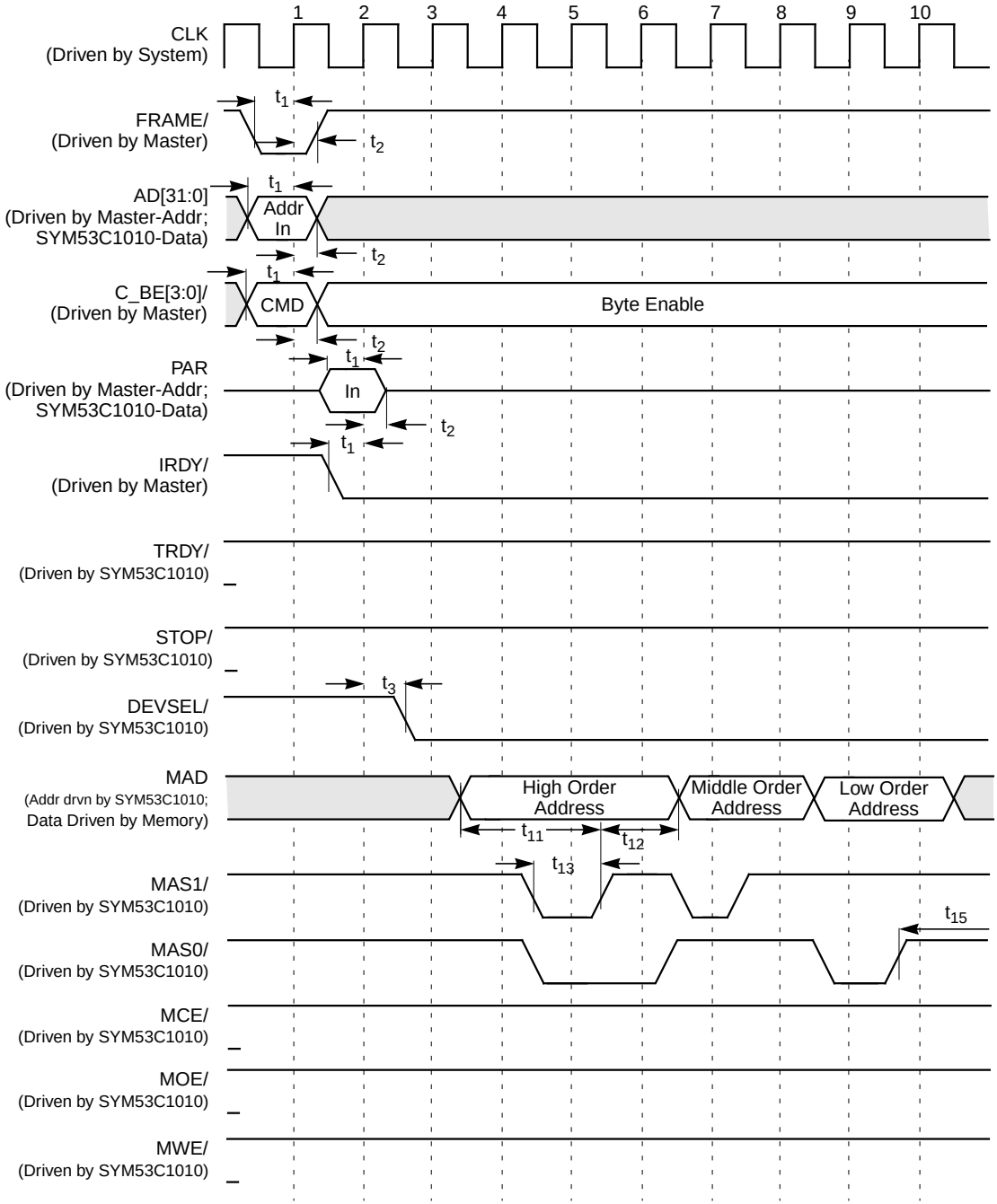


**Figure 6.25 External Memory Read (Cont.)**

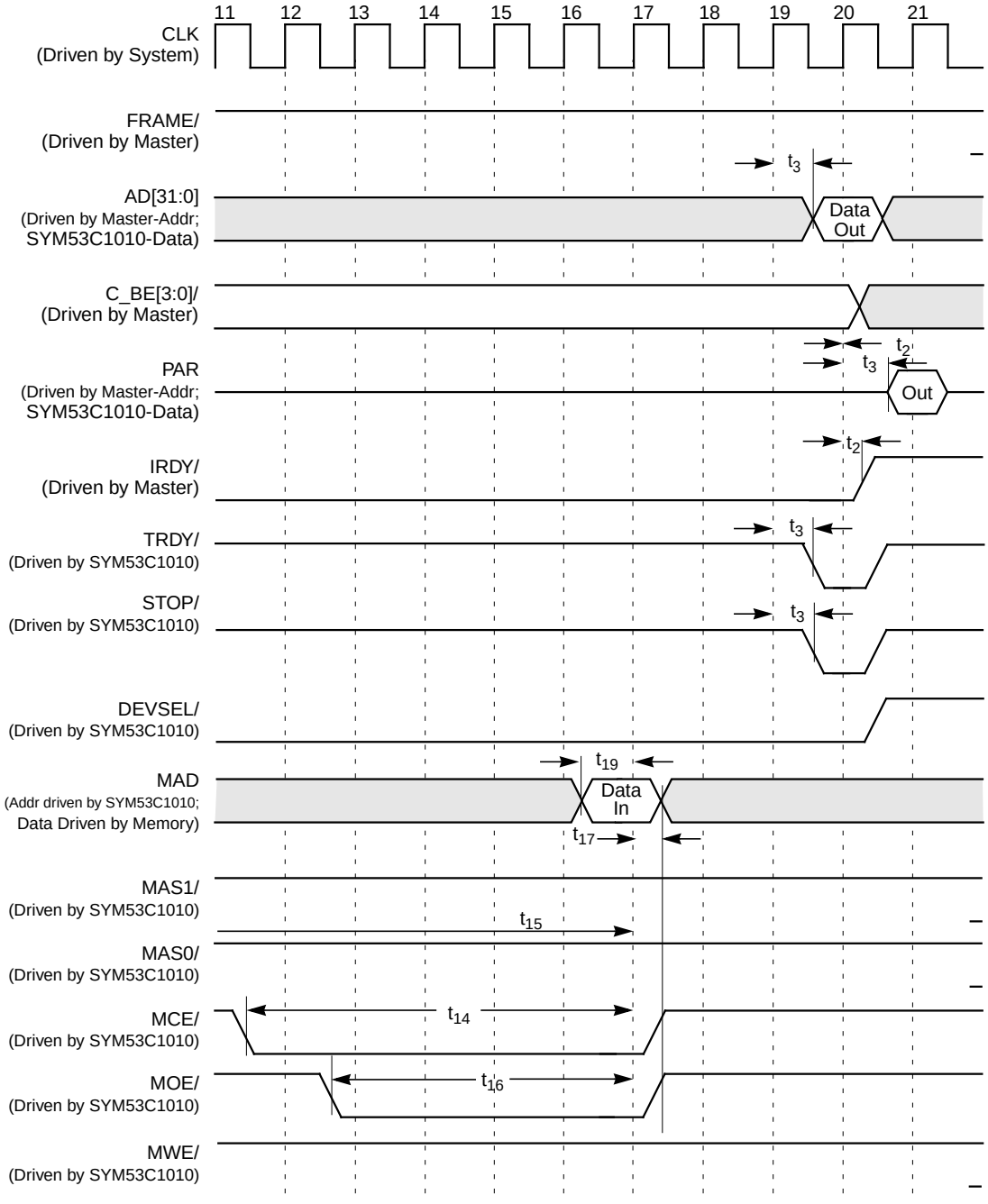


**Table 6.32 External Memory Write**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_1$ | Shared signal input setup time | 7 | – | ns |
| $t_2$ | Shared signal input hold time | 0 | – | ns |
| $t_3$ | CLK to shared signal output valid | 2 | 11 | ns |
| $t_{11}$ | Address setup to MAS/ HIGH | 25 | – | ns |
| $t_{12}$ | Address hold from MAS/ HIGH | 15 | – | ns |
| $t_{13}$ | MAS/ pulse width | 25 | – | ns |
| $t_{20}$ | Data setup to MWE/ LOW | 30 | – | ns |
| $t_{21}$ | Data hold from MWE/ HIGH | 20 | – | ns |
| $t_{22}$ | MWE/ pulse width | 100 | – | ns |
| $t_{23}$ | Address setup to MWE/ LOW | 60 | – | ns |
| $t_{24}$ | MCE/ LOW to MWE/ HIGH | 120 | – | ns |
| $t_{25}$ | MCE/ LOW to MWE/ LOW | 25 | – | ns |
| $t_{26}$ | MWE/ HIGH to MCE/ HIGH | 25 | – | ns |

**Figure 6.26 External Memory Write**

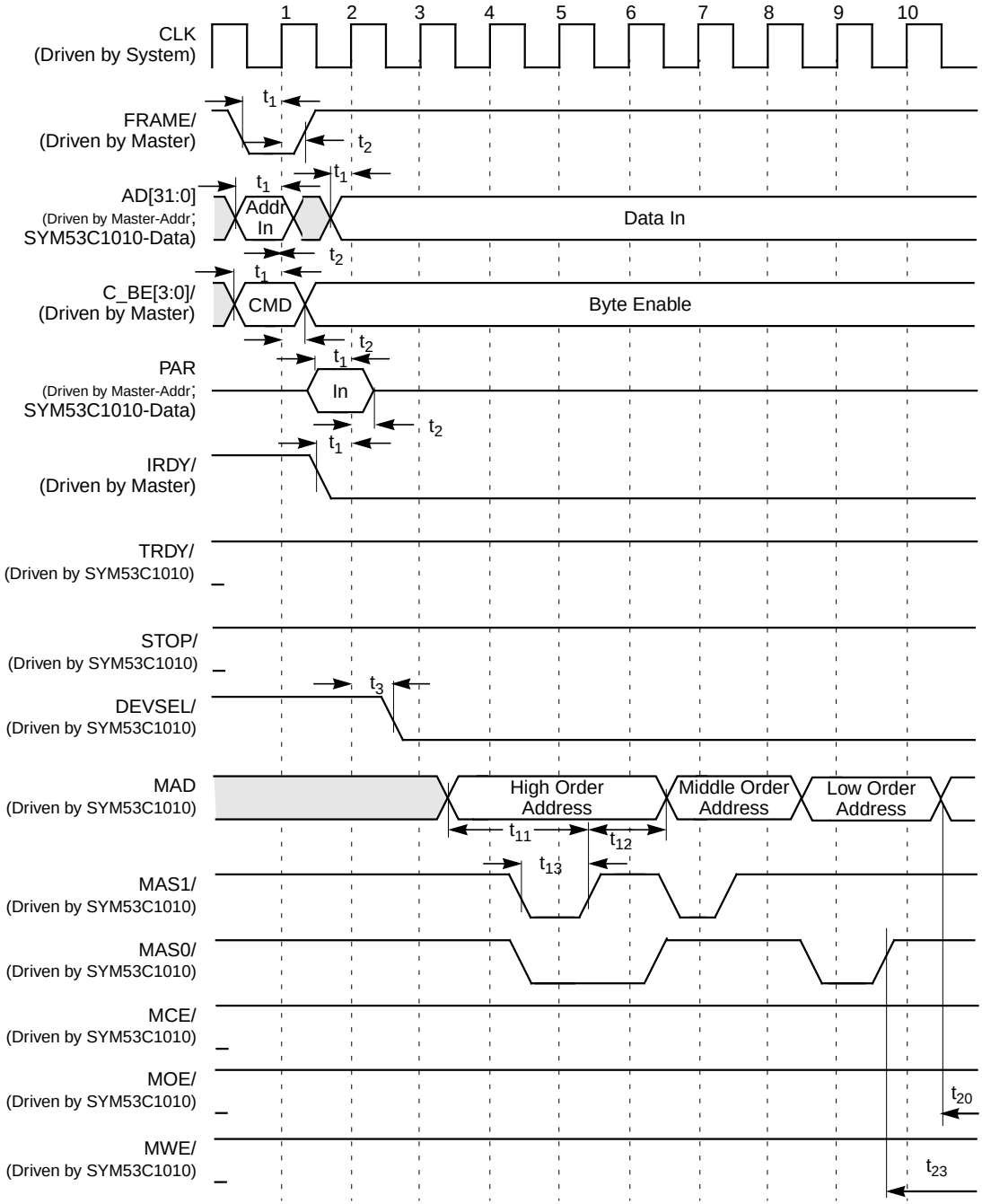


**Figure 6.26 External Memory Write (Cont.)**

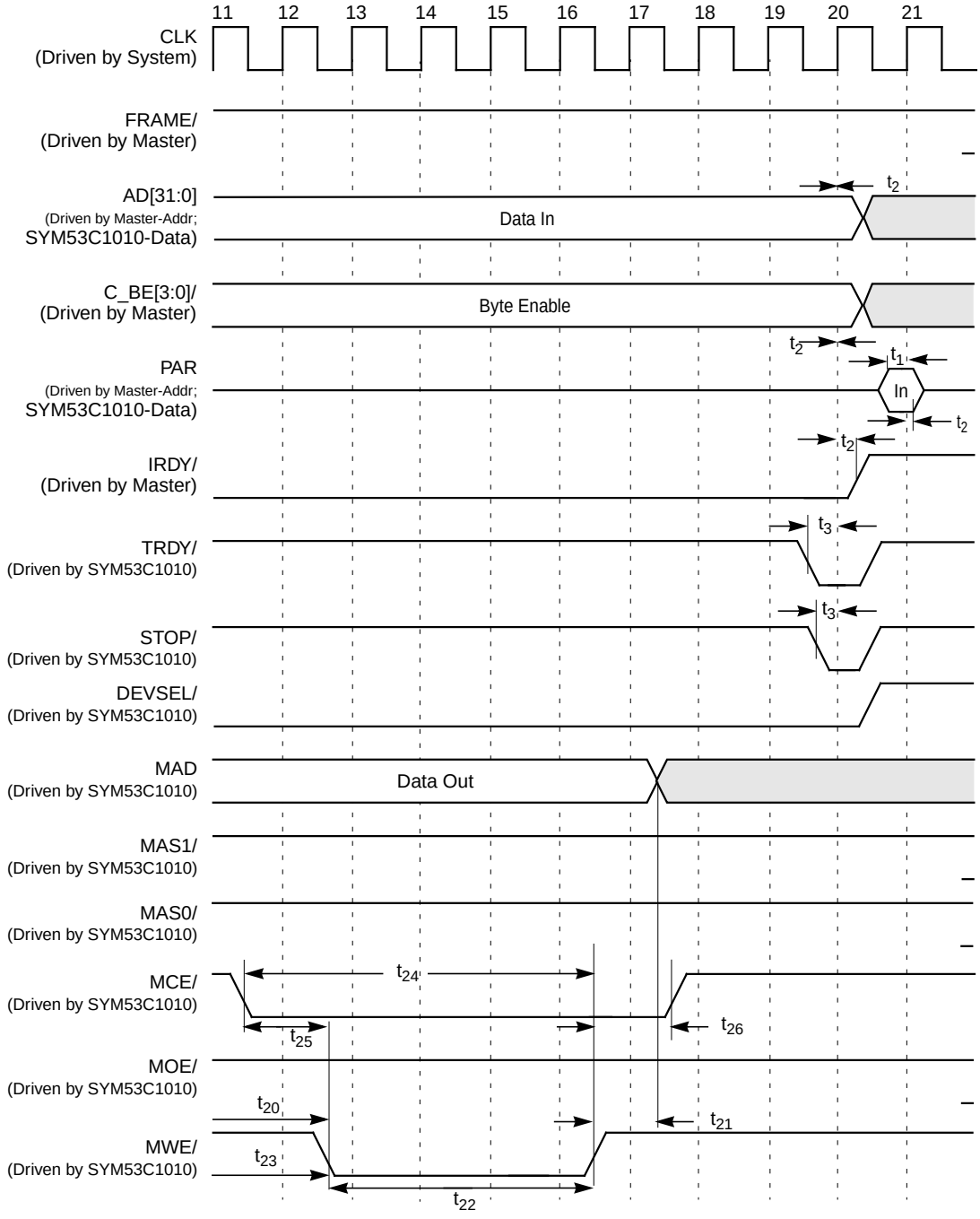


**Table 6.33 Normal/Fast Memory (≥ 128 Kbytes) Single Byte Access Read Cycle**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_{11}$ | Address setup to MAS/ high | 25 | – | ns |
| $t_{12}$ | Address hold from MAS/ high | 15 | – | ns |
| $t_{13}$ | MAS/ pulse width | 25 | – | ns |
| $t_{14}$ | MCE/ LOW to data clocked in | 150 | – | ns |
| $t_{15}$ | Address valid to data clocked in | 205 | – | ns |
| $t_{16}$ | MOE/ LOW to data clocked in | 100 | – | ns |
| $t_{17}$ | Data hold from address, MOE/, MCE/ change | 0 | – | ns |
| $t_{18}$ | Address out from MOE/, MCE/ HIGH | 50 | – | ns |
| $t_{19}$ | Data setup to CLK HIGH | 5 | – | ns |

**Figure 6.27 Normal/Fast Memory (≥ 128 Kbytes) Single Byte Access Read Cycle**

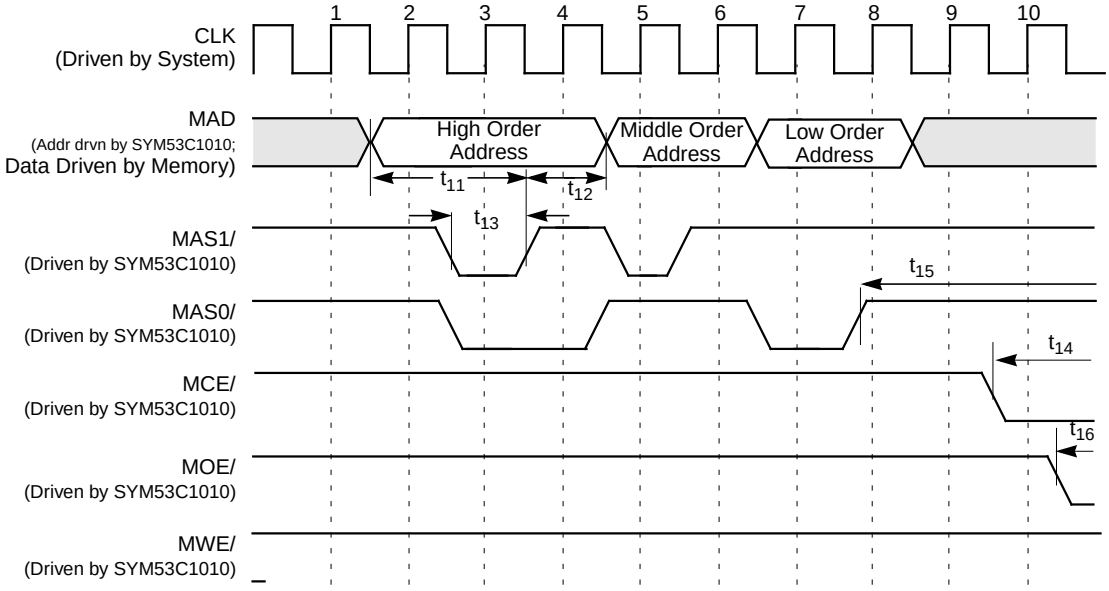


**Figure 6.27 Normal/Fast Memory (≥ 128 Kbytes) Single Byte Access Read Cycle (Cont.)**

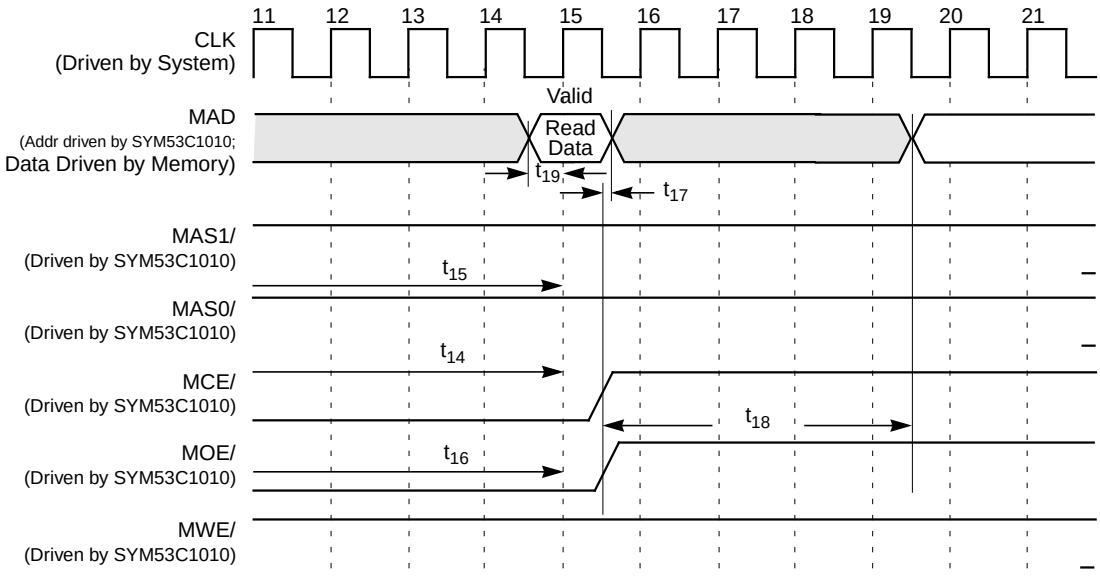


**Table 6.34 Normal/Fast Memory (≥ 128 Kbytes) Single Byte Access Write Cycle**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_{11}$ | Address setup to MAS/ HIGH | 25 | – | ns |
| $t_{12}$ | Address hold from MAS/ HIGH | 15 | – | ns |
| $t_{13}$ | MAS/ pulse width | 25 | – | ns |
| $t_{20}$ | Data setup to MWE/ LOW | 30 | – | ns |
| $t_{21}$ | Data hold from MWE/ HIGH | 20 | – | ns |
| $t_{22}$ | MWE/ pulse width | 100 | – | ns |
| $t_{23}$ | Address setup to MWE/ LOW | 60 | – | ns |
| $t_{24}$ | MCE/ LOW to MWE/ HIGH | 120 | – | ns |
| $t_{25}$ | MCE/ LOW to MWE/ LOW | 25 | – | ns |
| $t_{26}$ | MWE/ HIGH to MCE/ HIGH | 25 | – | ns |

**Figure 6.28 Normal/Fast Memory (≥ 128 Kbytes) Single Byte Access Write Cycle**

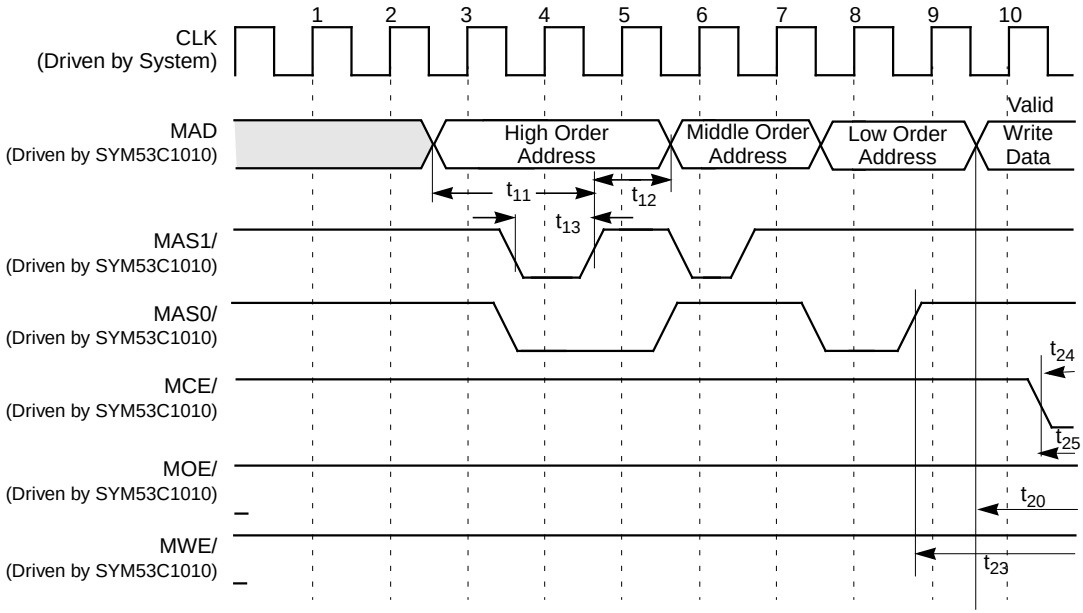


**Figure 6.28 Normal/Fast Memory (≥ 128 Kbytes) Single Byte Access Write Cycle (Cont.)**

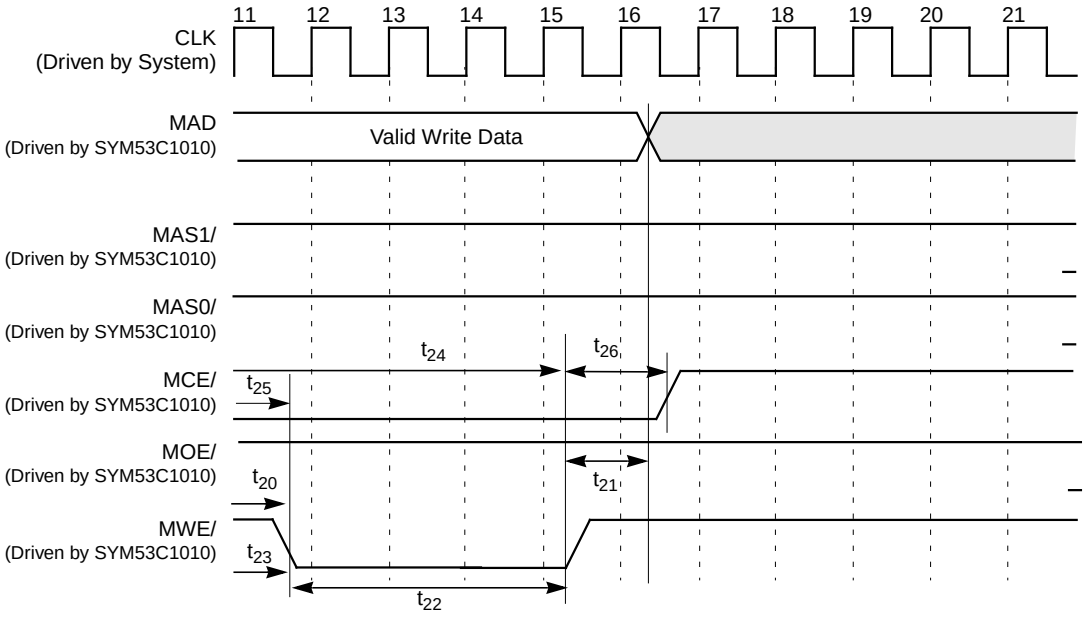


**Figure 6.29 Normal/Fast Memory (≥ 128 Kbytes) Multiple Byte Access Read Cycle**

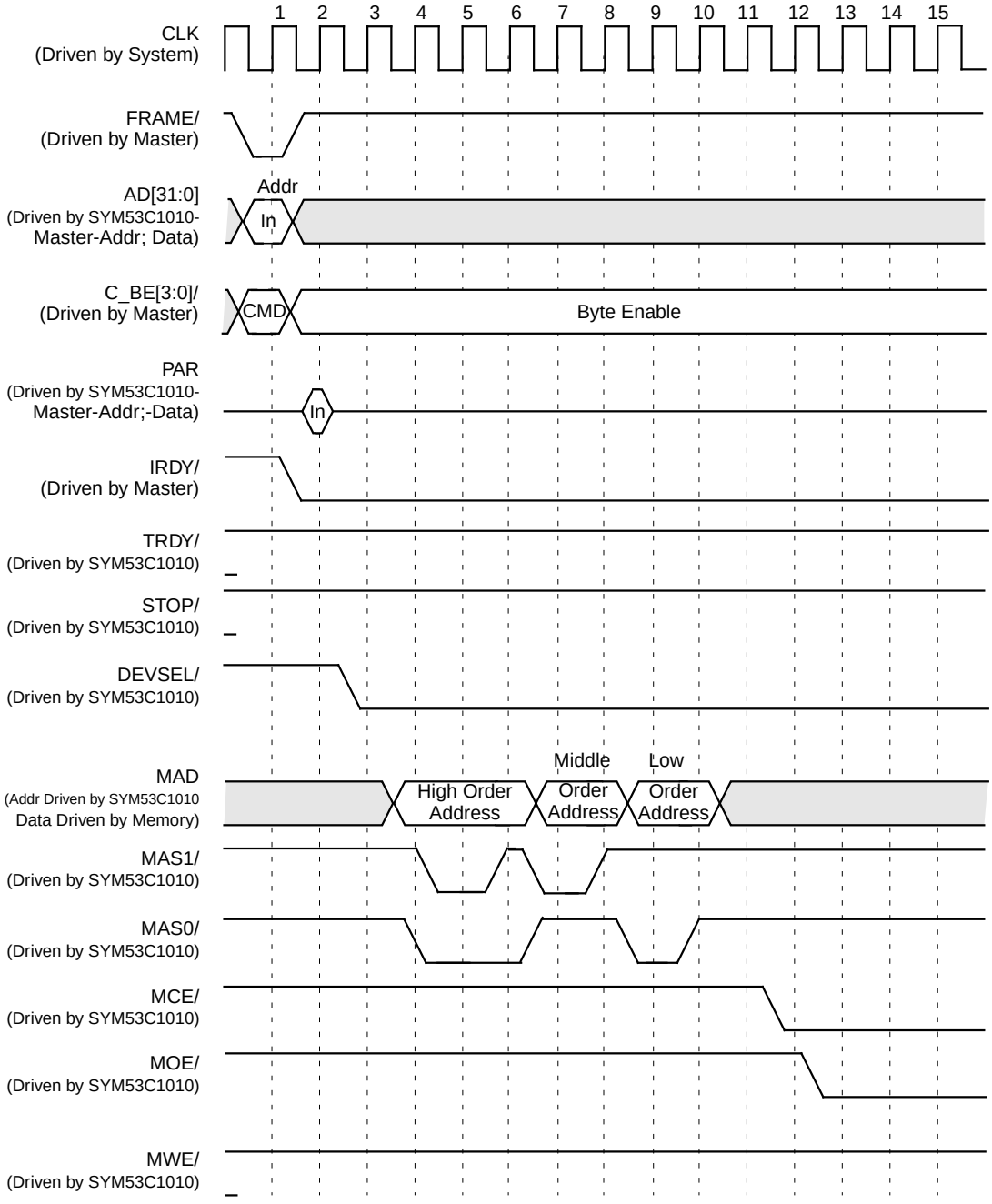


**Figure 6.29 Normal/Fast Memory (≥ 128 Kbytes) Multiple Byte Access Read Cycle (Cont.)**

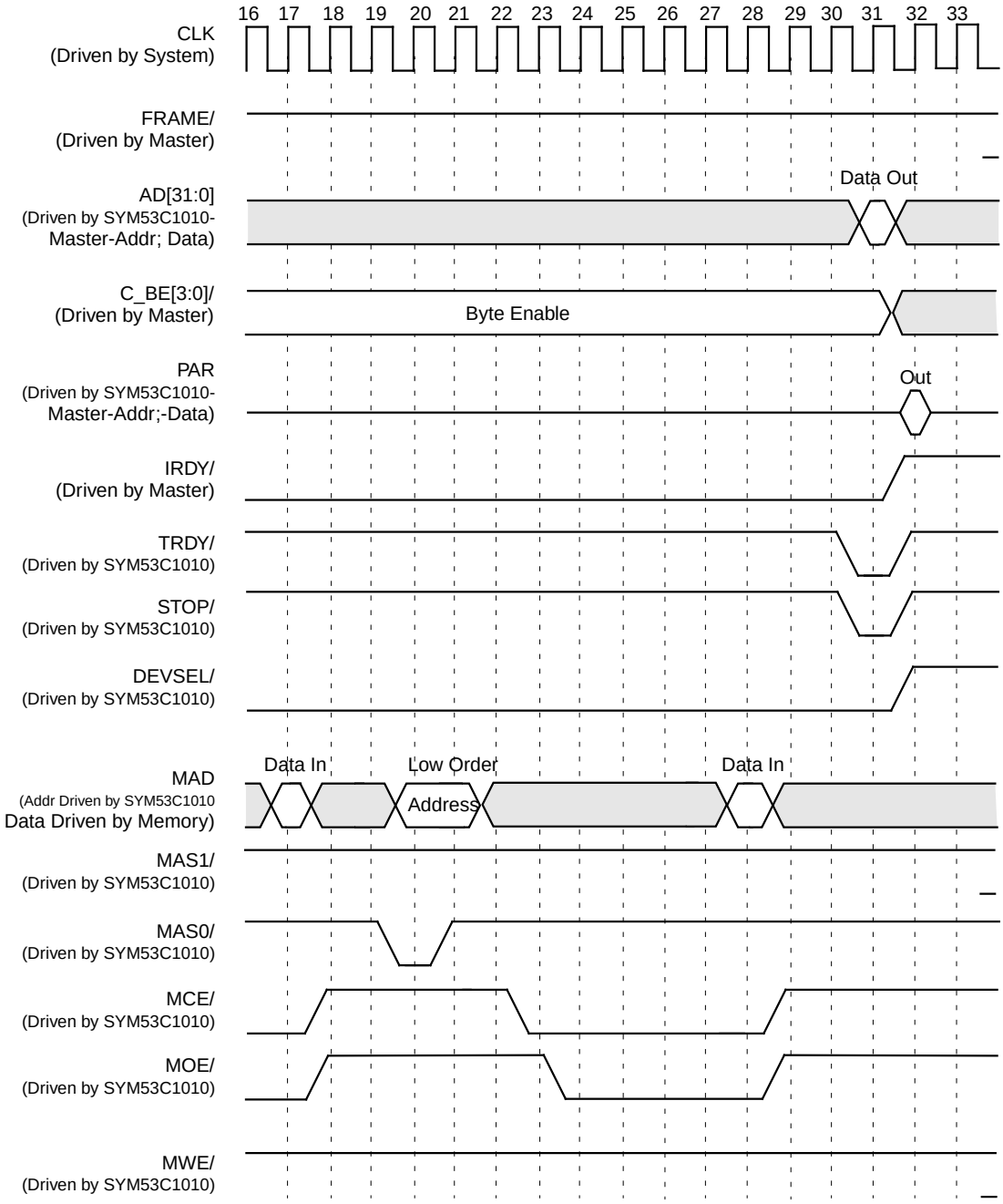


**Figure 6.30 Normal/Fast Memory (≥ 128 Kbytes) Multiple Byte Access Write Cycle**

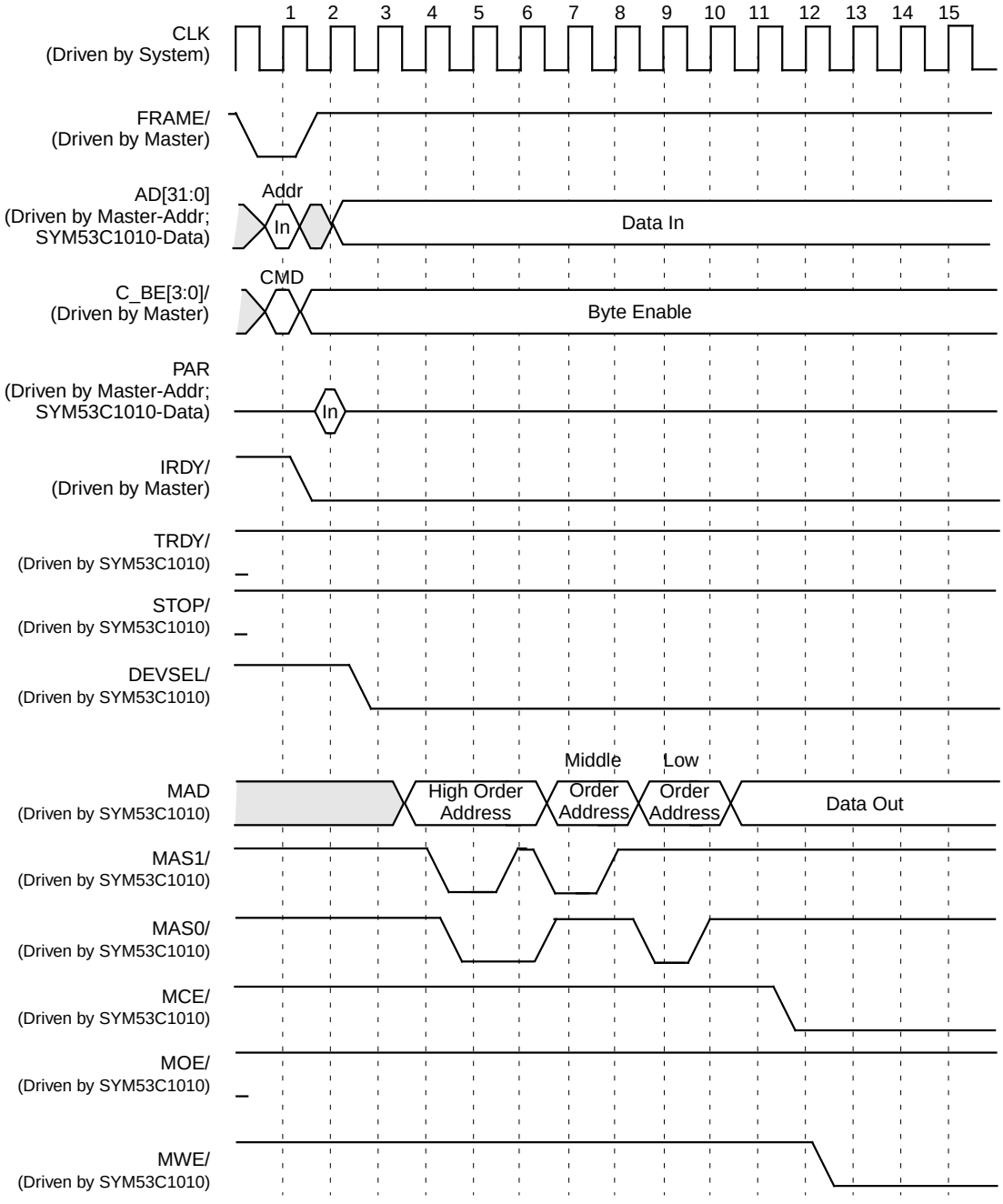


**Figure 6.30 Normal/Fast Memory (≥ 128 Kbytes) Multiple Byte Access Write Cycle (Cont.)**

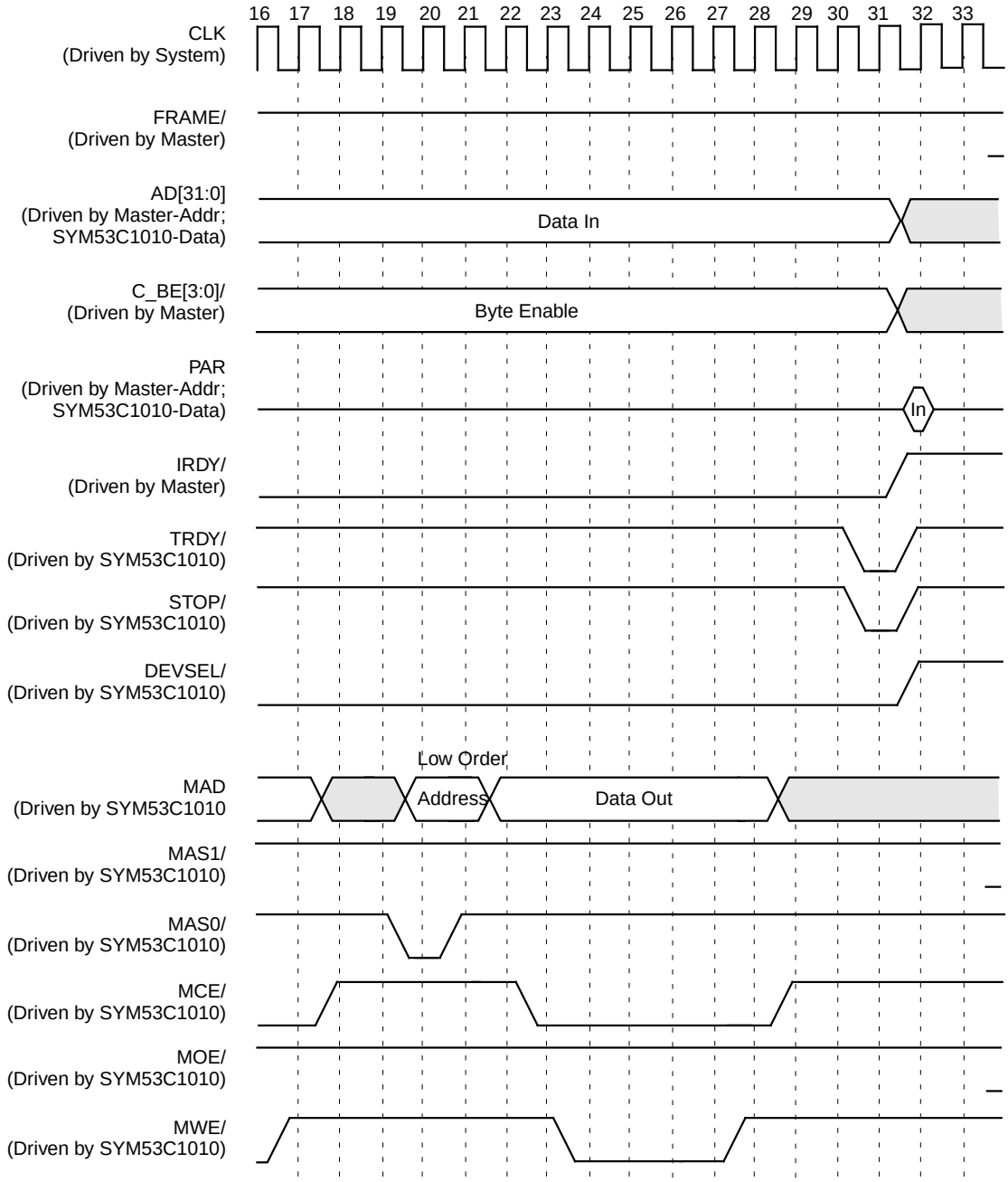


**Table 6.35 Slow Memory (≥ 128 Kbytes) Read Cycle**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_{11}$ | Address setup to MAS/ HIGH | 25 | – | ns |
| $t_{12}$ | Address hold from MAS/ HIGH | 15 | – | ns |
| $t_{13}$ | MAS/ pulse width | 25 | – | ns |
| $t_{14}$ | MCE/ LOW to data clocked in | 150 | – | ns |
| $t_{15}$ | Address valid to data clocked in | 205 | – | ns |
| $t_{16}$ | MOE/ LOW to data clocked in | 100 | – | ns |
| $t_{17}$ | Data hold from address, MOE/, MCE/ change | 0 | – | ns |
| $t_{18}$ | Address out from MOE/, MCE/ HIGH | 50 | – | ns |
| $t_{19}$ | Data setup to CLK HIGH | 5 | – | ns |

**Figure 6.31 Slow Memory (≥ 128 Kbytes) Read Cycle**

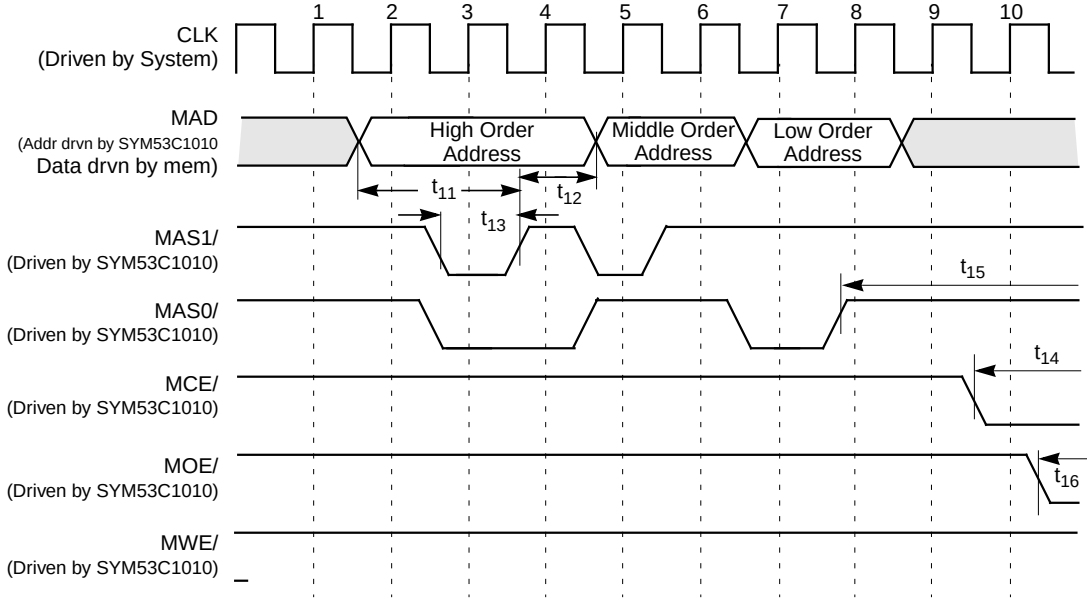


**Figure 6.31 Slow Memory (≥ 128 Kbytes) Read Cycle (Cont.)**

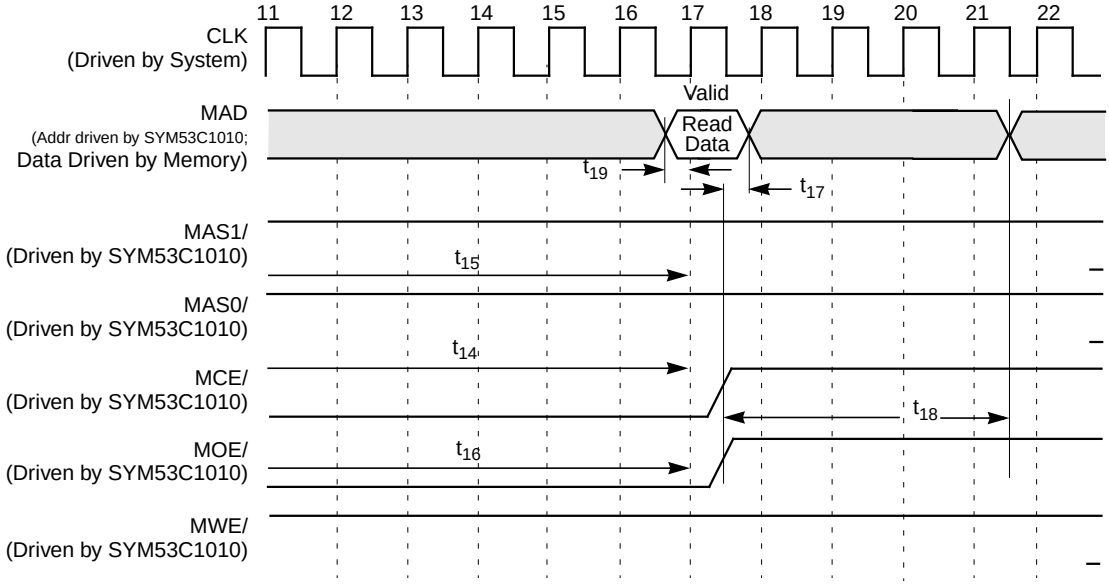


**Table 6.36 Slow Memory (≥ 128 Kbytes) Write Cycle**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_{11}$ | Address setup to MAS/ HIGH | 25 | – | ns |
| $t_{12}$ | Address hold from MAS/ HIGH | 15 | – | ns |
| $t_{13}$ | MAS/ pulse width | 25 | – | ns |
| $t_{20}$ | Data setup to MWE/ LOW | 30 | – | ns |
| $t_{21}$ | Data hold from MWE/ HIGH | 20 | – | ns |
| $t_{22}$ | MWE/ pulse width | 100 | – | ns |
| $t_{23}$ | Address setup to MWE/ LOW | 60 | – | ns |
| $t_{24}$ | MCE/ LOW to MWE/ HIGH | 120 | – | ns |
| $t_{25}$ | MCE/ LOW to MWE/ LOW | 25 | – | ns |
| $t_{26}$ | MWE/ HIGH to MCE/ HIGH | 25 | – | ns |

**Figure 6.32 Slow Memory (≥ 128 Kbytes) Write Cycle**

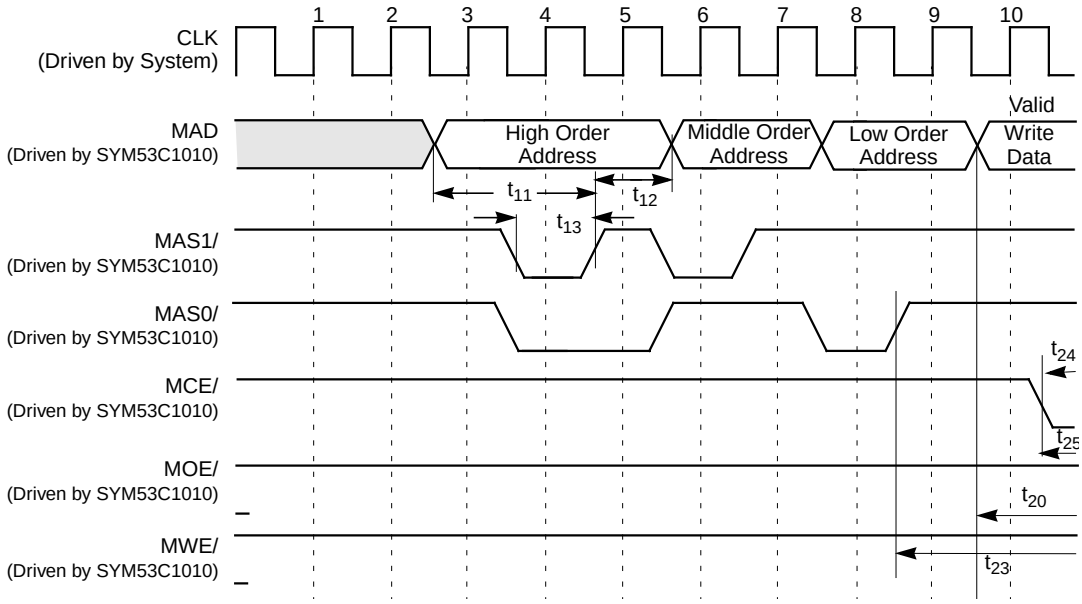


**Figure 6.32 Slow Memory (≥ 128 Kbytes) Write Cycle (Cont.)**

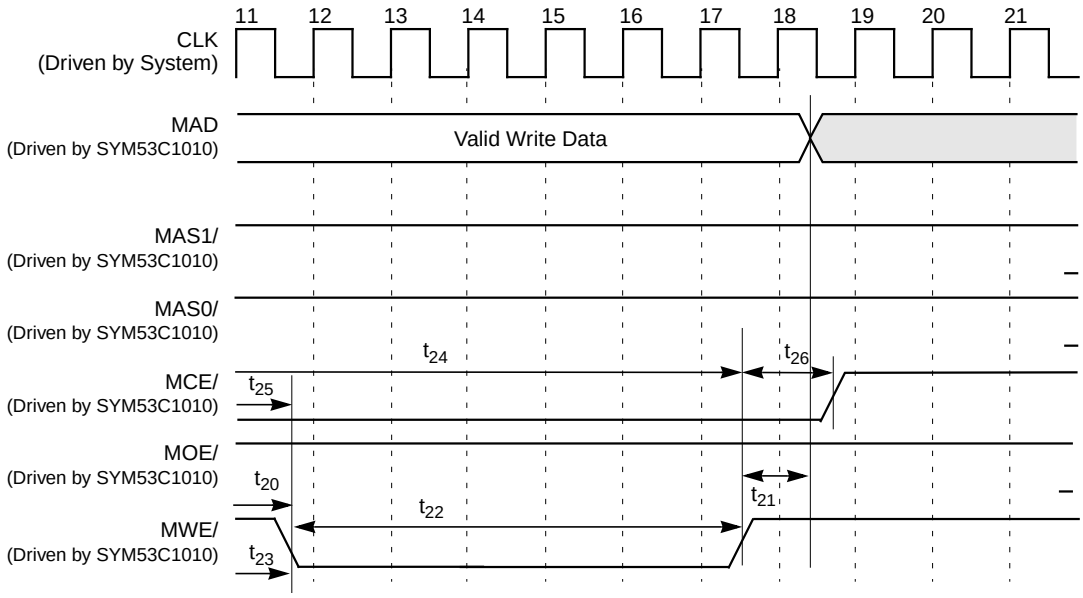


**Table 6.37 ≤ 64 Kbytes ROM Read Cycle**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_{11}$ | Address setup to MAS/ HIGH | 25 | – | ns |
| $t_{12}$ | Address hold from MAS/ HIGH | 15 | – | ns |
| $t_{13}$ | MAS/ pulse width | 25 | – | ns |
| $t_{14}$ | MCE/ LOW to data clocked in | 150 | – | ns |
| $t_{15}$ | Address valid to data clocked in | 205 | – | ns |
| $t_{16}$ | MOE/ LOW to data clocked in | 100 | – | ns |
| $t_{17}$ | Data hold from address, MOE/, MCE/ change | 0 | – | ns |
| $t_{18}$ | Address out from MOE/, MCE/ HIGH | 50 | – | ns |
| $t_{19}$ | Data setup to CLK HIGH | 5 | – | ns |

**Figure 6.33 ≤ 64 Kbytes ROM Read Cycle**

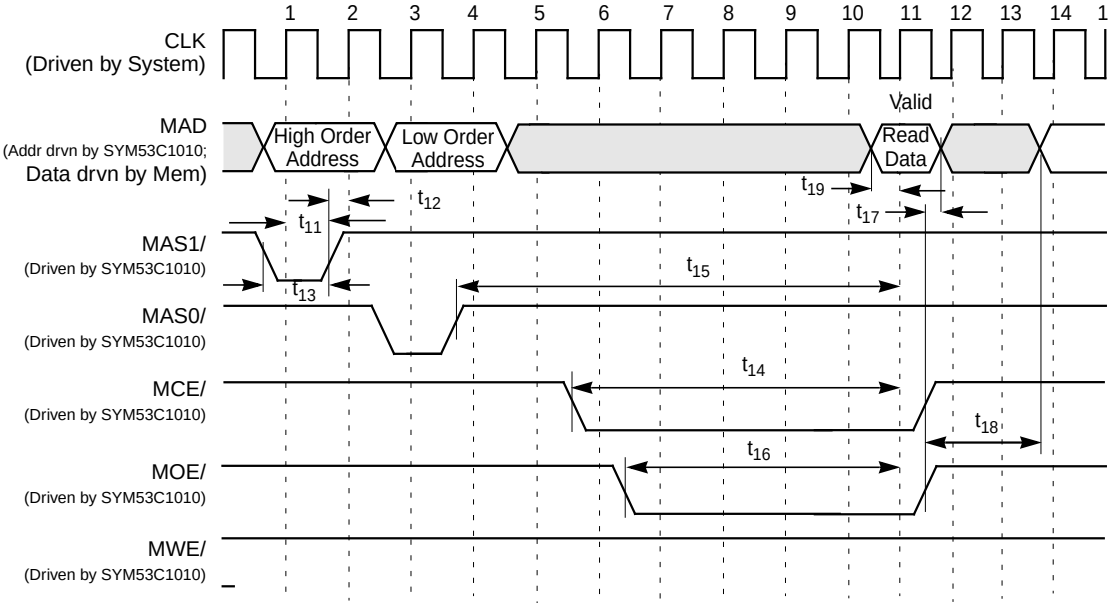
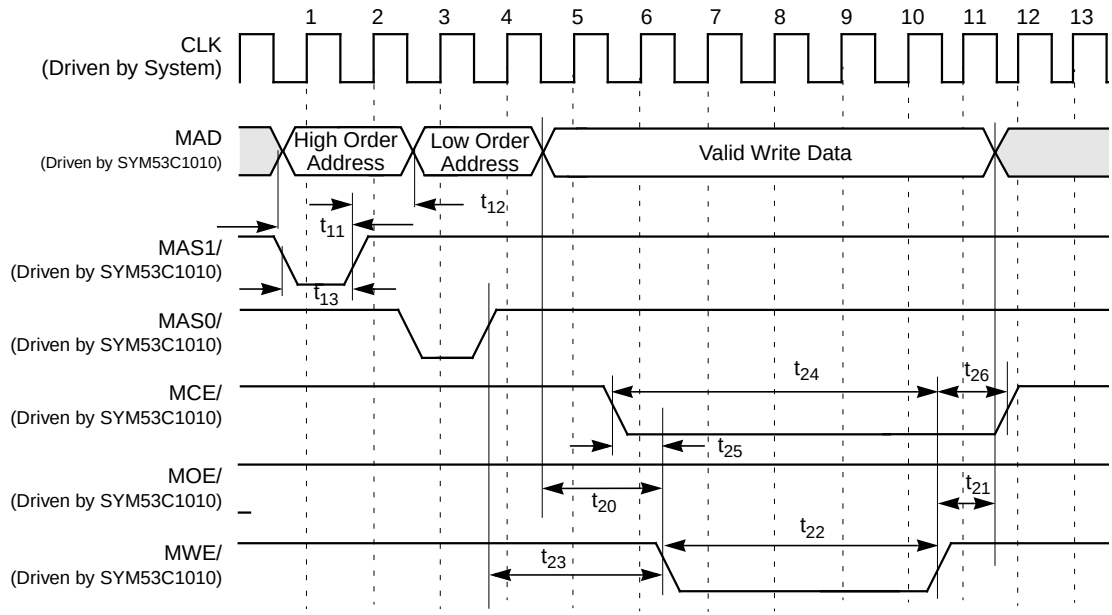


**Table 6.38 ≤ 64 Kbytes ROM Write Cycle**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_{11}$ | Address setup to MAS/ HIGH | 25 | – | ns |
| $t_{12}$ | Address hold from MAS/ HIGH | 15 | – | ns |
| $t_{13}$ | MAS/ pulse width | 25 | – | ns |
| $t_{20}$ | Data setup to MWE/ LOW | 30 | – | ns |
| $t_{21}$ | Data hold from MWE/ HIGH | 20 | – | ns |
| $t_{22}$ | MWE/ pulse width | 100 | – | ns |
| $t_{23}$ | Address setup to MWE/ LOW | 60 | – | ns |
| $t_{24}$ | MCE/ LOW to MWE/ HIGH | 120 | – | ns |
| $t_{25}$ | MCE/ LOW to MWE/ LOW | 25 | – | ns |
| $t_{26}$ | MWE/ HIGH to MCE/ HIGH | 25 | – | ns |

**Figure 6.34 ≤ 64 Kbytes ROM Write Cycle**



## 6.5 SCSI Timing Diagrams

Figure 6.35 through Figure 6.40 represent signal activity on the SCSI bus. This section includes timing diagrams for asynchronous and synchronous access to the SCSI bus.

### 6.5.1 Asynchronous Timing

**Table 6.39 Initiator Asynchronous Send**

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $t_1$ | SACK/ asserted from SREQ/ asserted | 5 | – | ns |
| $t_2$ | SACK/ deasserted from SREQ/ deasserted | 5 | – | ns |
| $t_3$ | Data setup to SACK/ asserted | 55 | – | ns |
| $t_4$ | Data hold from SREQ/ deasserted | 0 | – | ns |

**Figure 6.35 Initiator Asynchronous Send**

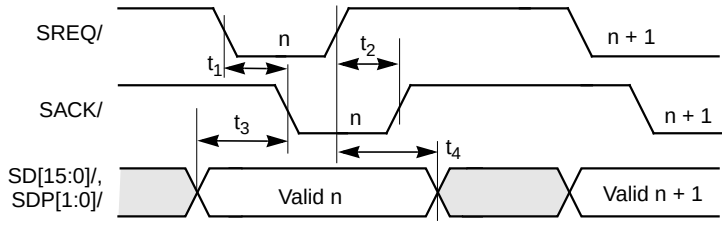


**Table 6.40 Initiator Asynchronous Receive**

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $t_1$ | SACK/ asserted from SREQ/ asserted | 5 | – | ns |
| $t_2$ | SACK/ deasserted from SREQ/ deasserted | 5 | – | ns |
| $t_3$ | Data setup to SREQ/ asserted | 0 | – | ns |
| $t_4$ | Data hold from SACK/ asserted | 0 | – | ns |

**Figure 6.36 Initiator Asynchronous Receive**

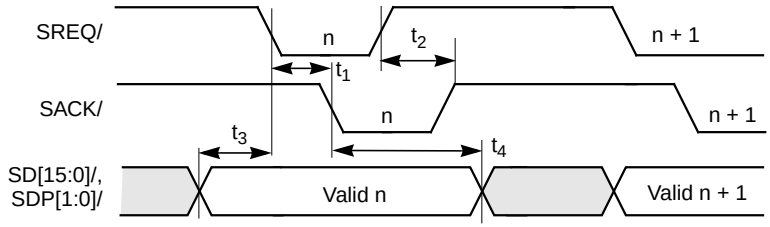



**Table 6.41 Target Asynchronous Send**

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $t_1$ | SREQ/ deasserted from SACK/ asserted | 5 | – | ns |
| $t_2$ | SREQ/ asserted from SACK/ deasserted | 5 | – | ns |
| $t_3$ | Data setup to SREQ/ asserted | 55 | – | ns |
| $t_4$ | Data hold from SACK/ asserted | 0 | – | ns |

**Figure 6.37 Target Asynchronous Send**

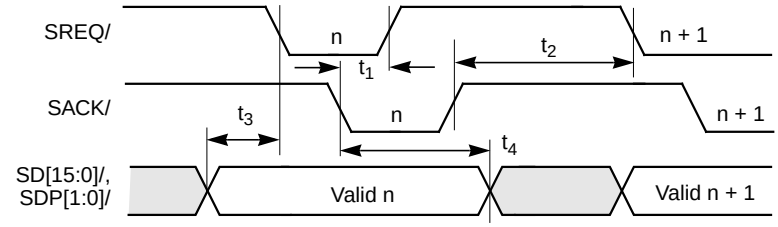
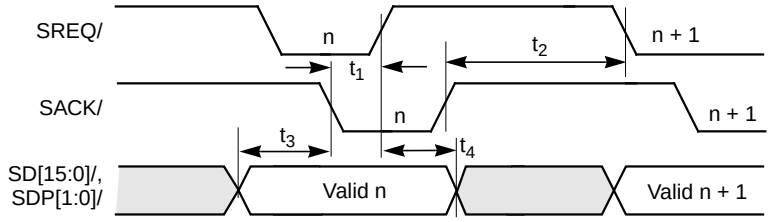


**Table 6.42 Target Asynchronous Receive**

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $t_1$ | SREQ/ deasserted from SACK/ asserted | 5 | – | ns |
| $t_2$ | SREQ/ asserted from SACK/ deasserted | 5 | – | ns |
| $t_3$ | Data setup to SACK/ asserted | 0 | – | ns |
| $t_4$ | Data hold from SREQ/ deasserted | 0 | – | ns |

**Figure 6.38 Target Asynchronous Receive**



## 6.5.2 Synchronous Timing

### 6.5.2.1 Single Transition Data Clocking

**Table 6.43 SCSI-1 Transfers (SE 5.0 Mbytes)**

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $t_{ST1}$ | Send SREQ/ or SACK/ assertion pulse width | 80 | – | ns |
| $t_{ST2}$ | Send SREQ/ or SACK/ deassertion pulse width | 80 | – | ns |
| $t_{ST1}$ | Receive SREQ/ or SACK/ assertion pulse width | 70 | – | ns |
| $t_{ST2}$ | Receive SREQ/ or SACK/ deassertion pulse width | 70 | – | ns |
| $t_{ST3}$ | Send data setup to SREQ/ or SACK/ asserted | 24 | – | ns |
| $t_{ST4}$ | Send data hold from SREQ/ or SACK/ asserted | 54 | – | ns |
| $t_{ST5}$ | Receive data setup to SREQ/ or SACK/ asserted | 14 | – | ns |
| $t_{ST6}$ | Receive data hold from SREQ/ or SACK/ asserted | 24 | – | ns |

**Table 6.44 SCSI-2 Fast Transfers 10.0 Mbytes (8-Bit Transfers) or 20.0 Mbytes (16-Bit Transfers) 40 MHz Clock**

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $t_{ST1}$ | Send SREQ/ or SACK/ assertion pulse width | 30 | – | ns |
| $t_{ST2}$ | Send SREQ/ or SACK/ deassertion pulse width | 30 | – | ns |
| $t_{ST1}$ | Receive SREQ/ or SACK/ assertion pulse width | 22 | – | ns |
| $t_{ST2}$ | Receive SREQ/ or SACK/ deassertion pulse width | 22 | – | ns |
| $t_{ST3}$ | Send data setup to SREQ/ or SACK/ asserted | 24 | – | ns |
| $t_{ST4}$ | Send data hold from SREQ/ or SACK/ asserted | 34 | – | ns |
| $t_{ST5}$ | Receive data setup to SREQ/ or SACK/ asserted | 14 | – | ns |
| $t_{ST6}$ | Receive data hold from SREQ/ or SACK/ asserted | 24 | – | ns |

**Table 6.45 Ultra SCSI SE Transfers 20.0 Mbytes (8-Bit Transfers) or 40.0 Mbytes (16-Bit Transfers) Quadrupled 40 MHz Clock[1]**

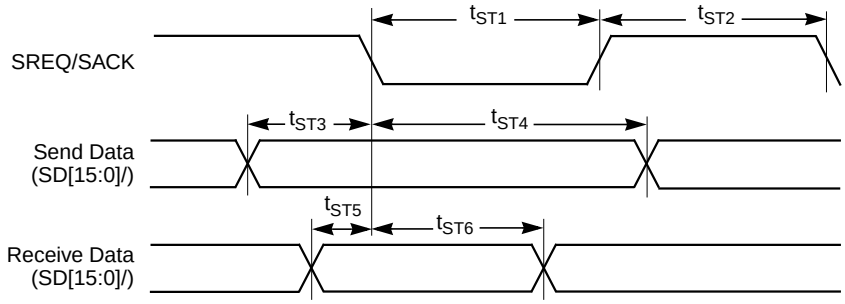
| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_{ST1}$ | Send SREQ/ or SACK/ assertion pulse width | 15 | – | ns |
| $t_{ST2}$ | Send SREQ/ or SACK/ deassertion pulse width | 15 | – | ns |
| $t_{ST1}$ | Receive SREQ/ or SACK/ assertion pulse width | 11 | – | ns |
| $t_{ST2}$ | Receive SREQ/ or SACK/ deassertion pulse width | 11 | – | ns |
| $t_{ST3}$ | Send data setup to SREQ/ or SACK/ asserted | 12 | – | ns |
| $t_{ST4}$ | Send data hold from SREQ/ or SACK/ asserted | 17 | – | ns |
| $t_{ST5}$ | Receive data setup to SREQ/ or SACK/ asserted | 6 | – | ns |
| $t_{ST6}$ | Receive data hold from SREQ/ or SACK/ asserted | 11 | – | ns |

1. Note: for fast SCSI, set the TolerANT Enable bit (bit 7 in SCSI Test Three (STEST3)).

**Table 6.46 Ultra2 SCSI Transfers 40.0 Mbyte (8-Bit Transfers) or 80.0 Mbyte (16-Bit Transfers) Quadrupled 40 MHz Clock**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_{ST1}$ | Send SREQ/ or SACK/ assertion pulse width | 8 | – | ns |
| $t_{ST2}$ | Send SREQ/ or SACK/ deassertion pulse width | 8 | – | ns |
| $t_{ST1}$ | Receive SREQ/ or SACK/ assertion pulse width | 6.5 | – | ns |
| $t_{ST2}$ | Receive SREQ/ or SACK/ deassertion pulse width | 6.5 | – | ns |
| $t_{ST3}$ | Send data setup to SREQ/ or SACK/ asserted | 9.5 | – | ns |
| $t_{ST4}$ | Send data hold from SREQ/ or SACK/ asserted | 9.5 | – | ns |
| $t_{ST5}$ | Receive data setup to SREQ/ or SACK/ asserted | 4.5 | – | ns |
| $t_{ST6}$ | Receive data hold from SREQ/ or SACK/ asserted | 4.5 | – | ns |

**Figure 6.39 Initiator and Target ST Synchronous Transfer**




### 6.5.2.2 Double Transition Data Clocking

**Table 6.47 SCSI-2 Fast Transfers 10.0 Mbytes (8-Bit Transfers) or 20.0 Mbytes (16-Bit Transfers) 40 MHz Clock**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_{DT1}$ | Send SREQ/ or SACK/ assertion pulse width | 92 | – | ns |
| $t_{DT2}$ | Send SREQ/ or SACK/ deassertion pulse width | 92 | – | ns |
| $t_{DT1}$ | Receive SREQ/ or SACK/ assertion pulse width | 80 | – | ns |
| $t_{DT2}$ | Receive SREQ/ or SACK/ deassertion pulse width | 80 | – | ns |
| $t_{DT3}$ | Send data setup to SREQ/ or SACK/ transition | 40 | – | ns |
| $t_{DT4}$ | Send data hold from SREQ/ or SACK/ transition | 40 | – | ns |
| $t_{DT5}$ | Receive data setup to SREQ/ or SACK/ transition | 10 | – | ns |
| $t_{DT6}$ | Receive data hold from SREQ/ or SACK/ transition | 10 | – | ns |
| $t_{DT7}$ | Send CRC Request Setup to SREQ/ transition | 50 | – | ns |
| $t_{DT8}$ | Send CRC Request Hold to SREQ/ transition | 40 | – | ns |
| $t_{DT9}$ | Receive CRC Request Setup to SREQ/ transition | 17 | – | ns |
| $t_{DT10}$ | Receive CRC Request Hold to SREQ/ transition | 10 | – | ns |

**Table 6.48 Ultra SCSI SE Transfers 20.0 Mbytes (8-Bit Transfers) or 40.0 Mbytes (16-Bit Transfers) Quadrupled 40 MHz Clock**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_{DT1}$ | Send SREQ/ or SACK/ assertion pulse width | 46 | – | ns |
| $t_{DT2}$ | Send SREQ/ or SACK/ deassertion pulse width | 46 | – | ns |
| $t_{DT1}$ | Receive SREQ/ or SACK/ assertion pulse width | 40 | – | ns |
| $t_{DT2}$ | Receive SREQ/ or SACK/ deassertion pulse width | 40 | – | ns |
| $t_{DT3}$ | Send data setup to SREQ/ or SACK/ transition | 20 | – | ns |
| $t_{DT4}$ | Send data hold from SREQ/ or SACK/ transition | 20 | – | ns |
| $t_{DT5}$ | Receive data setup to SREQ/ or SACK/ transition | 5 | – | ns |
| $t_{DT6}$ | Receive data hold from SREQ/ or SACK/ transition | 5 | – | ns |
| $t_{DT7}$ | Send CRC Request Setup to SREQ/ transition | 30 | – | ns |
| $t_{DT8}$ | Send CRC Request Hold to SREQ/ transition | 20 | – | ns |
| $t_{DT9}$ | Receive CRC Request Setup to SREQ/ transition | 12 | – | ns |
| $t_{DT10}$ | Receive CRC Request Hold to SREQ/ transition | 5 | – | ns |

**Table 6.49 Ultra2 SCSI Transfers 40.0 Mbyte (8-Bit Transfers) or 80.0 Mbyte (16-Bit Transfers) Quadrupled 40 MHz Clock**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_{DT1}$ | Send SREQ/ or SACK/ assertion pulse width | 23 | – | ns |
| $t_{DT2}$ | Send SREQ/ or SACK/ deassertion pulse width | 23 | – | ns |
| $t_{DT1}$ | Receive SREQ/ or SACK/ assertion pulse width | 20 | – | ns |
| $t_{DT2}$ | Receive SREQ/ or SACK/ deassertion pulse width | 20 | – | ns |
| $t_{DT3}$ | Send data setup to SREQ/ or SACK/ transition | 10 | – | ns |
| $t_{DT4}$ | Send data hold from SREQ/ or SACK/ transition | 10 | – | ns |
| $t_{DT5}$ | Receive data setup to SREQ/ or SACK/ transition | 2.5 | – | ns |
| $t_{DT6}$ | Receive data hold from SREQ/ or SACK/ transition | 2.5 | – | ns |
| $t_{DT7}$ | Send CRC Request Setup to SREQ/ transition | 20 | – | ns |
| $t_{DT8}$ | Send CRC Request Hold to SREQ/ transition | 10 | – | ns |
| $t_{DT9}$ | Receive CRC Request Setup to SREQ/ transition | 9.5 | – | ns |
| $t_{DT10}$ | Receive CRC Request Hold to SREQ/ transition | 2.5 | – | ns |

**Table 6.50 Ultra3 SCSI Transfers 160.0 Mbyte (16-Bit Transfers) Quadrupled 40 MHz Clock**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_{DT1}$ | Send SREQ/ or SACK/ assertion pulse width | 11.5 | – | ns |
| $t_{DT2}$ | Send SREQ/ or SACK/ deassertion pulse width | 11.5 | – | ns |
| $t_{DT1}$ | Receive SREQ/ or SACK/ assertion pulse width | 10 | – | ns |
| $t_{DT2}$ | Receive SREQ/ or SACK/ deassertion pulse width | 10 | – | ns |
| $t_{DT3}$ | Send data setup to SREQ/ or SACK/ transition | 5 | – | ns |
| $t_{DT4}$ | Send data hold from SREQ/ or SACK/ transition | 5 | – | ns |
| $t_{DT5}$ | Receive data setup to SREQ/ or SACK/ transition | 1.25 | – | ns |
| $t_{DT6}$ | Receive data hold from SREQ/ or SACK/ transition | 1.25 | – | ns |
| $t_{DT7}$ | Send CRC Request Setup to SREQ/ transition | 15 | – | ns |
| $t_{DT8}$ | Send CRC Request Hold to SREQ/ transition | 5 | – | ns |
| $t_{DT9}$ | Receive CRC Request Setup to SREQ/ transition | 8.25 | – | ns |
| $t_{DT10}$ | Receive CRC Request Hold to SREQ/ transition | 1.25 | – | ns |

**Figure 6.40 Initiator and Target DT Synchronous Transfer**

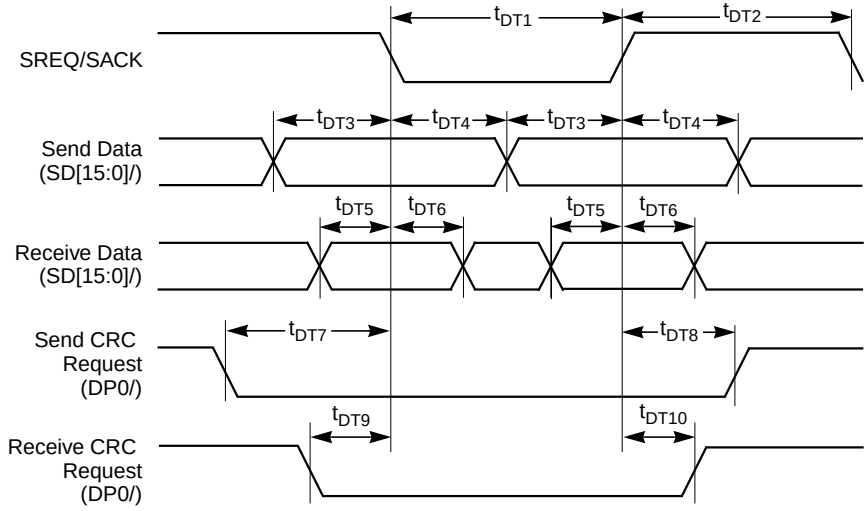
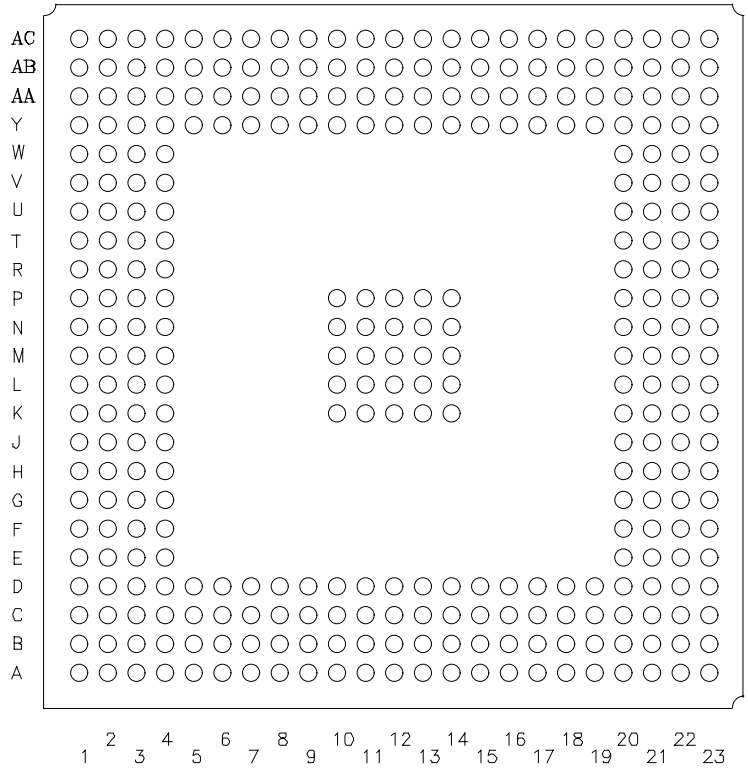
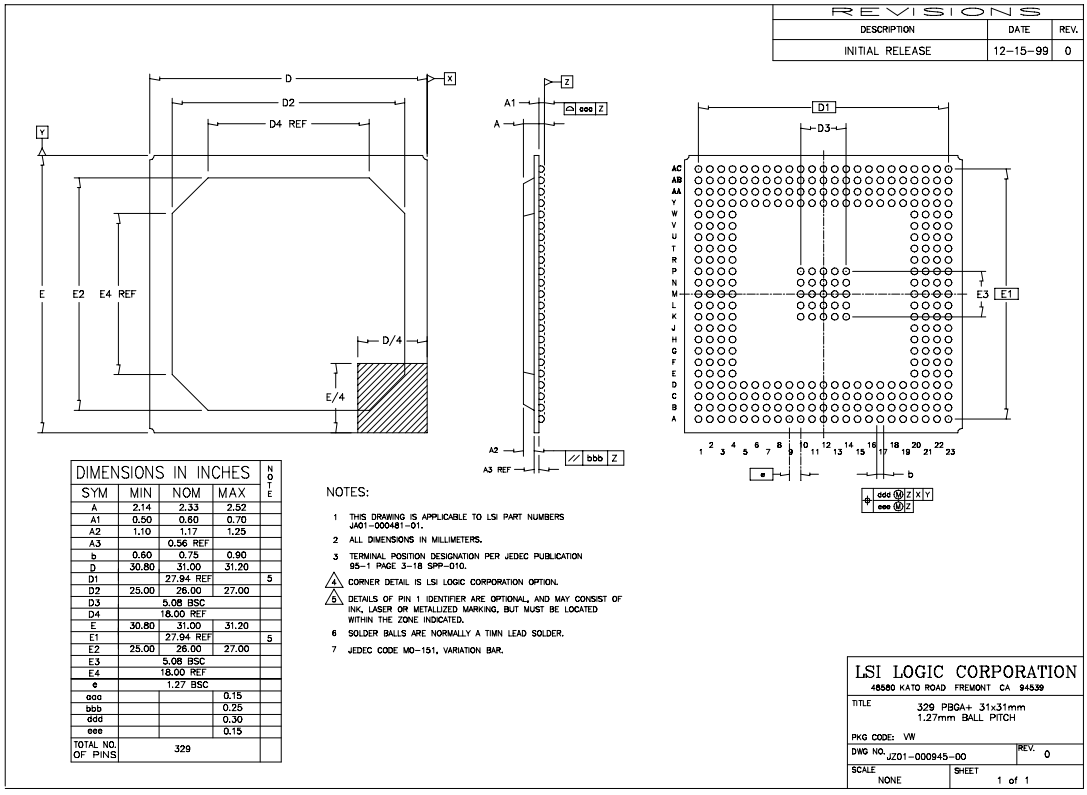


**Figure 6.41 SYM53C1010-33 329 Ball Grid Array (Bottom View)**



## Figure 6.42 SYM53C1010-33 329 BGA Mechanical Drawing




**Important:** **This drawing may not be the latest version. For board layout and manufacturing, obtain the most recent engineering drawings from your LSI Logic marketing representative by requesting the outline drawing for package code VW.**

# Appendix A
# Register Summary

**Table A.1 SYM53C1010-33 PCI Register Map**

| Register Name | Address | Read/Write | Page |
|---|---|---|---|
| **PCI Registers** | | | |
| Vendor ID | 0x00–0x01 | Read Only | 4-3 |
| Device ID | 0x02–0x03 | Read Only | 4-3 |
| Command | 0x04–0x05 | Read/Write | 4-3 |
| Status | 0x06–0x07 | Read/Write | 4-5 |
| Revision ID (RID) | 0x08 | Read Only | 4-7 |
| Class Code (CC) | 0x09–0x0B | Read Only | 4-7 |
| Cache Line Size (CLS) | 0x0C | Read/Write | 4-7 |
| Latency Timer (LT) | 0x0D | Read/Write | 4-8 |
| Header Type (HT) | 0x0E | Read Only | 4-9 |
| Reserved | 0x0F | – | 4-9 |
| Base Address Register Zero (BAR0) (I/O) | 0x10–0x13 | Read/Write | 4-9 |
| Base Address Register One (BAR1) (MEMORY) | 0x14–0x17 | Read/Write | 4-10 |
| Base Address Register Two (BAR2) (MEMORY) | 0x18–0x1B | Read/Write | 4-10 |
| Base Address Register Three (BAR3) (SCRIPTS RAM) | 0x1C–0x1F | Read/Write | 4-11 |
| Base Address Register Four (BAR4) (SCRIPTS RAM) | 0x20–0x23 | Read/Write | 4-11 |
| Reserved | 0x24–0x27 | – | 4-12 |
| Reserved | 0x28–0x2B | – | 4-12 |
| Subsystem Vendor ID (SVID) | 0x2C–0x2D | Read Only | 4-12 |

**Table A.1 SYM53C1010-33 PCI Register Map (Cont.)**

| Register Name | Address | Read/Write | Page |
|---|---|---|---|
| Subsystem ID (SID) | 0x2E–0x2F | Read Only | 4-13 |
| Expansion ROM Base Address (ERBA) | 0x30–0x33 | Read/Write | 4-14 |
| Capabilities Pointer (CP) | 0x34 | Read Only | 4-15 |
| Reserved | 0x35–0x37 | – | 4-15 |
| Reserved | 0x38–0x3B | – | 4-15 |
| Interrupt Line (IL) | 0x3C | Read/Write | 4-16 |
| Interrupt Pin (IP) | 0x3D | Read Only | 4-16 |
| Min_Gnt (MG) | 0x3E | Read Only | 4-17 |
| Max_Lat (ML) | 0x3F | Read Only | 4-17 |
| Capability ID (CID) | 0x40 | Read Only | 4-18 |
| Next Item Pointer (NIP) | 0x41 | Read Only | 4-18 |
| Power Management Capabilities (PMC) | 0x42–0x43 | Read Only | 4-18 |
| Power Management Control/Status (PMCSR) | 0x44–0x45 | Read/Write | 4-20 |
| Bridge Support Extensions (PMCSR_BSE) | 0x46 | Read Only | 4-21 |
| Data | 0x47 | Read Only | 4-21 |

**Table A.2 SYM53C1010-33 SCSI Register Map**

| Register Name | Address | Read/Write | Page |
|---|---|---|---|
| **SCSI Registers** | | | |
| SCSI Control Zero (SCNTL0) | 0x00 | Read/Write | 4-24 |
| SCSI Control One (SCNTL1) | 0x01 | Read/Write | 4-28 |
| SCSI Control Two (SCNTL2) | 0x02 | Read/Write | 4-30 |
| SCSI Control Three (SCNTL3) | 0x03 | Read/Write | 4-32 |
| SCSI Chip ID (SCID) | 0x04 | Read/Write | 4-33 |
| SCSI Transfer (SXFER) | 0x05 | Read/Write | 4-34 |

**Table A.2 SYM53C1010-33 SCSI Register Map (Cont.)**

| Register Name | Address | Read/Write | Page |
|---|---|---|---|
| SCSI Destination ID (SDID) | 0x06 | Read/Write | 4-35 |
| General Purpose (GPREG) | 0x07 | Read/Write | 4-36 |
| SCSI First Byte Received (SFBR) | 0x08 | Read/Write | 4-37 |
| SCSI Output Control Latch (SOCL) | 0x09 | Read/Write | 4-38 |
| SCSI Selector ID (SSID) | 0x0A | Read Only | 4-39 |
| SCSI Bus Control Lines (SBCL) | 0x0B | Read Only | 4-40 |
| DMA Status (DSTAT) | 0x0C | Read Only | 4-40 |
| SCSI Status Zero (SSTAT0) | 0x0D | Read Only | 4-43 |
| SCSI Status One (SSTAT1) | 0x0E | Read Only | 4-45 |
| SCSI Status Two (SSTAT2) | 0x0F | Read Only | 4-46 |
| Data Structure Address (DSA) | 0x10–0x13 | Read/Write | 4-47 |
| Interrupt Status Zero (ISTAT0) | 0x14 | Read/Write | 4-47 |
| Interrupt Status One (ISTAT1) | 0x15 | Read/Write | 4-51 |
| Mailbox Zero (MBOX0) | 0x16 | Read/Write | 4-52 |
| Mailbox One (MBOX1) | 0x17 | Read/Write | 4-52 |
| Chip Test Zero (CTEST0) | 0x18 | Read/Write | 4-53 |
| Chip Test One (CTEST1) | 0x19 | Read Only | 4-53 |
| Chip Test Two (CTEST2) | 0x1A | Read Only (bit 3 write) | 4-54 |
| Chip Test Three (CTEST3) | 0x1B | Read/Write | 4-55 |
| Temporary (TEMP) | 0x1C–0x1F | Read/Write | 4-56 |
| Reserved | 0x20 | – | 4-57 |
| Chip Test Four (CTEST4) | 0x21 | Read/Write | 4-57 |
| Chip Test Five (CTEST5) | 0x22 | Read/Write | 4-59 |
| Chip Test Six (CTEST6) | 0x23 | Read/Write | 4-60 |
| DMA Byte Counter (DBC) | 0x24–0x26 | Read/Write | 4-61 |

**Table A.2 SYM53C1010-33 SCSI Register Map (Cont.)**

| Register Name | Address | Read/Write | Page |
|---|---|---|---|
| DMA Command (DCMD) | 0x27 | Read/Write | 4-62 |
| DMA Next Address (DNAD) | 0x28–0x2B | Read/Write | 4-62 |
| DMA SCRIPTS Pointer (DSP) | 0x2C–0x2F | Read/Write | 4-63 |
| DMA SCRIPTS Pointer Save (DSPS) | 0x30–0x33 | Read/Write | 4-63 |
| Scratch Register A (SCRATCHA) | 0x34–0x37 | Read/Write | 4-64 |
| DMA Mode (DMODE) | 0x38 | Read/Write | 4-64 |
| DMA Interrupt Enable (DIEN) | 0x39 | Read/Write | 4-67 |
| Scratch Byte Register (SBR) | 0x3A | Read/Write | 4-68 |
| DMA Control (DCNTL) | 0x3B | Read/Write | 4-68 |
| Adder Sum Output (ADDER) | 0x3C–0x3F | Read Only | 4-71 |
| SCSI Interrupt Enable Zero (SIEN0) | 0x40 | Read/Write | 4-71 |
| SCSI Interrupt Enable One (SIEN1) | 0x41 | Read/Write | 4-73 |
| SCSI Interrupt Status Zero (SIST0) | 0x42 | Read Only | 4-75 |
| SCSI Interrupt Status One (SIST1) | 0x43 | Read Only | 4-78 |
| Reserved | 0x44 | – | 4-79 |
| SCSI Wide Residue (SWIDE) | 0x45 | Read/Write | 4-79 |
| Reserved | 0x46 | – | 4-79 |
| General Purpose Pin Control (GPCNTL) | 0x47 | Read/Write | 4-80 |
| SCSI Timer Zero (STIME0) | 0x48 | Read/Write | 4-81 |
| SCSI Timer One (STIME1) | 0x49 | Read/Write | 4-82 |
| Response ID Zero (RESPID0) | 0x4A | Read/Write | 4-84 |
| Response ID One (RESPID1) | 0x4B | Read/Write | 4-84 |
| SCSI Test Zero (STEST0) | 0x4C | Read Only | 4-85 |
| SCSI Test One (STEST1) | 0x4D | Read/Write | 4-86 |
| SCSI Test Two (STEST2) | 0x4E | Read/Write | 4-87 |

**Table A.2 SYM53C1010-33 SCSI Register Map (Cont.)**

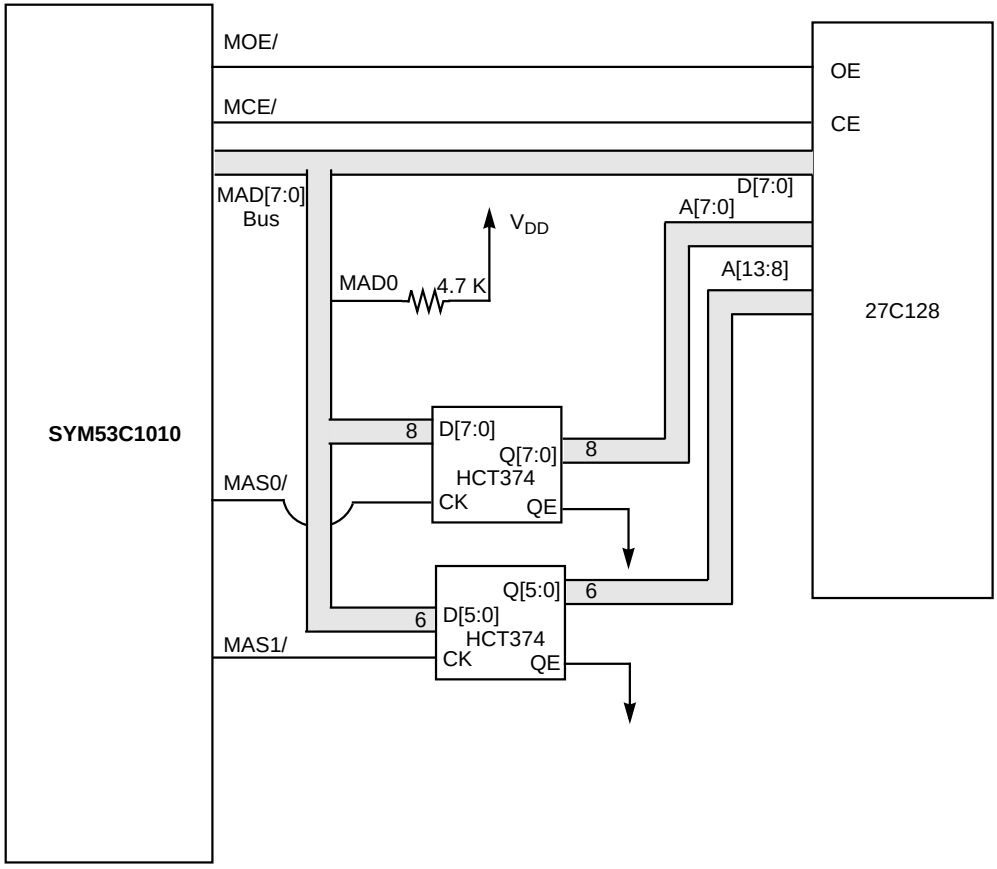
| Register Name | Address | Read/Write | Page |
|---|---|---|---|
| SCSI Test Three (STEST3) | 0x4F | Read/Write | 4-89 |
| SCSI Input Data Latch (SIDL) | 0x50–0x51 | Read Only | 4-90 |
| SCSI Test Four (STEST4) | 0x52 | Read Only | 4-91 |
| Current Inbound SCSI Offset (CSO) | 0x53 | Read Only | 4-92 |
| SCSI Output Data Latch (SODL) | 0x54–0x55 | Read/Write | 4-92 |
| Chip Control Zero (CCNTL0) | 0x56 | Read/Write | 4-93 |
| Chip Control One (CCNTL1) | 0x57 | Read/Write | 4-95 |
| SCSI Bus Data Lines (SBDL) | 0x58–0x59 | Read Only | 4-96 |
| Reserved | 0x5A | – | 4-97 |
| Chip Control Three (CCNTL3) | 0x5B | Read/Write | 4-97 |
| Scratch Register B (SCRATCHB) | 0x5C–0x5F | Read/Write | 4-98 |
| Scratch Registers C–R (SCRATCHC–SCRATCHR) | 0x60–0x9F | Read/Write | 4-99 |
| Memory Move Read Selector (MMRS) | 0xA0–0xA3 | Read/Write | 4-99 |
| Memory Move Write Selector (MMWS) | 0xA4–0xA7 | Read/Write | 4-100 |
| SCRIPT Fetch Selector (SFS) | 0xA8–0xAB | Read/Write | 4-100 |
| DSA Relative Selector (DRS) | 0xAC–0xAF | Read/Write | 4-101 |
| Static Block Move Selector (SBMS) | 0xB0–0xB3 | Read/Write | 4-101 |
| Dynamic Block Move Selector (DBMS) | 0xB4–0xB7 | Read/Write | 4-102 |
| DMA Next Address 64 (DNAD64) | 0xB8–0xBB | Read/Write | 4-102 |
| SCSI Control Four (SCNTL4) | 0xBC | Read/Write | 4-102 |
| Reserved | 0xBD | – | 4-111 |
| AIP Control Zero (AIPCNTL0) | 0xBE | Read/Write | 4-111 |
| AIP Control One (AIPCNTL1) | 0xBF | Read/Write | 4-112 |
| **Phase Mismatch Jump Registers** | | | |
| Phase Mismatch Jump Address One (PMJAD1) | 0xC0–0xC3 | Read/Write | 4-112 |

**Table A.2 SYM53C1010-33 SCSI Register Map (Cont.)**

| Register Name | Address | Read/Write | Page |
|---|---|---|---|
| Phase Mismatch Jump Address Two (PMJAD2) | 0xC4–0xC7 | Read/Write | 4-113 |
| Remaining Byte Count (RBC) | 0xC8–0xCB | Read/Write | 4-113 |
| Updated Address (UA) | 0xCC–0xCF | Read/Write | 4-114 |
| Entry Storage Address (ESA) | 0xD0–0xD3 | Read/Write | 4-115 |
| Instruction Address (IA) | 0xD4–0xD7 | Read/Write | 4-115 |
| SCSI Byte Count (SBC) | 0xD8–0xDA | Read Only | 4-116 |
| Reserved | 0xDB | – | 4-116 |
| Cumulative SCSI Byte Count (CSBC) | 0xDC–0xDF | Read/Write | 4-117 |
| CRC Pad Byte Value (CRCPAD) | 0xE0–0xE1 | Read/Write | 4-117 |
| CRC Control Zero (CRCCNTL0) | 0xE2 | Read/Write | 4-118 |
| CRC Control One (CRCCNTL1) | 0xE3 | Read/Write | 4-119 |
| CRC Data (CRCD) | 0xE4–0xE7 | Read/Write | 4-120 |
| Reserved | 0xE8–0xEF | – | 4-121 |
| DMA FIFO Byte Count (DFBC) | 0xF0–0xF1 | Read Only | 4-122 |
| Reserved | 0xF2–0xF3 | – | 4-122 |
| Reserved | 0xF4–0xFF | – | 4-122 |
| **Shadow Registers** | | | |
| Shadowed Scratch Register A (SCRATCHA) | 0x34–0x37 | Read/Write | 4-123 |
| Shadowed SCSI SGE Status 0 | 0x42 | Read/Write | 4-123 |
| Shadowed SCSI Interrupt Status One (SIST1) | 0x43 | Read Only | 4-124 |
| Shadowed Scratch Register B (SCRATCHB) | 0x5C–0x5F | Read/Write | 4-125 |
| Shadowed Scratch Registers C–R (SCRATCHC–SCRATCHR) | 0x60–0x9F | Read/Write | 4-125 |
| Shadowed Memory Move Read Selector (MMRS) | 0xA0–0xA3 | Read/Write | 4-125 |
| Shadowed Memory Move Write Selector (MMWS) | 0xA4–0xA7 | Read/Write | 4-126 |
| Shadowed SCRIPT Fetch Selector (SFS) | 0xA8–0xAB | Read/Write | 4-126 |

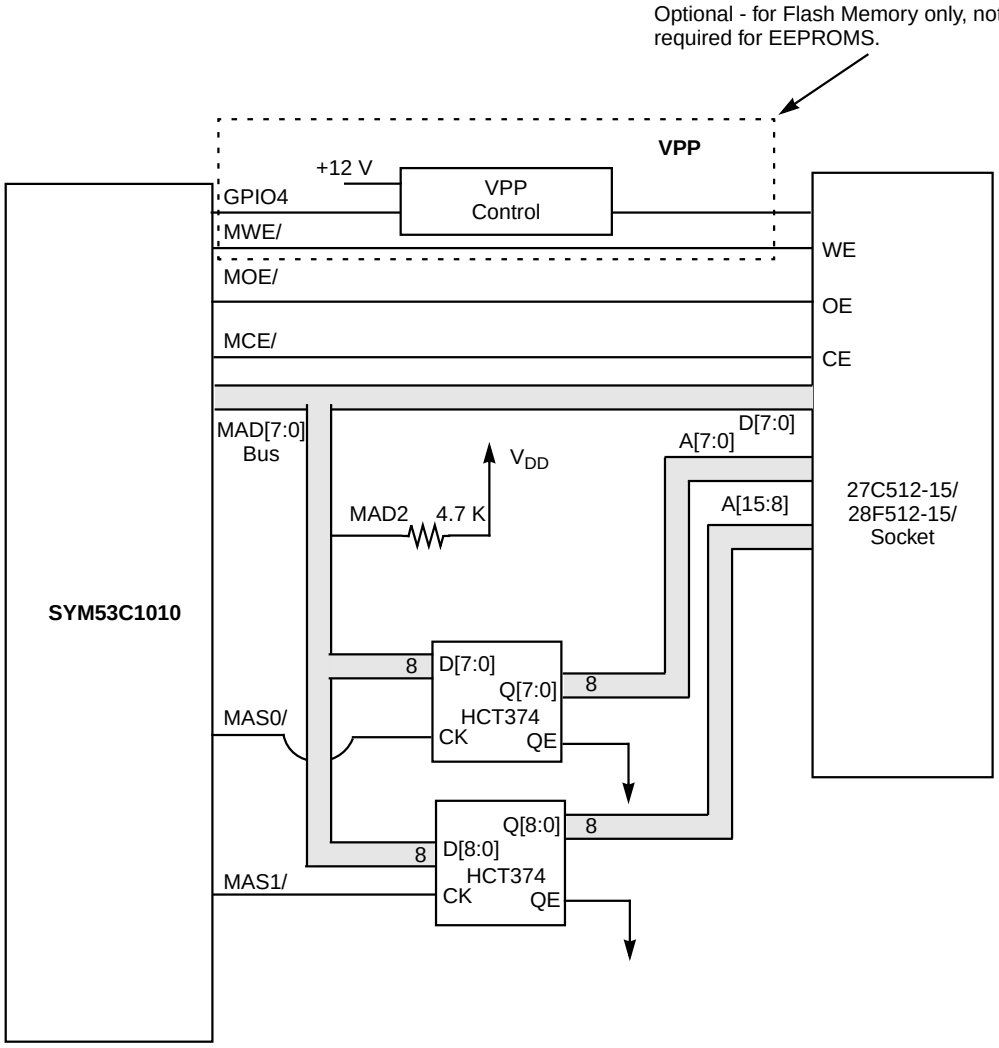
# Appendix B
# External Memory Interface Diagram Examples

**Figure B.1 16 Kbyte Interface with 200 ns Memory**



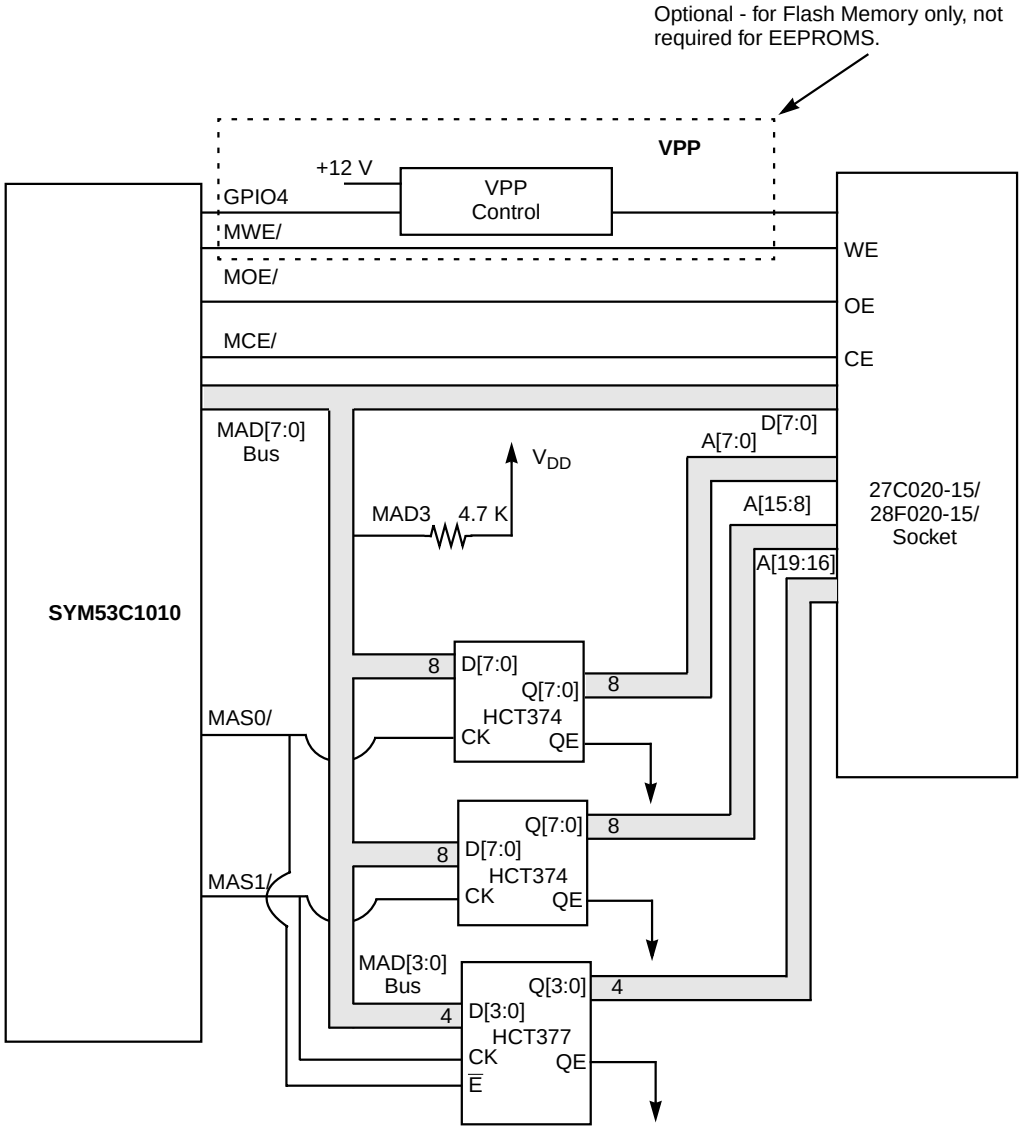

Note: MAD[3:1] pulled low internally. MAD bus sense logic enabled for 16 Kbyte of slow memory (200 ns devices @ 33 MHz).

**Figure B.2 64 Kbyte Interface with 150 ns Memory**



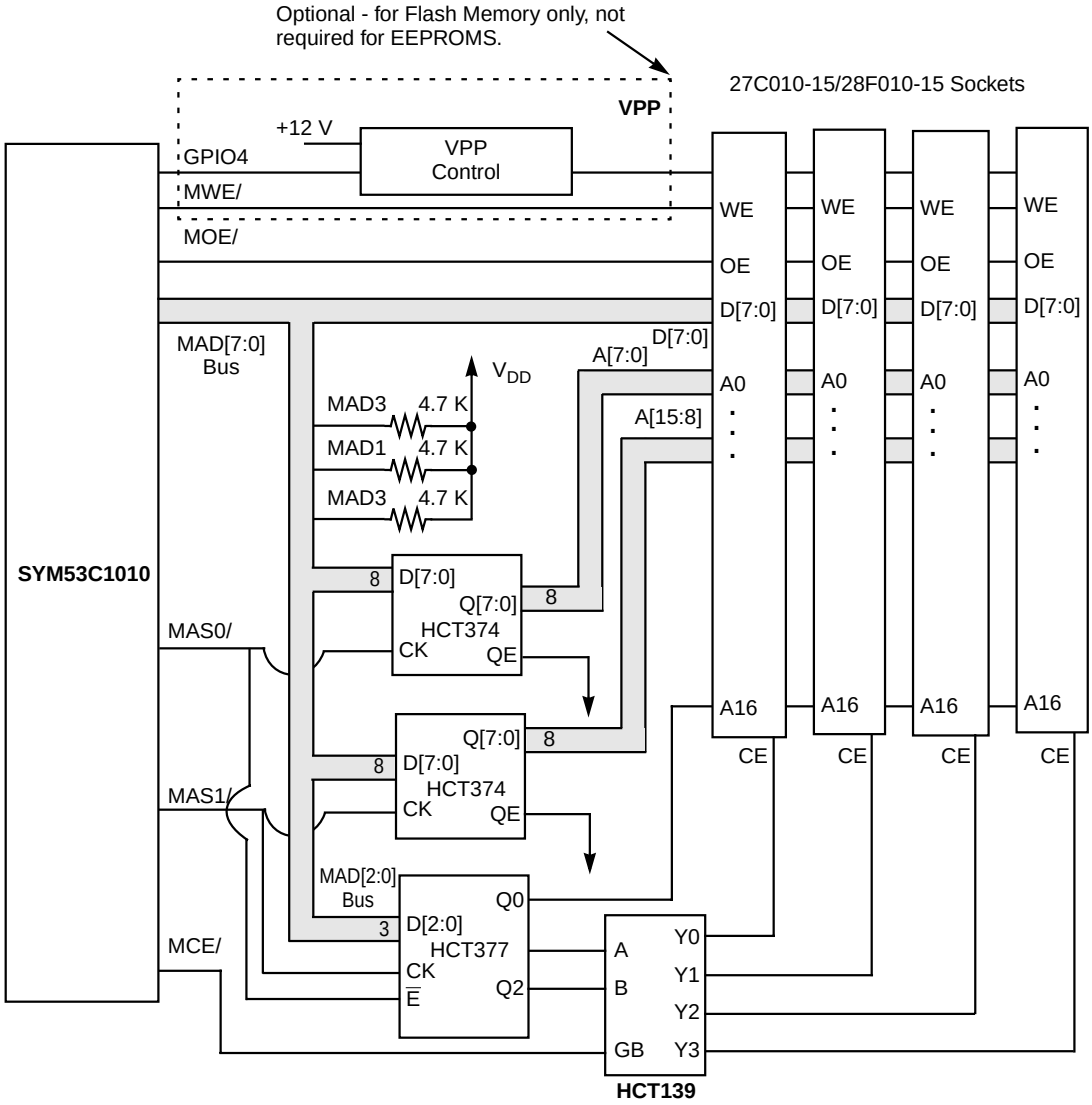

Note: MAD 3, 1, 0 pulled low internally. MAD bus sense logic enabled for 64 Kbyte of fast memory (150 ns devices @ 33 MHz).

**Figure B.3 128, 256, 512 Kbyte or 1 Mbyte Interface with 150 ns Memory**




Note: MAD[2:0] pulled low internally. MAD bus sense logic enabled for 128, 256, 512 Kbytes, or 1 Mbyte of fast memory (150 ns devices @ 33 MHz). The HCT374s may be replaced with HCT377s.

**Figure B.4 512 Kbyte Interface with 150 ns Memory**




Note: MAD2 pulled low internally. MAD bus sense logic enabled for 512 Kbytes of slow memory (150 ns devices, additional time required for HCT139 @ 33 MHz). The HCT374s may be replaced with HCT377s.

# Index

## Symbols

## Numerics



## A



## B

## C



## D

## E



## F



## G



## H



## I

## J



## L



## M

## N



## O



## P



## R

## S

## T

## U



## V



## W

# Customer Feedback

We would appreciate your feedback on this document. Please copy the following page, add your comments, and fax it to us at the number shown.

If appropriate, please also fax copies of any marked-up pages from this document.

Important: Please include your name, phone number, fax number, and company address so that we may contact you directly for clarification or additional information.

Thank you for your help in improving the quality of our documents.

## Reader's Comments

Fax your comments to: LSI Logic Corporation
Technical Publications
M/S E-198
Fax: 408.433.4333

Please tell us how you rate this document: *Symbios® SYM53C1010-33 PCI to Dual Channel Ultra3 SCSI Multifunction Controller Technical Manual.* Place a check mark in the appropriate blank for each category.

| | Excellent | Good | Average | Fair | Poor |
|---|---|---|---|---|---|
| Completeness of information | ____ | ____ | ____ | ____ | ____ |
| Clarity of information | ____ | ____ | ____ | ____ | ____ |
| Ease of finding information | ____ | ____ | ____ | ____ | ____ |
| Technical content | ____ | ____ | ____ | ____ | ____ |
| Usefulness of examples and illustrations | ____ | ____ | ____ | ____ | ____ |
| Overall manual | ____ | ____ | ____ | ____ | ____ |

What could we do to improve this document?

_____

_____

_____

_____

If you found errors in this document, please specify the error and page number. If appropriate, please fax a marked-up copy of the page(s).

_____

_____

_____

Please complete the information below so that we may contact you directly for clarification or additional information.

Name _____ Date _____

Telephone _____ Fax _____

Title _____

Department _____ Mail Stop _____

Company Name _____

Street _____

City, State, Zip _____

# U.S. Distributors by State

A. E. Avnet Electronics
http://www.hh.avnet.com
B. M. Bell Microproducts, Inc. (for HAB's)
http://www.bellmicro.com
I. E. Insight Electronics
http://www.insight-electronics.com
W. E. Wyle Electronics
http://www.wyle.com

**Alabama**
Daphne
I. E. Tel: 334.626.6190
Huntsville
A. E. Tel: 256.837.8700
B. M. Tel: 256.705.3559
I. E. Tel: 256.830.1222
W. E. Tel: 800.964.9953

**Alaska**
A. E. Tel: 800.332.8638

**Arizona**
Phoenix
A. E. Tel: 480.736.7000
B. M. Tel: 602.267.9551
W. E. Tel: 800.528.4040
Tempe
I. E. Tel: 480.829.1800
Tucson
A. E. Tel: 520.742.0515

**Arkansas**
W. E. Tel: 972.235.9953

**California**
Agoura Hills
B. M. Tel: 818.865.0266
Granite Bay
B. M. Tel: 916.523.7047
Irvine
A. E. Tel: 949.789.4100
B. M. Tel: 949.470.2900
I. E. Tel: 949.727.3291
W. E. Tel: 800.626.9953
Los Angeles
A. E. Tel: 818.594.0404
W. E. Tel: 800.288.9953
Sacramento
A. E. Tel: 916.632.4500
W. E. Tel: 800.627.9953
San Diego
A. E. Tel: 858.385.7500
B. M. Tel: 858.597.3010
I. E. Tel: 800.677.6011
W. E. Tel: 800.829.9953
San Jose
A. E. Tel: 408.435.3500
B. M. Tel: 408.436.0881
I. E. Tel: 408.952.7000
Santa Clara
W. E. Tel: 800.866.9953
Woodland Hills
A. E. Tel: 818.594.0404
Westlake Village
I. E. Tel: 818.707.2101

**Colorado**
Denver
A. E. Tel: 303.790.1662
B. M. Tel: 303.846.3065
W. E. Tel: 800.933.9953
Englewood
I. E. Tel: 303.649.1800
Idaho Springs
B. M. Tel: 303.567.0703

**Connecticut**
Cheshire
A. E. Tel: 203.271.5700
I. E. Tel: 203.272.5843
Wallingford
W. E. Tel: 800.605.9953

**Delaware**
North/South
A. E. Tel: 800.526.4812
Tel: 800.638.5988
B. M. Tel: 302.328.8968
W. E. Tel: 856.439.9110

**Florida**
Altamonte Springs
B. M. Tel: 407.682.1199
I. E. Tel: 407.834.6310
Boca Raton
I. E. Tel: 561.997.2540
Bonita Springs
B. M. Tel: 941.498.6011
Clearwater
I. E. Tel: 727.524.8850
Fort Lauderdale
A. E. Tel: 954.484.5482
W. E. Tel: 800.568.9953
Miami
B. M. Tel: 305.477.6406
Orlando
A. E. Tel: 407.657.3300
W. E. Tel: 407.740.7450
Tampa
W. E. Tel: 800.395.9953
St. Petersburg
A. E. Tel: 727.507.5000

**Georgia**
Atlanta
A. E. Tel: 770.623.4400
B. M. Tel: 770.980.4922
W. E. Tel: 800.876.9953
Duluth
I. E. Tel: 678.584.0812

**Hawaii**
A. E. Tel: 800.851.2282

**Idaho**
A. E. Tel: 801.365.3800
W. E. Tel: 801.974.9953

**Illinois**
North/South
A. E. Tel: 847.797.7300
Tel: 314.291.5350
Chicago
B. M. Tel: 847.413.8530
W. E. Tel: 800.853.9953
Schaumburg
I. E. Tel: 847.885.9700

**Indiana**
Fort Wayne
I. E. Tel: 219.436.4250
W. E. Tel: 888.358.9953
Indianapolis
A. E. Tel: 317.575.3500

**Iowa**
W. E. Tel: 612.853.2280
Cedar Rapids
A. E. Tel: 319.393.0033

**Kansas**
W. E. Tel: 303.457.9953
Kansas City
A. E. Tel: 913.663.7900
Lenexa
I. E. Tel: 913.492.0408

**Kentucky**
W. E. Tel: 937.436.9953
Central/Northern/ Western
A. E. Tel: 800.984.9503
Tel: 800.767.0329
Tel: 800.829.0146

**Louisiana**
W. E. Tel: 713.854.9953
North/South
A. E. Tel: 800.231.0253
Tel: 800.231.5775

**Maine**
A. E. Tel: 800.272.9255
W. E. Tel: 781.271.9953

**Maryland**
Baltimore
A. E. Tel: 410.720.3400
W. E. Tel: 800.863.9953
Columbia
B. M. Tel: 800.673.7461
I. E. Tel: 410.381.3131

**Massachusetts**
Boston
A. E. Tel: 978.532.9808
W. E. Tel: 800.444.9953
Burlington
I. E. Tel: 781.270.9400
Marlborough
B. M. Tel: 800.673.7459
Woburn
B. M. Tel: 800.552.4305

**Michigan**
Brighton
I. E. Tel: 810.229.7710
Detroit
A. E. Tel: 734.416.5800
W. E. Tel: 888.318.9953
Clarkston
B. M. Tel: 877.922.9363

**Minnesota**
Champlin
B. M. Tel: 800.557.2566
Eden Prairie
B. M. Tel: 800.255.1469
Minneapolis
A. E. Tel: 612.346.3000
W. E. Tel: 800.860.9953
St. Louis Park
I. E. Tel: 612.525.9999

**Mississippi**
A. E. Tel: 800.633.2918
W. E. Tel: 256.830.1119

**Missouri**
W. E. Tel: 630.620.0969
St. Louis
A. E. Tel: 314.291.5350
I. E. Tel: 314.872.2182

**Montana**
A. E. Tel: 800.526.1741
W. E. Tel: 801.974.9953

**Nebraska**
A. E. Tel: 800.332.4375
W. E. Tel: 303.457.9953

**Nevada**
Las Vegas
A. E. Tel: 800.528.8471
W. E. Tel: 702.765.7117

**New Hampshire**
A. E. Tel: 800.272.9255
W. E. Tel: 781.271.9953

**New Jersey**
North/South
A. E. Tel: 201.515.1641
Tel: 609.222.6400
Mt. Laurel
I. E. Tel: 856.222.9566
Pine Brook
B. M. Tel: 973.244.9668
W. E. Tel: 800.862.9953
Parsippany
I. E. Tel: 973.299.4425
Wayne
W. E. Tel: 973.237.9010

**New Mexico**
W. E. Tel: 480.804.7000
Albuquerque
A. E. Tel: 505.293.5119

## U.S. Distributors by State (Continued)

**New York**
Hauppauge
I. E. Tel: 516.761.0960
Long Island
A. E. Tel: 516.434.7400
W. E. Tel: 800.861.9953
Rochester
A. E. Tel: 716.475.9130
I. E. Tel: 716.242.7790
W. E. Tel: 800.319.9953
Smithtown
B. M. Tel: 800.543.2008
Syracuse
A. E. Tel: 315.449.4927

**North Carolina**
Raleigh
A. E. Tel: 919.859.9159
I. E. Tel: 919.873.9922
W. E. Tel: 800.560.9953

**North Dakota**
A. E. Tel: 800.829.0116
W. E. Tel: 612.853.2280

**Ohio**
Cleveland
A. E. Tel: 216.498.1100
W. E. Tel: 800.763.9953
Dayton
A. E. Tel: 614.888.3313
I. E. Tel: 937.253.7501
W. E. Tel: 800.575.9953
Strongsville
B. M. Tel: 440.238.0404
Valley View
I. E. Tel: 216.520.4333

**Oklahoma**
W. E. Tel: 972.235.9953
Tulsa
A. E. Tel: 918.459.6000
I. E. Tel: 918.665.4664

**Oregon**
Beaverton
B. M. Tel: 503.524.1075
I. E. Tel: 503.644.3300
Portland
A. E. Tel: 503.526.6200
W. E. Tel: 800.879.9953

**Pennsylvania**
Mercer
I. E. Tel: 412.662.2707
Philadelphia
A. E. Tel: 800.526.4812
B. M. Tel: 877.351.2355
W. E. Tel: 800.871.9953
Pittsburgh
A. E. Tel: 412.281.4150
W. E. Tel: 440.248.9996

**Rhode Island**
A. E. 800.272.9255
W. E. Tel: 781.271.9953

**South Carolina**
A. E. Tel: 919.872.0712
W. E. Tel: 919.469.1502

**South Dakota**
A. E. Tel: 800.829.0116
W. E. Tel: 612.853.2280

**Tennessee**
W. E. Tel: 256.830.1119
East/West
A. E. Tel: 800.241.8182
Tel: 800.633.2918

**Texas**
Arlington
B. M. Tel: 817.417.5993
Austin
A. E. Tel: 512.219.3700
B. M. Tel: 512.258.0725
I. E. Tel: 512.719.3090
W. E. Tel: 800.365.9953
Dallas
A. E. Tel: 214.553.4300
B. M. Tel: 972.783.4191
W. E. Tel: 800.955.9953
El Paso
A. E. Tel: 800.526.9238
Houston
A. E. Tel: 713.781.6100
B. M. Tel: 713.917.0663
W. E. Tel: 800.888.9953
Richardson
I. E. Tel: 972.783.0800
Rio Grande Valley
A. E. Tel: 210.412.2047
Stafford
I. E. Tel: 281.277.8200

**Utah**
Centerville
B. M. Tel: 801.295.3900
Murray
I. E. Tel: 801.288.9001
Salt Lake City
A. E. Tel: 801.365.3800
W. E. Tel: 800.477.9953

**Vermont**
A. E. Tel: 800.272.9255
W. E. Tel: 716.334.5970

**Virginia**
A. E. Tel: 800.638.5988
W. E. Tel: 301.604.8488
Haymarket
B. M. Tel: 703.754.3399
Springfield
B. M. Tel: 703.644.9045

**Washington**
Kirkland
I. E. Tel: 425.820.8100
Maple Valley
B. M. Tel: 206.223.0080
Seattle
A. E. Tel: 425.882.7000
W. E. Tel: 800.248.9953

**West Virginia**
A. E. Tel: 800.638.5988

**Wisconsin**
Milwaukee
A. E. Tel: 414.513.1500
W. E. Tel: 800.867.9953
Wauwatosa
I. E. Tel: 414.258.5338

**Wyoming**
A. E. Tel: 800.332.9326
W. E. Tel: 801.974.9953

# Direct Sales Representatives by State (Component and HAB)

E. A. Earle Associates
E. L. Electrodyne - UT
GRP Group 2000
I. S. Infinity Sales, Inc.
ION ION Associates, Inc.
R. A. Rathsburg Associates, Inc.
SGY Synergy Associates, Inc.

**Arizona**
Tempe
E. A. Tel: 480.921.3305

**California**
Calabasas
I. S. Tel: 818.880.6480
Irvine
I. S. Tel: 714.833.0300
San Diego
E. A. Tel: 619.278.5441

**Illinois**
Elmhurst
R. A. Tel: 630.516.8400

**Indiana**
Cicero
R. A. Tel: 317.984.8608
Ligonier
R. A. Tel: 219.894.3184
Plainfield
R. A. Tel: 317.838.0360

**Massachusetts**
Burlington
SGY Tel: 781.238.0870

**Michigan**
Byron Center
R. A. Tel: 616.554.1460
Good Rich
R. A. Tel: 810.636.6060
Novi
R. A. Tel: 810.615.4000

**North Carolina**
Cary
GRP Tel: 919.481.1530

**Ohio**
Columbus
R. A. Tel: 614.457.2242
Dayton
R. A. Tel: 513.291.4001
Independence
R. A. Tel: 216.447.8825

**Pennsylvania**
Somerset
R. A. Tel: 814.445.6976

**Texas**
Austin
ION Tel: 512.794.9006
Arlington
ION Tel: 817.695.8000
Houston
ION Tel: 281.376.2000

**Utah**
Salt Lake City
E. L. Tel: 801.264.8050

**Wisconsin**
Muskego
R. A. Tel: 414.679.8250
Saukville
R. A. Tel: 414.268.1152

# Sales Offices and Design Resource Centers

**LSI Logic Corporation**
**Corporate Headquarters**
**1551 McCarthy Blvd**
**Milpitas CA 95035**
**Tel: 408.433.8000**
**Fax: 408.433.8989**

**NORTH AMERICA**

**California**
Irvine
18301 Von Karman Ave
Suite 900
Irvine, CA 92612
♦ Tel: 949.809.4600
Fax: 949.809.4444

Pleasanton Design Center
5050 Hopyard Road, 3rd Floor
Suite 300
Pleasanton, CA 94588
Tel: 925.730.8800
Fax: 925.730.8700

San Diego
7585 Ronson Road
Suite 100
San Diego, CA 92111
Tel: 858.467.6981
Fax: 858.496.0548

Silicon Valley
1551 McCarthy Blvd
Sales Office
M/S C-500
Milpitas, CA 95035
♦ Tel: 408.433.8000
Fax: 408.954.3353
Design Center
M/S C-410
Tel: 408.433.8000
Fax: 408.433.7695

Wireless Design Center
11452 El Camino Real
Suite 210
San Diego, CA 92130
Tel: 858.350.5560
Fax: 858.350.0171

**Colorado**
Boulder
4940 Pearl East Circle
Suite 201
Boulder, CO 80301
♦ Tel: 303.447.3800
Fax: 303.541.0641

Colorado Springs
4420 Arrowswest Drive
Colorado Springs, CO 80907
Tel: 719.533.7000
Fax: 719.533.7020

Fort Collins
2001 Danfield Court
Fort Collins, CO 80525
Tel: 970.223.5100
Fax: 970.206.5549

**Florida**
Boca Raton
2255 Glades Road
Suite 324A
Boca Raton, FL 33431
Tel: 561.989.3236
Fax: 561.989.3237

**Georgia**
Alpharetta
2475 North Winds Parkway
Suite 200
Alpharetta, GA 30004
Tel: 770.753.6146
Fax: 770.753.6147

**Illinois**
Oakbrook Terrace
Two Mid American Plaza
Suite 800
Oakbrook Terrace, IL 60181
Tel: 630.954.2234
Fax: 630.954.2235

**Kentucky**
Bowling Green
1262 Chestnut Street
Bowling Green, KY 42101
Tel: 270.793.0010
Fax: 270.793.0040

**Maryland**
Bethesda
6903 Rockledge Drive
Suite 230
Bethesda, MD 20817
Tel: 301.897.5800
Fax: 301.897.8389

**Massachusetts**
Waltham
200 West Street
Waltham, MA 02451
♦ Tel: 781.890.0180
Fax: 781.890.6158

Burlington - Mint Technology
77 South Bedford Street
Burlington, MA 01803
Tel: 781.685.3800
Fax: 781.685.3801

**Minnesota**
Minneapolis
8300 Norman Center Drive
Suite 730
Minneapolis, MN 55437
♦ Tel: 612.921.8300
Fax: 612.921.8399

**New Jersey**
Red Bank
125 Half Mile Road
Suite 200
Red Bank, NJ 07701
Tel: 732.933.2656
Fax: 732.933.2643

Cherry Hill - Mint Technology
215 Longstone Drive
Cherry Hill, NJ 08003
Tel: 856.489.5530
Fax: 856.489.5531

**New York**
Fairport
550 Willowbrook Office Park
Fairport, NY 14450
Tel: 716.218.0020
Fax: 716.218.9010

**North Carolina**
Raleigh
Phase II
4601 Six Forks Road
Suite 528
Raleigh, NC 27609
Tel: 919.785.4520
Fax: 919.783.8909

**Oregon**
Beaverton
15455 NW Greenbrier Parkway
Suite 235
Beaverton, OR 97006
Tel: 503.645.0589
Fax: 503.645.6612

**Texas**
Austin
9020 Capital of TX Highway North
Building 1
Suite 150
Austin, TX 78759
Tel: 512.388.7294
Fax: 512.388.4171

Plano
500 North Central Expressway
Suite 440
Plano, TX 75074
♦ Tel: 972.244.5000
Fax: 972.244.5001

Houston
20405 State Highway 249
Suite 450
Houston, TX 77070
Tel: 281.379.7800
Fax: 281.379.7818

**Canada**
**Ontario**
Ottawa
260 Hearst Way
Suite 400
Kanata, ON K2L 3H1
♦ Tel: 613.592.1263
Fax: 613.592.3253

**INTERNATIONAL**

**France**
Paris
**LSI Logic S.A.**
**Immeuble Europa**
53 bis Avenue de l'Europe
B.P. 139
78148 Velizy-Villacoublay
Cedex, Paris
♦ Tel: 33.1.34.63.13.13
Fax: 33.1.34.63.13.19

**Germany**
Munich
**LSI Logic GmbH**
Orleansstrasse 4
81669 Munich
♦ Tel: 49.89.4.58.33.0
Fax: 49.89.4.58.33.108

Stuttgart
Mittlerer Pfad 4
D-70499 Stuttgart
♦ Tel: 49.711.13.96.90
Fax: 49.711.86.61.428

**Italy**
Milan
**LSI Logic S.P.A.**
Centro Direzionale Colleoni Palazzo
Orione Ingresso 1
20041 Agrate Brianza, Milano
♦ Tel: 39.039.687371
Fax: 39.039.6057867

**Japan**
Tokyo
**LSI Logic K.K.**
Rivage-Shinagawa Bldg. 14F
4-1-8 Kounan
Minato-ku, Tokyo 108-0075
♦ Tel: 81.3.5463.7821
Fax: 81.3.5463.7820

Osaka
Crystal Tower 14F
1-2-27 Shiromi
Chuo-ku, Osaka 540-6014
♦ Tel: 81.6.947.5281
Fax: 81.6.947.5287

# Sales Offices and Design Resource Centers (Continued)

**Korea**
Seoul
**LSI Logic Corporation of Korea Ltd**
10th Fl., Haesung 1 Bldg.
942, Daechi-dong,
Kangnam-ku, Seoul, 135-283
Tel: 82.2.528.3400
Fax: 82.2.528.2250

**The Netherlands**
Eindhoven
**LSI Logic Europe Ltd**
World Trade Center Eindhoven
Building 'Rijder'
Bogert 26
5612 LZ Eindhoven
Tel: 31.40.265.3580
Fax: 31.40.296.2109

**Singapore**
Singapore
**LSI Logic Pte Ltd**
7 Temasek Boulevard
#28-02 Suntec Tower One
Singapore 038987
Tel: 65.334.9061
Fax: 65.334.4749

**Sweden**
Stockholm
**LSI Logic AB**
Finlandsgatan 14
164 74 Kista
♦ Tel: 46.8.444.15.00
Fax: 46.8.750.66.47

**Taiwan**
Taipei
**LSI Logic Asia, Inc.**
**Taiwan Branch**
10/F 156 Min Sheng E. Road
Section 3
Taipei, Taiwan R.O.C.
Tel: 886.2.2718.7828
Fax: 886.2.2718.8869

**United Kingdom**
Bracknell
**LSI Logic Europe Ltd**
Greenwood House
London Road
Bracknell, Berkshire RG12 2UB
♦ Tel: 44.1344.426544
Fax: 44.1344.481039

♦ Sales Offices with Design Resource Centers

# International Distributors

**Australia**
New South Wales
**Reptechnic Pty Ltd**
3/36 Bydown Street
Neutral Bay, NSW 2089
♦ Tel: 612.9953.9844
Fax: 612.9953.9683

**Belgium**
**Acal nv/sa**
Lozenberg 4
1932 Zaventem
Tel: 32.2.7205983
Fax: 32.2.7251014

**China**
Beijing
**LSI Logic International Services Inc.**
**Beijing Representative Office**
Room 708
Canway Building
66 Nan Li Shi Lu
Xicheng District
Beijing 100045, China
Tel: 86.10.6804.2534 to 38
Fax: 86.10.6804.2521

**France**
Rungis Cedex
**Azzurri Technology France**
22 Rue Saarinen
Sillic 274
94578 Rungis Cedex
Tel: 33.1.41806310
Fax: 33.1.41730340

**Germany**
Haar
**EBV Elektronik**
Hans-Pinsel Str. 4
D-85540 Haar
Tel: 49.89.4600980
Fax: 49.89.46009840

Munich
**Avnet Emg GmbH**
Stahlgruberring 12
81829 Munich
Tel: 49.89.45110102
Fax: 49.89.42.27.75

Wuennenberg-Haaren
**Peacock AG**
Graf-Zepplin-Str 14
D-33181 Wuennenberg-Haaren
Tel: 49.2957.79.1692
Fax: 49.2957.79.9341

**Hong Kong**
Hong Kong
**AVT Industrial Ltd**
Unit 608 Tower 1
Cheung Sha Wan Plaza
833 Cheung Sha Wan Road
Kowloon, Hong Kong
Tel: 852.2428.0008
Fax: 852.2401.2105

**Serial System (HK) Ltd**
2301 Nanyang Plaza
57 Hung To Road, Kwun Tong
Kowloon, Hong Kong
Tel: 852.2995.7538
Fax: 852.2950.0386

**India**
Bangalore
**Spike Technologies India Private Ltd**
951, Vijayalakshmi Complex,
2nd Floor, 24th Main,
J P Nagar II Phase,
Bangalore, India 560078
♦ Tel: 91.80.664.5530
Fax: 91.80.664.9748

**Israel**
Tel Aviv
**Eastronics Ltd**
11 Rozanis Street
P.O. Box 39300
Tel Aviv 61392
Tel: 972.3.6458777
Fax: 972.3.6458666

**Japan**
Tokyo
**Daito Electron**
Sogo Kojimachi No.3 Bldg
1-6 Kojimachi
Chiyoda-ku, Tokyo 102-8730
Tel: 81.3.3264.0326
Fax: 81.3.3261.3984

**Global Electronics Corporation**
Nichibei Time24 Bldg. 35 Tansu-cho
Shinjuku-ku, Tokyo 162-0833
Tel: 81.3.3260.1411
Fax: 81.3.3260.7100
Technical Center
Tel: 81.471.43.8200

**Marubeni Solutions**
1-26-20 Higashi
Shibuya-ku, Tokyo 150-0001
Tel: 81.3.5778.8662
Fax: 81.3.5778.8669

**Shinki Electronics**
Myuru Daikanyama 3F
3-7-3 Ebisu Minami
Shibuya-ku, Tokyo 150-0022
Tel: 81.3.3760.3110
Fax: 81.3.3760.3101

Yokohama-City
**Innotech**
2-15-10 Shin Yokohama
Kohoku-ku
Yokohama-City, 222-8580
Tel: 81.45.474.9037
Fax: 81.45.474.9065

**Macnica Corporation**
Hakusan High-Tech Park
1-22-2 Hadusan, Midori-Ku,
Yokohama-City, 226-8505
Tel: 81.45.939.6140
Fax: 81.45.939.6141

**The Netherlands**
Eindhoven
**Acal Nederland b.v.**
Beatrix de Rijkweg 8
5657 EG Eindhoven
Tel: 31.40.2.502602
Fax: 31.40.2.510255

**Switzerland**
Brugg
**LSI Logic Sulzer AG**
Mattenstrasse 6a
CH 2555 Brugg
Tel: 41.32.3743232
Fax: 41.32.3743233

**Taiwan**
Taipei
**Avnet-Mercuries Corporation, Ltd**
14F, No. 145,
Sec. 2, Chien Kuo N. Road
Taipei, Taiwan, R.O.C.
Tel: 886.2.2516.7303
Fax: 886.2.2505.7391

**Lumax International Corporation, Ltd**
7th Fl., 52, Sec. 3
Nan-Kang Road
Taipei, Taiwan, R.O.C.
Tel: 886.2.2788.3656
Fax: 886.2.2788.3568

**Prospect Technology Corporation, Ltd**
4Fl., No. 34, Chu Luen Street
Taipei, Taiwan, R.O.C.
Tel: 886.2.2721.9533
Fax: 886.2.2773.3756

**Wintech Microeletronics Co., Ltd**
7F., No. 34, Sec. 3, Pateh Road
Taipei, Taiwan, R.O.C.
Tel: 886.2.2579.5858
Fax: 886.2.2570.3123

**United Kingdom**
Maidenhead
**Azzurri Technology Ltd**
16 Grove Park Business Estate
Waltham Road
White Waltham
Maidenhead, Berkshire SL6 3LW
Tel: 44.1628.826826
Fax: 44.1628.829730

Milton Keynes
**Ingram Micro (UK) Ltd**
Garamonde Drive
Wymbush
Milton Keynes
Buckinghamshire MK8 8DF
Tel: 44.1908.260422

Swindon
**EBV Elektronik**
12 Interface Business Park
Bincknoll Lane
Wootton Bassett,
Swindon, Wiltshire SN4 8SY
Tel: 44.1793.849933
Fax: 44.1793.859555

♦ Sales Offices with Design Resource Centers