TECHNICAL MANUAL

LSI53C895A
PCI to Ultra2
SCSI Controller

Version 2.2

April 2001

LSI LOGIC®

S14028.B

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

Document DB14-000089-03, Fourth Edition (April 2001)
This document describes the LSI Logic LSI53C895A PCI to Ultra2 SCSI Controller and will remain the official reference source for all revisions/releases of this product until rescinded by an update.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) X3710-1142D.

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

SR/HH

Preface

This book is the primary reference and technical manual for the LSI53C895A PCI to Ultra2 SCSI Controller. It contains a complete functional description for the product and also includes complete physical and electrical specifications.

Audience

This manual provides reference information on the LSI53C895A PCI to Ultra2 SCSI Controller. It is intended for system designers and programmers who are using this device to design an Ultra2 SCSI port for PCI-based personal computers, workstations, servers or embedded applications.

Organization

This document has the following chapters and appendixes:

- Chapter 1, **General Description**, includes general information about the LSI53C895A.
- Chapter 2, **Functional Description**, describes the main functional areas of the chip in more detail, including interfaces to the SCSI bus and external memory.
- Chapter 3, **Signal Descriptions**, contains pin diagrams 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 LSI53C895A.

- Chapter 6, **Electrical 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 LSI53C895A to external ROMs.

Related Publications

For background information, 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); 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 World Wide Web Home Page
www.lsilogic.com

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	Advance information version of the manual.
1.0	7/99	Preliminary version of the manual.
1.1	9/99	PCI timings corrected in Chapter 6, Table 6.3, and Figure 6.41 corrected.
2.0	2/00	Final version.
2.1	7/00	Added Figure 6.43.
2.2	4/01	All product names changed from SYM to LSI. Updated DC electrical specifications and test conditions.

Contents

Figures

Tables

Chapter 1
General Description

Chapter 1 is divided into the following sections:

The LSI53C895A PCI to Ultra2 SCSI Controller brings Ultra2 SCSI performance to host adapter, workstation, and general computer designs, making it easy to add a high-performance SCSI bus to any PCI system. It supports Ultra2 SCSI transfer rates and allows increased SCSI connectivity and cable length with Low Voltage Differential (LVD) signaling for SCSI devices. The LSI53C895A is packaged in a 208 Plastic Quad Flat Pack (PQFP) and a 272 Ball Grid Array (BGA). The LSI53C895A can be used as a drop-in replacement for the LSI53C895.

The LSI53C895A has a local memory bus for local storage of the device’s BIOS ROM in flash memory or standard EEPROMs. The LSI53C895A supports programming of local flash memory for updates to BIOS. Appendix B, “External Memory Interface Diagram Examples,” has system diagrams showing the connections of the LSI53C895A with an external ROM or flash memory.

LVDlink™ technology is the LSI Logic implementation of LVD. LVDlink transceivers allow the LSI53C895A to perform either Single-Ended (SE) or LVD transfers. It also supports external High Voltage Differential (HVD) transceivers. The LSI53C895A integrates a high-performance SCSI core, a 64-bit PCI bus master DMA core, and the LSI Logic SCSI SCRIPTS™ processor to meet the flexibility requirements of SCSI-3 and Ultra2 SCSI

standards. It implements multithreaded I/O algorithms with a minimum of processor intervention, solving the protocol overhead problems of previous intelligent and nonintelligent adapter designs.

Figure 1.1 illustrates a typical LSI53C895A system and Figure 1.2 illustrates a typical LSI53C895A board application.

Figure 1.1 Typical LSI53C895A System Application

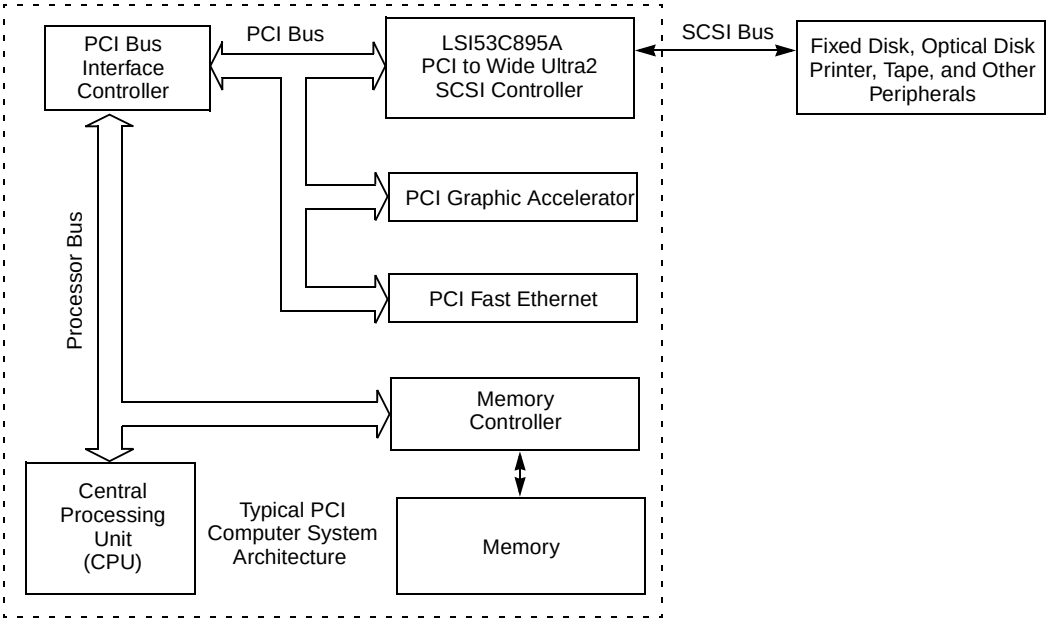
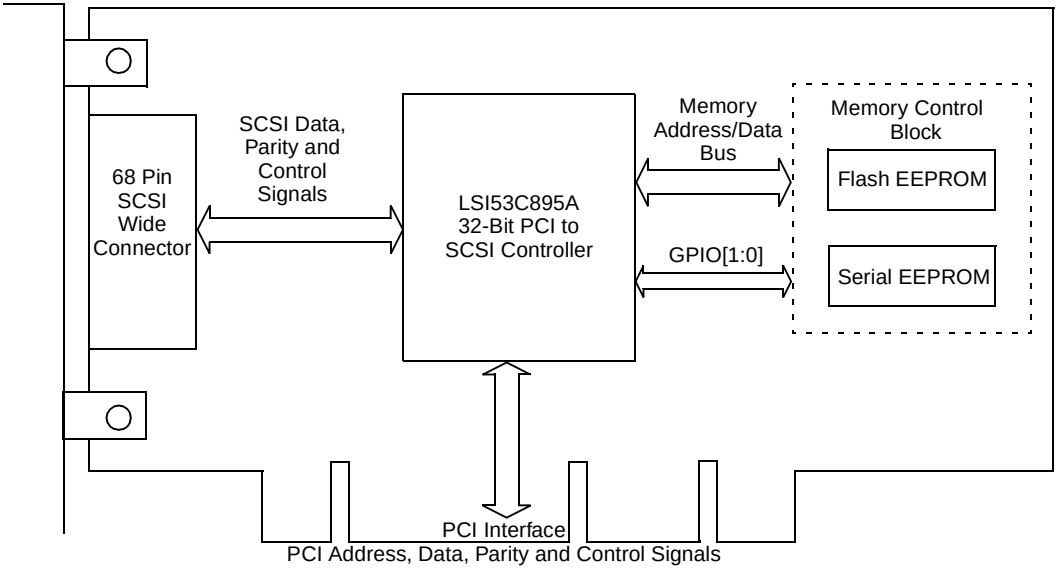


Figure 1.2 Typical LSI53C895A Board Application



1.1 New Features in the LSI53C895A

The LSI53C895A is a drop-in replacement for the LSI53C895 PCI to Ultra2 SCSI Controller, with these additional benefits:

- Supports 32-bit PCI Interface with 64-bit addressing.
- Handles SCSI phase mismatches in SCRIPTS without interrupting the CPU.
- Supports JTAG boundary scanning.
- Supports RAID ready alternative interrupt signaling.
- Supports PC99 Power Management.
  - Automatically downloads Subsystem Vendor ID, Subsystem ID, and PCI power management levels D0, D1, D2, and D3.
- Improves PCI bus efficiency through improved PCI caching design.
- Transfers Load/Store data to or from 8 Kbytes of internal SCRIPTS RAM.

Additional features of the LSI53C895A include:

- Hardware control of SCSI activity LED.
- Nine GPIO Pins.
- 32-bit ISTAT registers (Interrupt Status Zero (ISTAT0), Interrupt Status One (ISTAT1), Mailbox Zero (MBOX0), and Mailbox One (MBOX1)).
- Optional 944 byte DMA FIFO supports large block transfers at Ultra2 SCSI speeds. The default FIFO size of 112 bytes is also supported.

1.2 Benefits of Ultra2 SCSI

Ultra2 SCSI is an extension of the SPI-2 draft standard that allows faster synchronous SCSI transfer rates. It also defines a new physical layer, LVD SCSI, that provides an incremental evolution from SCSI-2 and Ultra SCSI. When enabled, Ultra2 SCSI performs 40 megatransfers per second, resulting in approximately twice the synchronous transfer rates of Ultra SCSI. The LSI53C895A can perform 16-bit, Ultra2 SCSI synchronous transfers as fast as 80 Mbytes/s. This advantage is most noticeable in heavily loaded systems or with applications with large block requirements, such as video on-demand and image processing.

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

1.3 Benefits of LVDlink

The LSI53C895A supports LVD for SCSI. 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 HVD SCSI without the added cost of external differential

transceivers. Ultra2 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 LSI53C895A features universal LVDlink transceivers that can support LVD SCSI, SE, and HVD modes. The LVDlink technology also supports HVD signaling in legacy systems when external transceivers are connected to the LSI53C895A. This allows use of the LSI53C895A in both legacy and Ultra2 SCSI applications.

1.4 TolerANT® Technology

The LSI53C895A features TolerANT technology, which includes active negation on the SCSI drivers and input signal filtering on the SCSI receivers. Active negation actively drives the SCSI Request, Acknowledge, Data, and Parity signals HIGH rather than allowing them to be passively pulled up by terminators. Active negation is enabled by setting bit 7 in the SCSI Test Three (STEST3) register.

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, the single biggest reliability issue with SCSI operations.

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, so other devices on the bus are also protected from data corruption. When used with the LVDlink 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.

1.5 LSI53C895A Benefits Summary

This section of the chapter provides an overview of the LSI53C895A features and benefits. It contains these topics:

1.5.1 SCSI Performance

To improve SCSI performance, the LSI53C895A:

- Has integrated LVDlink universal transceivers which:
  - Support SE, LVD, and HVD signals (with external transceivers).
  - Allow greater device connectivity and longer cable length.
  - Save the cost of external differential transceivers.
  - Support a long-term performance migration path.
- Bursts up to 512 bytes across the PCI bus through its 944 byte FIFO.
- Performs wide, Ultra2 SCSI synchronous transfers as fast as 80 Mbytes/s.
- Can handle phase mismatches in SCRIPTS without interrupting the system processor, eliminating the need for CPU intervention during an I/O disconnect/reselect sequence.
- Achieve Ultra2 SCSI transfer rates with an input frequency of 40 MHz with the on-chip SCSI clock quadrupler.
- Includes 8 Kbytes internal RAM for SCRIPTS instruction storage.
- Has 31 levels of SCSI synchronous offset.
- Supports variable block size and scatter/gather data transfers.

- Performs sustained memory-to-memory DMA transfers to approximately 100 Mbytes/s.
- Minimizes 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.
- Uses Load/Store SCRIPTS instructions which increase performance of data transfers to and from the chip registers without using PCI cycles.
- Has SCRIPTS support for 64-bit addressing.
- Supports multithreaded I/O algorithms in SCSI SCRIPTS with fast I/O context switching.
- Supports additional arithmetic capability with the Expanded Register Move instruction.

1.5.2 PCI Performance

To improve PCI performance, the LSI53C895A:

- Complies with PCI 2.2 specification.
- Supports 32-bit 33 MHz PCI interface with 64-bit addressing.
- Supports dual address cycles which can be generated for all SCRIPTS for > 4 Gbyte addressability.
- Bursts 2, 4, 8, 16, 32, 64, or 128 Dword transfers across the PCI bus.
- Supports 32-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 faster than 110 Mbytes/s (@ 33 MHz).
- Supports PCI Cache Line Size register.
- Supports PCI Write and Invalidate, Read Line, and Read Multiple commands.
- Complies with PCI Bus Power Management Specification Revision 1.1.

1.5.3 Integration

Features of the LSI53C895A which ease integration include:

- High-performance SCSI core.
- Integrated LVD transceivers.
- Full 32-bit PCI DMA bus master.
- Integrated SCRIPTS processor.
- Memory-to-Memory Move instructions allow use as a third-party PCI bus DMA controller.

1.5.4 Ease of Use

The LSI53C895A provides:

- Easy, drop-in replacement for the LSI53C895.
- Up to one megabyte of add-in memory support for BIOS and SCRIPTS storage.
- Reduced SCSI development effort.
- Compiler-compatible with existing LSI53C7XX and LSI53C8XX family SCRIPTS.
- Direct connection to PCI and SCSI SE, LVD and HVD (needs external transceivers).
- Development tools and sample SCSI SCRIPTS available.
- Nine GPIO pins.
- Maskable and pollable interrupts.
- Wide SCSI, A or P cable, and up to 15 devices supported.
- Three programmable SCSI timers: Select/Reselect, Handshake-to-Handshake, and General Purpose. The time-out period is programmable from 100 µs to greater than 25.6 seconds.
- Software for PC-based operating system support.
- Support for relative jumps.
- SCSI Selected as ID bits for responding with multiple IDs.

1.5.5 Flexibility

The LSI53C895A provides:

- Universal LVD transceivers are backward compatible with SE or HVD devices.
- High level programming interface (SCSI SCRIPTS).
- Ability to program local and bus flash memory.
- Selectable 112 or 944 byte DMA FIFO for backward compatibility.
- 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 Ultra2 SCSI transfer rates with a 40 MHz SCSI clock input.
- Selectable IRQ pin disable bit.
- Ability to route system clock to SCSI clock.
- Compatible with 3.3 V and 5 V PCI.

1.5.6 Reliability

Enhanced reliability features of the LSI53C895A include:

- 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).
- High proportion (> 25%) of device pins are power or ground.

- Power and ground isolation of I/O pads and internal chip logic.
- TolerANT technology, which 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.

1.5.7 Testability

The LSI53C895A provides improved testability through:

- Access to all SCSI signals through programmed I/O.
- SCSI loopback diagnostics.
- SCSI bus signal continuity checking.
- Support for single step mode operation.
- JTAG boundary scan.

Chapter 2
Functional Description

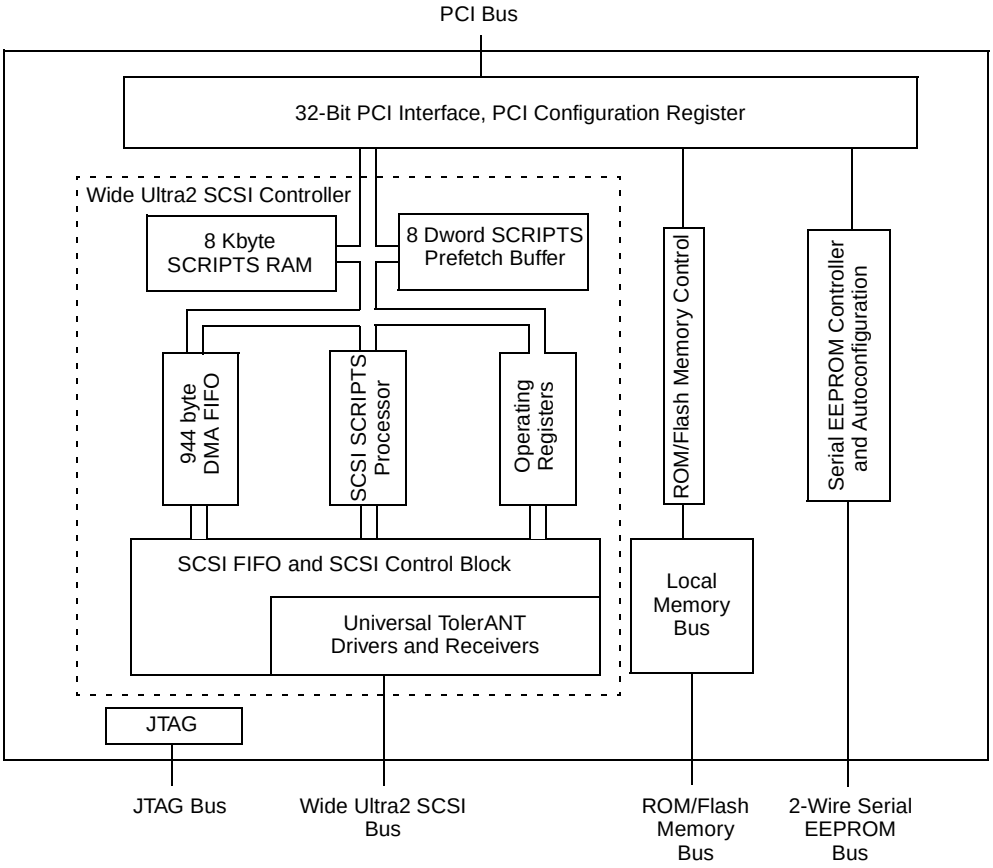
Chapter 2 is divided into the following sections:

The LSI53C895A PCI to Ultra2 SCSI Controller is composed of the following modules:

- 32-bit PCI Interface with 64-bit addressing
- PCI-to-Wide Ultra2 SCSI Controller
- ROM/Flash Memory Controller
- Serial EEPROM Controller

Figure 2.1 illustrates the relationship between these modules.

Figure 2.1 LSI53C895A Block Diagram



2.1 PCI Functional Description

The LSI53C895A implements a PCI-to-Wide Ultra2 SCSI controller.

2.1.1 PCI Addressing

There are three physical PCI-defined address spaces:

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

2.1.1.1 Configuration Space

The host processor uses the PCI configuration space to initialize the LSI53C895A through a defined set of configuration space registers. The Configuration registers are accessible only by system BIOS during PCI configuration cycles. The 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 eight lower order address bits, AD[7:0], select a specific 8-bit register. AD[10:8] are decoded as well, but they must be zero or the LSI53C895A does not respond. According to the PCI specification, AD[10:8] are reserved for multifunction devices.

At initialization time, each PCI device is assigned a base address for I/O and memory accesses. In the case of the LSI53C895A, the upper 24 bits of the address are selected. On every access, the LSI53C895A 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 for the LSI53C895A and the low-order eight bits define the register being accessed. A decode of C_BE[3:0]/ determines which registers and what type of access is to be 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 LSI53C895A. Base Address Register Zero (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, including the LSI53C895A. Base Address Register One (MEMORY) determines which 1 Kbyte memory area this device occupies. Base Address Register Two (SCRIPTS RAM) determines the 8 Kbyte memory area occupied by SCRIPTS RAM.

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 for the LSI53C895A

C_BE[3:0]/	Command Type	Supported as Master	Supported as Slave
0b0000	Interrupt Acknowledge	No	No
0b0001	Special Cycle	No	No
0b0010	I/O Read	Yes	Yes
0b0011	I/O Write	Yes	Yes
0b0100	Reserved	n/a	n/a
0b0101	Reserved	n/a	n/a
0b0110	Memory Read	Yes	Yes
0b0111	Memory Write	Yes	Yes
0b1000	Reserved	n/a	n/a
0b1001	Reserved	n/a	n/a
0b1010	Configuration Read	No	Yes
0b1011	Configuration Write	No	Yes
0b1100	Memory Read Multiple	Yes[1]	Yes (defaults to 0b0110)
0b1101	Dual Address Cycle (DAC)	Yes	No
0b1110	Memory Read Line	Yes[1]	Yes (defaults to 0b0110)
0b1111	Memory Write and Invalidate	Yes[2]	Yes (defaults to 0b0111)

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

2.1.2.1 Interrupt Acknowledge Command

The LSI53C895A 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 LSI53C895A 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 I/O Read command reads data from an agent mapped in I/O address space. All 32 address bits are decoded.

2.1.2.4 I/O Write Command

The I/O Write command writes data to an agent mapped in I/O address space. All 32 address bits are decoded.

2.1.2.5 Reserved Command

The LSI53C895A does not respond to this command as a slave and it never generates this command as a master.

2.1.2.6 Memory Read Command

The Memory Read command reads data from an agent mapped in the Memory Address Space. The target is free to do an anticipatory read for this command only if it can guarantee that such a read has no side effects.

2.1.2.7 Memory Write Command

The Memory Write command writes data to an agent mapped in the Memory Address Space. When the target returns “ready,” it assumes responsibility for the coherency (which includes ordering) of the subject data.

2.1.2.8 Configuration Read Command

The Configuration Read command reads 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.

2.1.2.9 Configuration Write Command

The Configuration Write command transfers data to the configuration space of each agent. An agent is selected when its IDSEL signal is asserted and AD[1:0] are 0b00.

2.1.2.10 Memory Read Multiple Command

This command is identical to the Memory Read command except that it additionally indicates that the master may intend to fetch more than one cache line before disconnecting. The LSI53C895A 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 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) and the ERMP bit (Enable Read Multiple, bit 2, DMA Mode (DMODE) register) are set.
- The Cache Line Size register for each function contains a legal burst size value (2, 4, 8, 16, 32, or 64) and that value is less than or equal to the DMODE burst size.
- The transfer will cross 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 in a single bus ownership. The number of cache lines to read is 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 amount of data to transfer, with the maximum allowable burst size determined from the DMA Mode (DMODE) burst size bits, and the Chip Test Five (CTEST5), bit 2.

2.1.2.11 Dual Address Cycle (DAC) Command

The LSI53C895A performs DACs when 64-bit addressing is required. Refer to the PCI 2.2 specification. If any of the selector registers contain a nonzero value, a DAC is generated. See 64-bit SCRIPTS Selectors in Chapter 4, “Registers,” for additional information.

2.1.2.12 Memory Read Line Command

This command is identical to the Memory Read command, except that 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 LSI53C895A takes advantage of the PCI 2.2 specification regarding issuing this command.

If the cache mode is disabled, Read Line commands are not 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 (Cache Line Size Enable, bit 7, DMA Control (DCNTL) register) and ERL (Enable Read Line, bit 3, DMA Mode (DMODE) register) bits are set.
- The Cache Line Size register must contain a legal burst size value in Dwords (2, 4, 8, 16, 32, 64, or 128) and that value is less than or equal to the DMA Mode (DMODE) burst size.
- The transfer will cross 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, even though the conditions for Read Line are met.

If the Read Multiple mode is enabled and the Read Line mode is disabled, 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 that it additionally guarantees a minimum transfer of one complete cache line; that is to say, 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 register at address 0x0C in PCI configuration space. The LSI53C895A 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:

1. The CLSE bit (Cache Line Size Enable, bit 7, DMA Control (DCNTL) register), WRIE bit (Write and Invalidate Enable, bit 0, Chip Test Three (CTEST3) register), and PCI configuration Command register, bit 4 are set.
2. The Cache Line Size register contains a legal burst size value in Dwords (2, 4, 8, 16, 32, 64, or 128) and that value is less than or equal to the DMA Mode (DMODE) burst size.
3. The chip has enough bytes in the DMA FIFO to complete at least one full cache line burst.
4. The chip is aligned to a cache line boundary.

When these conditions are met, the LSI53C895A issues a Memory 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 size of the transfer 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 amount of data to transfer, with the maximum allowable burst size determined from the DMA Mode

(DMODE) burst size bits, and Chip Test Five (CTEST5), bit 2. If multiple cache line size transfers are not desired, set the DMA Mode (DMODE) burst size to exactly the cache line size and the chip only issues single cache line transfers.

After each data transfer, the chip re-evaluates the burst size based on the amount of remaining data to transfer and again selects the highest possible multiple of the cache line size, and no larger than the DMA Mode (DMODE) burst size. The most likely scenario of this scheme is that the chip selects the DMA Mode (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 burst size is left. 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 such that when a latency time-out occurs, the LSI53C895A 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 – During a Memory Write and Invalidate transfer, if the target device issues a retry (STOP with no TRDY/, indicating that no data was transferred), 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 – During a Memory Write and Invalidate transfer, if the target device issues a disconnect the LSI53C895A 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 PCI Cache Mode

The LSI53C895A supports the PCI specification for an 8-bit Cache Line Size register located in the PCI configuration space. The Cache Line Size register provides the ability to sense and react to nonaligned

addresses corresponding to cache line boundaries. In conjunction with the Cache Line Size register, the PCI commands Memory Read Line, Memory Read Multiple, Memory Write and Invalidate are each software enabled or disabled to allow the user full flexibility in using these commands.

2.1.3.1 Enabling Cache Mode

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

- The Cache Line Size Enable bit in DMA Control (DCNTL) register must be set.
- The PCI Cache Line Size register must contain a valid binary cache size, i.e. 2, 4, 8, 16, 32, 64, or 128 Dwords. Only these values are considered 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 Chip Test Five (CTEST5) bit 2 are the burst length bits.
- The part must be doing a PCI Master transfer. The following PCI Master transactions do not utilize the PCI cache logic and thus no PCI cache command is issued during these types of cycles: a nonprefetch SCRIPTS fetch, a Load/Store data transfer, a data flush operation. All other types of PCI Master transactions will utilize the PCI cache logic.

The above four conditions must be met for the cache logic to control the type of PCI cache command that is issued, along with any alignment that may be 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.

2.1.3.2 Issuing Cache Commands

In order to issue each type of PCI cache command, the corresponding enable bit must be set (2 bits in the case of Memory Write and Invalidate). These bits are detailed below:

- To issue Memory Read Line commands, the Read Line enable bit in the DMA Mode (DMODE) register must be set.
- To issue Memory Read Multiple commands, the Read Multiple enable bit in the DMA Mode (DMODE) register must be set.
- To issue Memory Write and Invalidate commands, both the Write and Invalidate enables in the Chip Test Three (CTEST3) register and the PCI configuration command register must be set.

If the corresponding cache command being issued is not enabled then the cache logic falls back to the next command enabled. Specifically, if Memory Read Multiple is not enabled and Memory Read Lines are, read lines are issued in place of read multiple. If no cache commands are enabled, cache write alignment still occurs but no cache commands are issued, only memory reads and memory writes.

2.1.3.3 Memory Read Caching

The type of Memory Read command issued depends on the starting location of the transfer and the number of bytes being transferred. During reads, no cache alignment is done (this is not required nor optimal per PCI 2.2 specification) and reads will always be either a programmed burst length in size, as set in the DMA Mode (DMODE) and Chip Test Three (CTEST3) registers. In the case of a transfer which is smaller than the burst length, all bytes for that transfer are read in one PCI burst transaction. If the transfer will cross a Dword boundary (A[1:0] = 0b00) a Memory Read Line command is issued. When the transfer will cross a cache boundary (depends on cache line size programmed into the PCI configuration register), a Memory Read Multiple command is issued. If a transfer will not cross a Dword or cache boundary or if cache mode is not enabled a Memory Read command is issued.

2.1.3.4 Memory Write Caching

Writes are aligned in a single burst transfer to get to 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 are issued as long as the remaining byte count is greater than the Memory Write and Invalidate threshold. When the byte count goes below this threshold, a single Memory Write burst is issued to complete the transfer. 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 describes PCI cache mode alignment.

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

2.1.3.5 Examples:

The examples in this section employ the following abbreviations: 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: MRL (15 bytes)
MR (2 bytes)

C to D: MRM (5 bytes)

C to E: MRM (15 bytes)
MRM (6 bytes)

D to F: MRL (15 bytes)
MRL (16 bytes)
MR (1 byte)

A to H: MRL (15 bytes)
MRL (16 bytes)
MRL (16 bytes)
MRL (16 bytes)
MRL (16 bytes)
MR (2 bytes)

A to G: MRL (15 bytes)
MRL (16 bytes)
MRL (16 bytes)
MRL (16 bytes)
MR (3 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 (31 bytes)
MR (1 byte)

A to H: MRM (31 bytes)
MRM (32 bytes)
MRM (18 bytes)

A to G: MRM (31 bytes)
MRM (32 bytes)
MR (3 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 (63 bytes)
MRL (16 bytes)
MRM (2 bytes)

A to G: 2 transfers, MRM (63 bytes), MR (3 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 (15 bytes)
MWI (64 bytes)
MW (2 bytes)

A to G: MW (15 bytes)
MWI (32 bytes)
MW (18 bytes)

2.1.3.6 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 (that is, Source address [2:0] == Destination Address [2:0]), write aligning is not performed and Memory Write and Invalidate commands are not issued. The LSI53C895A is little endian only.

2.2 SCSI Functional Description

The LSI53C895A provides an Ultra2 SCSI controller that supports an 8-bit or 16-bit bus. The controller supports Wide Ultra2 SCSI synchronous transfer rates up to 80 Mbytes/s on a LVD SCSI bus. The SCSI core can be programmed with SCSI SCRIPTS, making it easy to “fine tune” the system for specific mass storage devices or Ultra2 SCSI requirements.

The LSI53C895A offers low level register access or a high-level control interface. Like first generation SCSI devices, the LSI53C895A is accessed as a register-oriented device. Error recovery and/or diagnostic procedures use the ability to sample and/or assert any signal on the SCSI bus. In support of SCSI loopback diagnostics, the SCSI core may perform a self-selection and operate as both an initiator and a target.

The LSI53C895A is controlled by the integrated SCRIPTS processor through a high-level logical interface. Commands controlling the SCSI core are fetched out of the main host memory or local memory. These commands instruct the SCSI core to Select, Reselect, Disconnect, Wait for a Disconnect, Transfer Information, Change Bus Phases and, in general, implement all aspects of the SCSI protocol. The SCRIPTS processor is a special high-speed processor optimized for SCSI protocol.

2.2.1 SCRIPTS Processor

The SCSI SCRIPTS processor allows both DMA and SCSI commands to be fetched from host memory or internal SCRIPTS RAM. Algorithms written in SCSI SCRIPTS control the actions of the SCSI and DMA cores. The SCRIPTS processor executes complex SCSI bus sequences independently of the host CPU.

Algorithms may be designed to tune SCSI bus performance, to adjust to new bus device types (such as scanners, communication gateways, etc.), or to incorporate changes in the SCSI-2 or SCSI-3 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 like Phase Mismatch.

2.2.1.1 Phase Mismatch Handling in SCRIPTS

The LSI53C895A 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.

Storing the appropriate information to later restart the I/O can be done through SCRIPTS, 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 powers up disabled and must be enabled by setting the Phase Mismatch Jump Enable bit (ENPMJ, bit 7 in the Chip Control 0 (CCNTL0) register).

Utilizing the information supplied in the Phase Mismatch Jump Address 1 (PMJAD1) and Phase Mismatch Jump Address 2 (PMJAD2) registers, described in Chapter 4, "Registers," SCRIPTS handles all overhead involved in a disconnect/reselect sequence with a modest number of instructions.

2.2.2 Internal SCRIPTS RAM

The LSI53C895A has 8 Kbyte (2048 x 32 bits) of internal, general purpose RAM. 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. Other types of access to the RAM by the chip, except Load/Store, use the PCI bus, as if they were external accesses. The SCRIPTS RAM powers up enabled by default.

The RAM can be relocated by the PCI system BIOS anywhere in the 32-bit address space. The Base Address Register Two (SCRIPTS RAM) in the PCI configuration space contains the base address of the internal RAM. To simplify loading of the SCRIPTS instructions, 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 RAM is byte accessible from the PCI bus and is visible to any bus mastering device on the bus. External accesses to the RAM (by the CPU) follow the same timing sequence as a standard slave register access, except that the required target wait-states drop from 5 to 3.

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 LSI53C895A, see Chapter 5, "SCSI SCRIPTS Instruction Set."

2.2.3 64-Bit Addressing in SCRIPTS

The LSI53C895A has a 32-bit PCI interface which provides 64-bit address capability in the initiator 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/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 default to zero, meaning the LSI53C895A powers-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 and Store instructions, and jumps are all instructions with 64-bit address capability.

Crossing the 4 Gbyte boundary on any one SCRIPTS operation is not permitted and software needs to take care that any given SCRIPTS operation will not cross the 4 Gbyte boundary.

2.2.4 Hardware Control of SCSI Activity LED

The LSI53C895A has the ability to control a LED through the GPIO_0 pin to indicate that it is connected to the SCSI bus. Formerly this function was done by a software driver.

When bit 5 (LED_CNTL) in the General Purpose Pin Control Zero (GPCNTL0) register is set and bit 6 (Fetch Enable) in the General Purpose Pin Control Zero (GPCNTL0) register is cleared and the LSI53C895A is not performing an EEPROM autodownload, then bit 3 (CON) in the Interrupt Status Zero (ISTAT0) register is presented at the GPIO_0 pin.

The CON (Connected) bit in Interrupt Status Zero (ISTAT0) is set anytime the LSI53C895A is connected to the SCSI bus either as an initiator or a target. This will happen after the LSI53C895A has successfully completed a selection or when it has successfully responded to a selection or reselection. It will also be set when the LSI53C895A wins arbitration in low level mode.

2.2.5 Designing an Ultra2 SCSI System

Since Ultra2 SCSI is based on existing SCSI standards, it can use existing driver programs as long as the software is able to negotiate for Ultra2 SCSI synchronous transfer rates. Additional software modifications are needed to take advantage of the new features in the LSI53C895A.

In the area of hardware, LVD SCSI is required to achieve Ultra2 SCSI transfer rates and to support the longer cable and additional devices on the bus. All devices on the bus must have LVD SCSI capabilities to guarantee Ultra2 SCSI transfer rates. For additional information on Ultra2 SCSI, refer to the SPI-2 working document which is available from the SCSI BBS referenced at the beginning of this manual.
Chapter 6, "Electrical Specifications," contains Ultra2 SCSI timing information. In addition to the guidelines in the draft standard, make the following software and hardware adjustments to accommodate Ultra2 SCSI transfers:

- Set the Ultra Enable bit to enable Ultra2 SCSI transfers.
- Set the TolerANT Enable bit, bit 7 in the SCSI Test Three (STEST3) register, whenever the Ultra Enable bit is set.
- Do not extend the SREQ/SACK filtering period with SCSI Test Two (STEST2) bit 1. When the Ultra Enable bit is set, the filtering period is fixed at 8 ns for Ultra2 SCSI or 15 ns for Ultra SCSI, regardless of the value of the SREQ/SACK Filtering bit.
- Use the SCSI clock quadrupler.

  A 40 MHz input must be supplied if using the SCSI clock quadrupler for an Ultra2 design.

2.2.5.1 Using the SCSI Clock Quadrupler

The LSI53C895A can quadruple the frequency of a 40 MHz SCSI clock, allowing the system to perform Ultra2 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:

Step 1. Set the SCLK Quadrupler Enable bit (SCSI Test One (STEST1), bit 3).

Step 2. Poll bit 5 of the SCSI Test Four (STEST4) register. The LSI53C895A sets this bit as soon as it locks in the 160 MHz frequency. The frequency lockup takes approximately 100 microseconds.

Step 3. Halt the SCSI clock by setting the Halt SCSI Clock bit (SCSI Test Three (STEST3), bit 5).

Step 4. Set the clock conversion factor using the SCF and CCF fields in the SCSI Control Three (SCNTL3) register.

Step 5. Set the SCLK Quadrupler Select bit (SCSI Test One (STEST1), bit 2).

Step 6. Clear the Halt SCSI Clock bit.

2.2.6 Prefetching SCRIPTS Instructions

When enabled by setting the Prefetch Enable bit (bit 5) in the DMA Control (DCNTL) register, the prefetch logic in the LSI53C895A fetches 8 Dwords of instructions. The prefetch logic automatically determines the maximum burst size that it can perform, based on the burst length as determined by the 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 will use PCI cache commands Memory Read Line, and Memory Read Multiple, if PCI caching is enabled.

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

The LSI53C895A may flush the contents of the prefetch unit under certain conditions, listed below, to ensure that the chip always operates from the most current version of the SCRIPTS instruction. When one of these conditions apply, the contents of the prefetch unit are automatically flushed.

- On every Memory Move instruction. The Memory Move instruction is often used to place modified code directly into memory. To make sure that the chip executes all recent modifications, the prefetch unit

flushes its contents and loads the modified code every 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 on this instruction 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 because the next instruction to execute 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. The 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 LSI53C895A 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 chip uses two accesses to obtain the four Dwords required, in two bursts of two Dwords each.

Note: This feature is only useful if Prefetching is disabled and SCRIPTS instructions are fetched from main memory. Due to the short SCRIPTS RAM access time, burst opcode fetching is not necessary when fetching instructions from this memory.

2.2.8 Load and Store Instructions

The LSI53C895A supports the Load and Store instruction type, which simplifies the movement of data between memory and the internal chip 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 will remain internal to the chip and will 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. This feature can be disabled by setting the DILS bit in the Chip Control 0 (CCNTL0) register. For more information on the Load and Store instructions, refer to Chapter 5, "SCSI SCRIPTS Instruction Set."

2.2.9 JTAG Boundary Scan Testing

The LSI53C895A includes support for JTAG boundary scan testing in accordance with the IEEE 1149.1 specification with one exception, which is explained in this section. This device accepts all required boundary scan instructions including the optional CLAMP, HIGH-Z, and IDCODE instructions.

The LSI53C895A 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 10 MHz TCK frequency for TDO and TDI.

Due to design constraints, the RST/ pin (system reset) always 3-states the SCSI pins when it is asserted. Boundary scan logic does not control this action, and this is not compliant with the specification. There are two solutions that resolve this issue:

1. Use the RST/ pin as a boundary scan compliance pin. When the pin is deasserted, the device is boundary scan compliant and when asserted, the device is noncompliant. To maintain compliance the RST/ pin must be driven HIGH.
2. When RST/ is asserted during boundary scan testing the expected output on the SCSI pins must be the HIGH-Z condition, and not what is contained in the boundary scan data registers for the SCSI pin output cells.

2.2.10 SCSI Loopback Mode

The LSI53C895A loopback mode allows testing of both initiator and target functions and, in effect, lets the chip communicate with itself. When the Loopback Enable bit is set in the SCSI Test Two (STEST2) register, bit 4, the LSI53C895A allows control of all SCSI signals whether the chip is operating in the initiator or target mode. For more information on this mode of operation refer to the LSI Logic *SCSI SCRIPTS Processors Programming Guide*.

2.2.11 Parity Options

The LSI53C895A implements a flexible parity scheme that allows control of the parity sense, allows parity checking to be turned on or off, and has the ability to deliberately send a byte with bad parity over the SCSI bus to test parity error recovery procedures. Table 2.3 defines the bits that are involved in parity control and observation. Table 2.4 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. Table 2.5 describes the options available when a parity error occurs. Figure 2.2 shows where parity checking is done in the LSI53C895A.

Table 2.3 Bits Used for Parity Control and Generation

Bit Name	Location	Description
Assert SATN/ on Parity Errors	SCSI Control Zero (SCNTL0), Bit 1	Causes the LSI53C895A to automatically assert SATN/ when it detects a SCSI parity error while operating as an initiator.
Enable Parity Checking	SCSI Control Zero (SCNTL0), Bit 3	Enables the LSI53C895A to check for parity errors. The LSI53C895A checks for odd parity.
Assert Even SCSI Parity	SCSI Control One (SCNTL1), Bit 2	Determines the SCSI parity sense generated by the LSI53C895A to the SCSI bus.
Disable Halt on SATN/ or a Parity Error (Target Mode Only)	SCSI Control One (SCNTL1), Bit 5	Causes the LSI53C895A not to halt operations when a parity error is detected in target mode.
Enable Parity Error Interrupt	SCSI Interrupt Enable Zero (SIEN0), Bit 0	Determines whether the LSI53C895A generates an interrupt when it detects a SCSI parity error.
Parity Error	SCSI Interrupt Status Zero (SIST0), Bit 0	This status bit is set whenever the LSI53C895A detects a parity 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	SSTAT 2, Bit 3 and 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	Enables parity checking during PCI master data phases.
Master Data Parity Error	DMA Status (DSTAT), Bit 6	Set when the LSI53C895A, 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.

Table 2.4 SCSI Parity Control

EPC[1]	ASEP[2]	Description
0	0	Does not check for parity errors. Parity is generated when sending SCSI data. Asserts odd parity when sending SCSI data.
0	1	Does not check for parity errors. Parity is generated when sending SCSI data. Asserts even parity when sending SCSI data.
1	0	Checks for odd parity on SCSI data received. Parity is generated when sending SCSI data. Asserts odd parity when sending SCSI data.
1	1	Checks for odd parity on SCSI data received. Parity is generated when sending SCSI data. Asserts even parity when sending SCSI data.

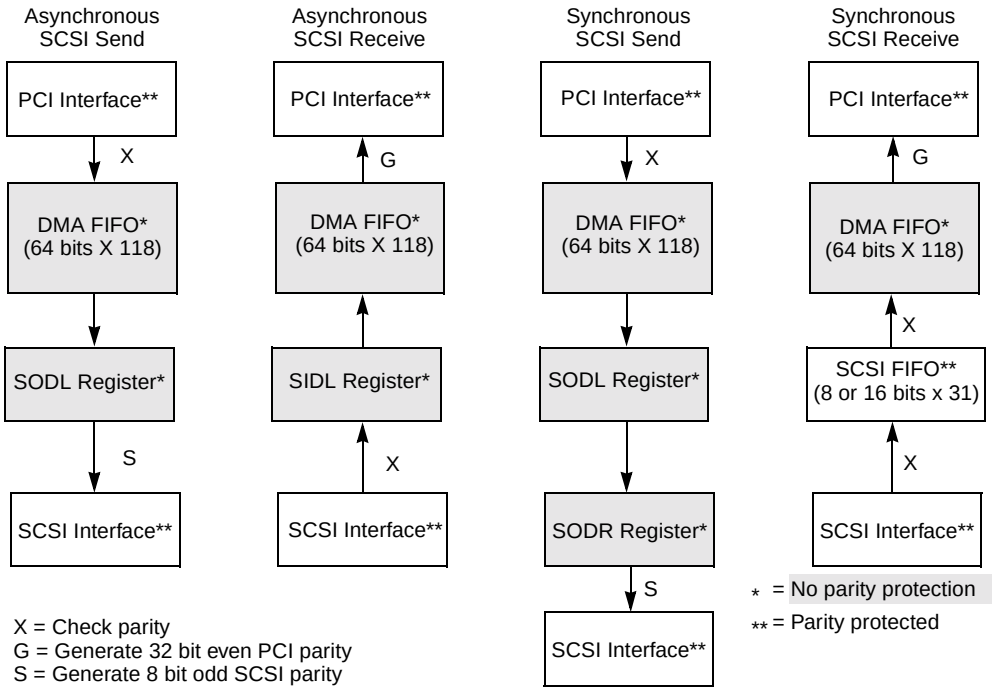
1. EPC = Enable Parity Checking (bit 3 SCSI Control Zero (SCNTL0)).
2. ASEP = Assert SCSI Even Parity (bit 2 SCSI Control One (SCNTL1)).

Table 2.5 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 Zero (SIEN0)).

Figure 2.2 Parity Checking/Generation



2.2.12 DMA FIFO

The DMA FIFO is 8 bytes wide by 118 transfers deep. The DMA FIFO is illustrated in Figure 2.3. The default DMA FIFO size is 112 bytes to assure compatibility with older products in the LSI53C8XX family.

The DMA FIFO size may be set to 944 bytes by setting the DMA FIFO Size bit, bit 5, in the Chip Test Five (CTEST5) register.

Figure 2.3 DMA FIFO Sections

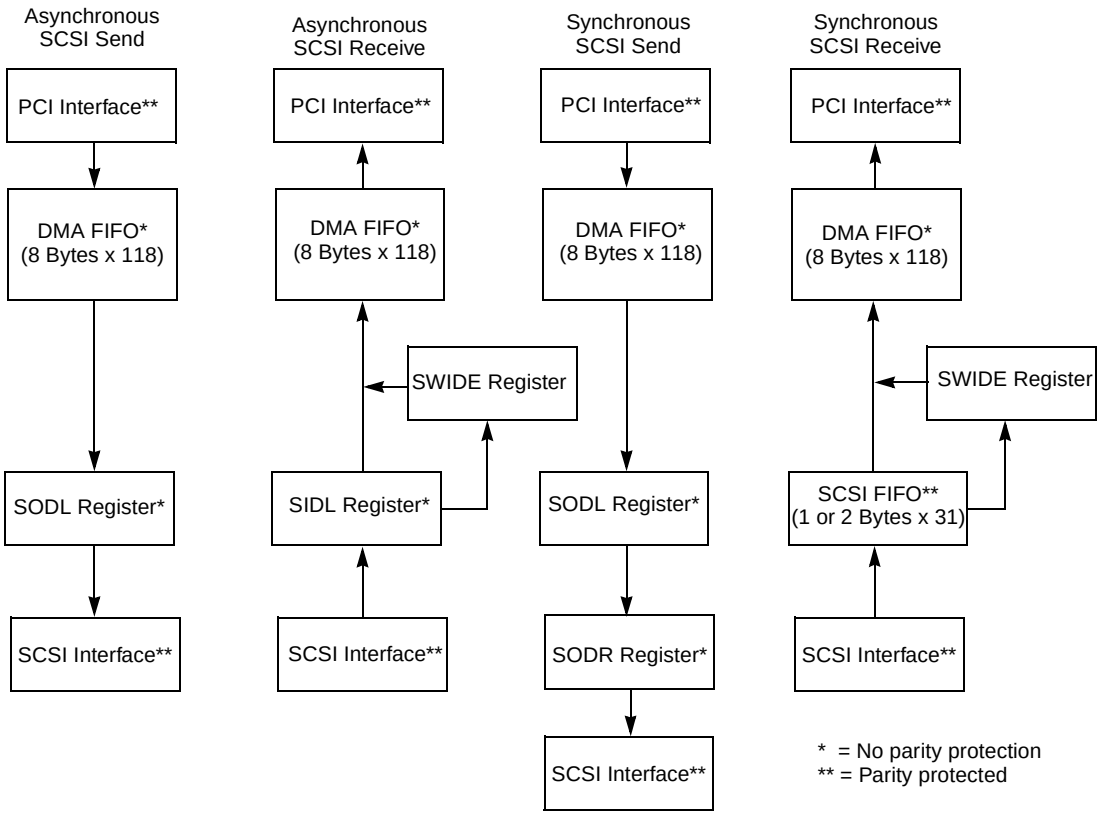
The LSI53C895A automatically supports misaligned DMA transfers. A 944-byte FIFO allows the LSI53C895A to support 2, 4, 8, 16, 32, 64, or 128 Dword bursts across the PCI bus interface.

2.2.12.1 Data Paths

The data path through the LSI53C895A is dependent on whether data is being moved into or out of the chip, and whether SCSI data is being transferred asynchronously or synchronously.

Figure 2.4 shows how data is moved to/from the SCSI bus in each of the different modes.

Figure 2.4 LSI53C895A Host Interface SCSI Data Paths



The following steps determine if any bytes remain in the data path when the chip halts an operation:

Asynchronous SCSI Send –

Step 1. If the DMA FIFO size is set to 112 bytes (bit 5 of the Chip Test Five (CTEST5) register cleared), look at the DMA FIFO (DFIFO) and DMA Byte Counter (DBC) registers and calculate if there are bytes left in the DMA FIFO. To make this calculation, subtract the seven least significant bits of the DBC register from the 7-bit value of the DFIFO register. AND the result with 0x7F for a byte count between zero and 112.

If the DMA FIFO size is set to 944 bytes (bit 5 of the Chip Test Five (CTEST5) register is set), subtract the 10 least significant bits of the DBC register from the 10-bit value of the DMA FIFO

Byte Offset Counter, which consists of bits [1:0] in the CTEST5 register and bits [7:0] of the DMA FIFO register. AND the result with 0x3FF for a byte count between zero and 944.

Step 2. Read bit 5 in the SCSI Status Zero (SSTAT0) and SCSI Status Two (SSTAT2) registers to determine if any bytes are left in the SCSI Output Data Latch (SODL) register. If bit 5 is set in the SSTAT0 or SSTAT2 register, then the least significant byte or the most significant byte in the SODL register is full, respectively. Checking this bit also reveals bytes left in the SODL register from a Chained Move operation with an odd byte count.

Synchronous SCSI Send –

Step 1. If the DMA FIFO size is set to 112 bytes (bit 5 of the Chip Test Five (CTEST5) register cleared), look at the DFIFO and DBC registers and calculate if there are bytes left in the DMA FIFO. To make this calculation, subtract the seven least significant bits of the DMA Byte Counter (DBC) register from the 7-bit value of the DMA FIFO (DFIFO) register. AND the result with 0x7F for a byte count between zero and 112.

If the DMA FIFO size is set to 944 bytes (bit 5 of the CTEST5 register is set), subtract the 10 least significant bits of the DBC register from the 10-bit value of the DMA FIFO Byte Offset Counter, which consists of bits [1:0] in the CTEST5 register and bits [7:0] of the DMA FIFO register. AND the result with 0x3FF for a byte count between zero and 944.

Step 2. Read bit 5 in the SCSI Status Zero (SSTAT0) and SCSI Status Two (SSTAT2) registers to determine if any bytes are left in the SCSI Output Data Latch (SODL) register. If bit 5 is set in the SSTAT0 or SSTAT2 register, then the least significant byte or the most significant byte in the SODL register is full, respectively. Checking this bit also reveals bytes left in the SODL register from a Chained Move operation with an odd byte count.

Step 3. Read bit 6 in the SCSI Status Zero (SSTAT0) and SCSI Status Two (SSTAT2) registers to determine if any bytes are left in the SODR register (a hidden buffer register which is not

accessible). If bit 6 is set in the SSTAT0 or SSTAT2 register, then the least significant byte or the most significant byte in the SODR register is full, respectively.

Asynchronous SCSI Receive –

Step 1. If the DMA FIFO size is set to 112 bytes (bit 5 of the Chip Test Five (CTEST5) register cleared), look at the DMA FIFO (DFIFO) and DMA Byte Counter (DBC) registers and calculate if there are bytes left in the DMA FIFO. To make this calculation, subtract the seven least significant bits of the DBC register from the 7-bit value of the DFIFO register. AND the result with 0x7F for a byte count between zero and 88.

If the DMA FIFO size is set to 944 bytes (bit 5 of the Chip Test Five (CTEST5) register is set), subtract the 10 least significant bits of the DMA Byte Counter (DBC) register from the 10-bit value of the DMA FIFO Byte Offset Counter, which consists of bits [1:0] in the CTEST5 register and bits [7:0] of the DMA FIFO register. AND the result with 0x3FF for a byte count between zero and 944.

Step 2. Read bit 7 in the SCSI Status Zero (SSTAT0) and SCSI Status Two (SSTAT2) registers to determine if any bytes are left in the SCSI Input Data Latch (SIDL) register. If bit 7 is set in the SSTAT0 or SSTAT2 register, then the least significant byte or the most significant byte is full, respectively.

Step 3. If any wide transfers have been performed using the Chained Move instruction, read the Wide SCSI Receive bit (SCSI Status Two (SSTAT2), bit 0) to determine whether a byte is left in the SCSI Wide Residue (SWIDE) register.

Synchronous SCSI Receive –

Step 1. If the DMA FIFO size is set to 112 bytes, subtract the seven least significant bits of the DMA Byte Counter (DBC) register from the 7-bit value of the DMA FIFO (DFIFO) register. AND the result with 0x7F for a byte count between zero and 112.

If the DMA FIFO size is set to 944 bytes (bit 5 of the Chip Test Five (CTEST5) register is set), subtract the 10 least significant bits of the DBC register from the 10-bit value of the DMA FIFO Byte Offset Counter, which consists of bits [1:0] in the Chip Test

Five (CTEST5) register and bits [7:0] of the DMA FIFO register. AND the result with 0x3FF for a byte count between zero and 944.

Step 2. Read the SCSI Status One (SSTAT1) register and examine bits [7:4], the binary representation of the number of valid bytes in the SCSI FIFO, to determine if any bytes are left in the SCSI FIFO.

Step 3. If any wide transfers have been performed using the Chained Move instruction, read the Wide SCSI Receive bit (SCSI Control Two (SCNTL2), bit 0) to determine whether a byte is left in the SCSI Wide Residue (SWIDE) register.

2.2.13 SCSI Bus Interface

The LSI53C895A performs SE and LVD transfers, and supports traditional HVD operation when the chip is connected to external HVD transceivers.

To support LVD SCSI, all SCSI data and control signals have both negative and positive signal lines. The negative signals perform the SCSI data and control function. In the SE mode the positive signals become virtual ground drivers. In the HVD mode, the positive signals provide directional control to the external transceivers. TolerANT technology provides signal filtering at the inputs of SREQ/ and SACK/ to increase immunity to signal reflections.

2.2.13.1 LVDlink Technology

To support greater device connectivity and a longer SCSI cable, the LSI53C895A features LVDlink technology, the LSI Logic implementation of LVD SCSI. LVDlink transceivers provide the inherent reliability of differential SCSI, and a long-term migration path of faster SCSI transfer rates.

LVDlink technology is based on current drive. Its low output current reduces the power needed to drive the SCSI bus, so that the I/O drivers can be integrated directly onto the chip. This reduces the cost and complexity compared to traditional HVD designs. LVDlink lowers the amplitude of noise reflections and allows higher transmission frequencies.

The LSI Logic LVDlink transceivers operate in LVD or SE modes. They allow the chip to detect a HVD signal when the chip is connected to external HVD transceivers. The LSI53C895A automatically detects which type of signal is connected, based on the voltage detected by the DIFFSENS pin. Bits 7 and 6 of the SCSI Test Four (STEST4) register contain the encoded value for the type of signal that is detected (LVD, SE, or HVD). Please see the SCSI Test Four (STEST4) register description for encoding and other bit information.

2.2.13.2 HVD Mode

To maintain backward compatibility with legacy systems, the LSI53C895A can operate in the HVD mode (when the chip is connected to external differential transceivers). In the HVD mode, the SD[15:0]+, SDP[1:0]+, REQ+, ACK+, SRST+, SBSY+, and SSEL+ signals control the direction of external differential pair transceivers. The LSI53C895A is placed in the HVD mode by setting the DIF bit, bit 5, of the SCSI Test Two (STEST2) register (0x4E). Setting this bit 3-states the BSY–, SEL–, and RST– pads so they can be used as pure input pins. In addition to the standard SCSI lines, the signals shown in Table 2.6 are used by the LSI53C895A during HVD operation.

Table 2.6 HVD Signals

Signal	Function
BSY+, SEL+, RST+	Active HIGH signals used to enable the differential drivers as outputs for SCSI signals BSY–, SEL–, and RST–, respectively.
SD[15:0]+, SDP[1:0]+	Active HIGH signals used to control the direction of the differential drivers for SCSI data and parity lines, respectively.
ACK+	Active HIGH signal used to control the direction of the differential drivers for the initiator group signals ATN– and ACK–.
REQ+	Active HIGH signal used to control the direction of the differential drivers for target group signals MSG–, C_D–, I/O– and REQ–.
DIFFSENS	Input to the LSI53C895A used to detect the voltage level of a SCSI signal to determine whether it is a SE, LVD, or high-power differential signal. The encoded result is displayed in SCSI Test Four (STEST4) bits 7 and 6.

In the example differential wiring diagram in Figure 2.5, the LSI53C895A is connected to TI SN75976 differential transceivers for Ultra SCSI operation. The recommended value of the pull-up resistor on the REQ–,

ACK–, MSG–, C_D–, I/O–, ATN–, SD[7:0]–, and SDP0– lines is 680 Ω when the Active Negation portion of LSI Logic TolerANT technology is not enabled. When TolerANT is enabled, the recommended resistor value on the REQ–, ACK–, SD[7:0]–, and SDP0– signals is 1.5 kΩ. The electrical characteristics of these pins change when TolerANT is enabled, permitting a higher resistor value.

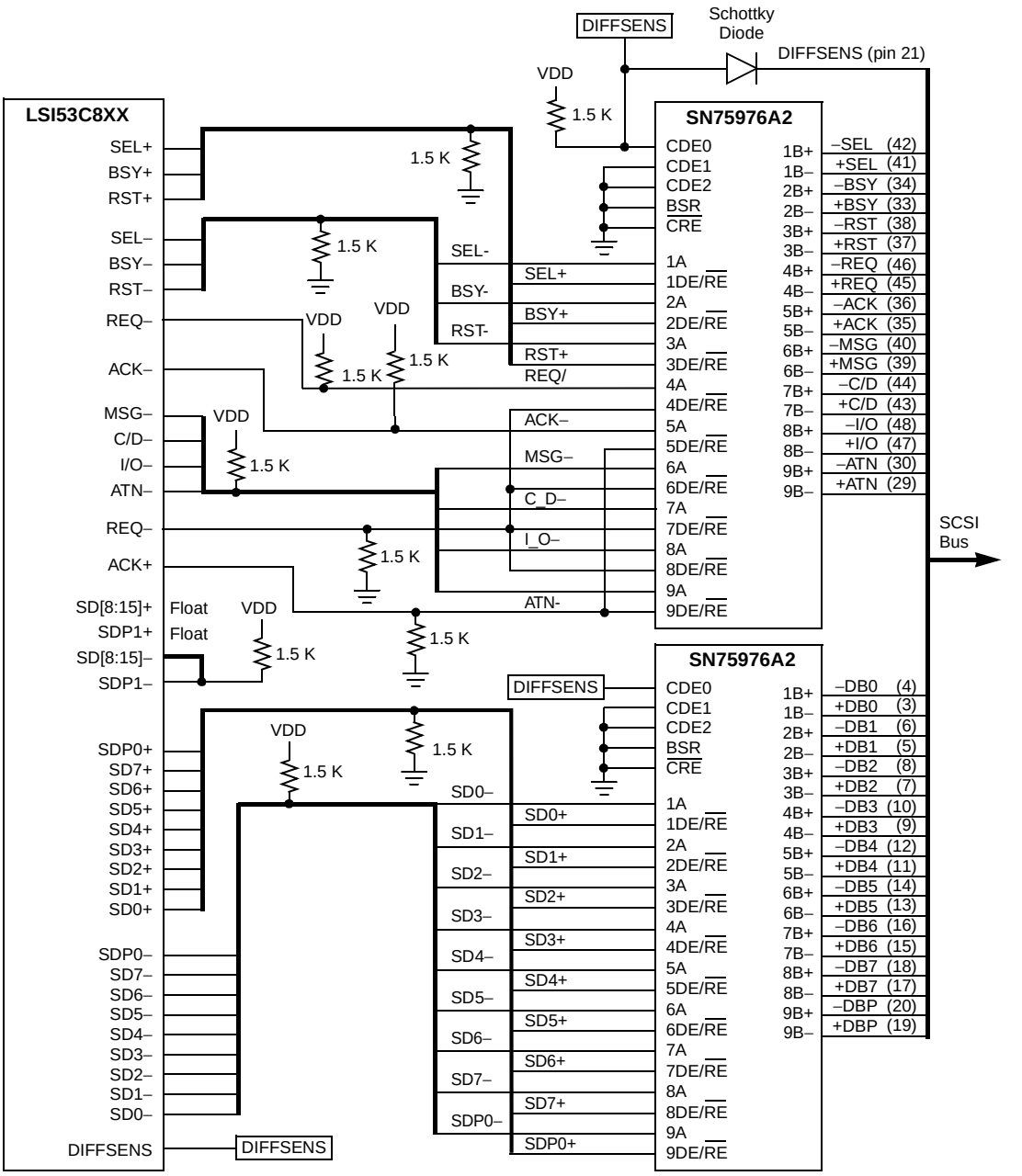
To interface the LSI53C895A to the SN75976A, connect the positive pins in the SCSI LVD pair of the LSI53C895A directly to the transceiver enables (xDE/RE/). These signals control the direction of the channels on the SN75976A.

The SCSI bidirectional control and data pins (SD[7:0]– SDP0–, SREQ–, SACK–, SMSG–, SI_O–, SC_D, and ATN–) of the LSI53C895A connect to the bidirectional data pins (nA) of the SN75976A with a pull-up resistor. The pull-up value should be no lower than the transceiver I_{OL} can tolerate, but not so high as to cause RC timing problems. The three remaining pins, SSEL–, SBSY– and SRST–, are connected to the SN75976A with a pull-down resistor. The pull-down resistors are required when the pins (nA) of the SN75976A are configured as inputs. When the data pins are inputs, the resistors provide a bias voltage to both the LSI53C895A pins (SSEL–, SBSY–, and SRST–) and the SN75976A data pins. Because the SSEL–, SBSY–, and SRST– pins on the LSI53C895A are inputs only, this configuration allows for the SSEL–, SBSY–, and SRST– SCSI signals to be asserted on the SCSI bus.

Note: The differential pairs on the SCSI bus are reversed when connected to the SN75976A, due to the active low nature of the SCSI bus.

8-Bit/16-Bit SCSI and the HVD Interface – In an 8-bit SCSI bus, the SD[15:8] pins on the LSI53C895A should be pulled up with a 1.5 kΩ resistor or terminated like the rest of the SCSI bus lines. This is very important, as errors may occur during reselection if these lines are left floating.

Figure 2.5 8-Bit HVD Wiring Diagram for Ultra2 SCSI



2.2.13.3 SCSI Termination

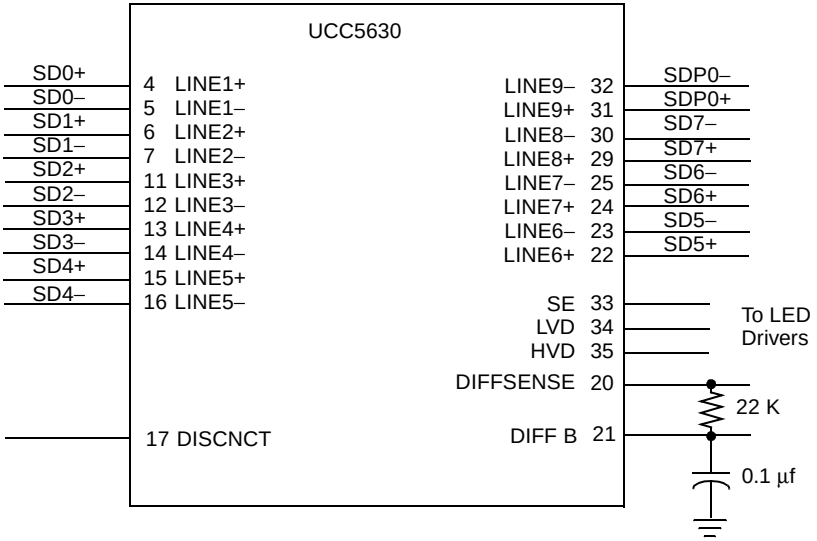
The terminator networks provide the biasing needed to pull signals to an inactive voltage level, and to match the impedance seen at the end of the cable with the characteristic impedance of the cable. Terminators must be installed at the extreme ends of the SCSI chain, and only at the ends. No system should ever have more or less than two terminators installed and active. 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 LSI53C895A, regulated (or active) termination is recommended. Figure 2.6 shows a Unitrode 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 LSI53C895A is to be used in a design that has only an 8-bit SCSI bus, all 16 data lines must still be terminated.

Figure 2.6 Regulated Termination for Ultra2 SCSI



DIFFSENSE connects to the SCSI bus Diffsense line to detect what type of devices (SE, LVD, or HVD) are connected to the SCSI bus. DISCNCT shuts down the terminator when it is not at the end of the bus. The disconnect pin low enables the terminator. Use additional UCC5630 terminators to terminate the SCSI control signals and wide SCSI data byte as needed.

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, the initiator SCRIPTS should start with a Set Initiator instruction or the target SCRIPTS should start with a Set Target instruction. The Selection and Reselection Enable bits (SCSI Chip ID (SCID) bits 5 and 6, respectively) should both be asserted so that the LSI53C895A may respond as an initiator or as a target. If only selection is enabled, the LSI53C895A cannot be reselected as an initiator. There are also status and interrupt bits in the SCSI Interrupt

Status Zero (SIST0) and SCSI Interrupt Enable Zero (SIEN0) registers, respectively, indicating that the LSI53C895A has been selected (bit 5) and reselected (bit 4).

2.2.15 Synchronous Operation

The LSI53C895A can transfer synchronous SCSI data in both the initiator and target modes. The SCSI Transfer (SXFER) register controls both the synchronous offset and the transfer period. It may be loaded by the CPU before SCRIPTS execution begins, from within SCRIPTS using a Table Indirect I/O instruction, or with a Read-Modify-Write instruction.

The LSI53C895A can receive data from the SCSI bus at a synchronous transfer period as short as 25 ns, regardless of the transfer period used to send data. The LSI53C895A 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 LSI53C895A can send synchronous data at intervals as short as 25 ns for Ultra2 SCSI, 50 ns for Ultra SCSI, 100 ns for fast SCSI and 200 ns for SCSI-1.

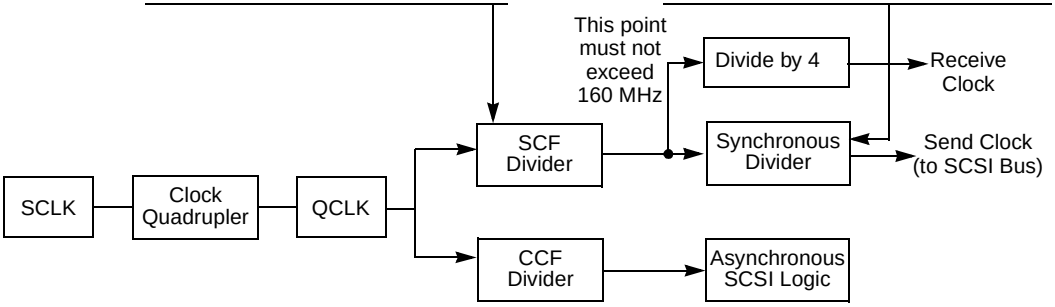
2.2.15.1 Determining the Data Transfer Rate

Synchronous data transfer rates are controlled by bits in two different registers of the LSI53C895A. Following is a brief description of the bits. Figure 2.7 illustrates the clock division factors used in each register, and the role of the register bits in determining the transfer rate.

Figure 2.7 Determining the Synchronous Transfer Rate

SCF2	SCF1	SCF0	SCF Divisor
0	0	1	1
0	1	0	1.5
0	1	1	2
1	0	0	3
0	0	0	3
1	0	1	4
1	1	0	6
1	1	1	8

TP2	TP1	TP0	XFERP Divisor
0	0	0	4
0	0	1	5
0	1	0	6
0	1	1	7
1	0	0	8
1	0	1	9
1	1	0	10
1	1	1	11



CCF2	CCF1	CCF0	Divisor	QCLK (MHz)
0	0	1	1	50.1–66.00
0	1	0	1.5	16.67–25.00
0	1	1	2	25.1–37.50
1	0	0	3	37.51–50.00
0	0	0	3	50.01–66.00
1	0	1	4	75.01–80.00
1	1	0	6	120
1	1	1	8	160

Example:
QCLK (Quadrupled SCSI Clock) = 160 MHz
SCF = 1 (/1), XFERP = 0 (/4), CCF = 7 (/8)

Synchronous send rate = (QCLK/SCF)/XFERP = (160/1) /4 = 40 Mbytes/s

Synchronous receive rate = (QCLK/SCF) /4 = (160/1) /4 = 40 Mbytes/s

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 being presented to the synchronous SCSI control logic. The output from this divider controls the rate at which data can be received; this rate must not exceed 160 MHz. The receive rate of synchronous SCSI data is one-fourth of the SCF divider output. For example, if SCLK is 160 MHz and the SCF value is set to divide by one, then the maximum rate at which data can be received is 40 MHz (160/(1*4) = 40).

2.2.15.3 SCSI Control Three (SCNTL3) Register, Bits [2:0] (CCF[2:0])

The CCF[2:0] bits select the factor by which the frequency of SCLK is divided before being presented to the asynchronous SCSI core logic. This divider must be set according to the input clock frequency in the table.

2.2.15.4 SCSI Transfer (SXFER) Register, Bits [7:5] (TP[2:0])

The TP[2:0] divider bits determine the SCSI synchronous transfer period when sending synchronous SCSI data in either the initiator or target mode. This value further divides the output from the SCF divider.

2.2.15.5 Ultra2 SCSI Synchronous Data Transfers

Ultra2 SCSI is an extension of the current Ultra SCSI synchronous transfer specifications. It allows synchronous transfer periods to be negotiated down as low as 25 ns, which is half the 50 ns period allowed under Ultra SCSI. This will allow a maximum transfer rate of 80 Mbytes/s on a 16-bit, LVD SCSI bus. The LSI53C895A has a SCSI clock quadrupler that must be enabled for the chip to perform Ultra2 SCSI transfers with a 40 MHz oscillator. In addition, the following bit values affect the chip's ability to support Ultra2 SCSI synchronous transfer rates:

- Clock Conversion Factor bits, SCSI Control Three (SCNTL3) register bits [2:0] and Synchronous Clock Conversion Factor bits, SCNTL3 register bits [6:4]. These fields support a value of 111 (binary), allowing the 160 MHz SCLK frequency to be divided by 8 for the asynchronous logic.
- Ultra2 SCSI Enable bit, SCSI Control Three (SCNTL3) register bit 7. Setting this bit enables Ultra2 SCSI synchronous transfers in systems that use the internal SCSI clock quadrupler.
- TolerANT Enable bit, SCSI Test Three (STEST3) register bit 7. Active negation must be enabled for the LSI53C895A to perform Ultra2 SCSI transfers.

  Note: The clock quadrupler requires a 40 MHz external clock. LSI Logic software assumes that the LSI53C895A is connected to a 40 MHz external clock, which is quadrupled to achieve Ultra2 SCSI transfer rates.

2.2.16 Interrupt Handling

The SCRIPTS processors in the LSI53C895A 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 LSI53C895A.

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 wastes CPU time that could be used for other system tasks. The preferred method of detecting interrupts in most systems is hardware interrupts. In this case, the LSI53C895A asserts the Interrupt Request (IRQ/) line that interrupts the microprocessor, causing the microprocessor to execute an interrupt service routine. A hybrid approach would use hardware interrupts for long waits, and use polling for short waits.

2.2.16.2 Registers

The registers in the LSI53C895A that are used for detecting or defining interrupts are Interrupt Status Zero (ISTAT0), Interrupt Status One (ISTAT1), Mailbox Zero (MBOX0), Mailbox One (MBOX1), SCSI Interrupt Status Zero (SIST0), SCSI Interrupt Status One (SIST1), DMA Status (DSTAT), SCSI Interrupt Enable Zero (SIEN0), SCSI Interrupt Enable One (SIEN1), DMA Control (DCNTL), and DMA Interrupt Enable (DIEN).

ISTAT – The ISTAT register includes the Interrupt Status Zero (ISTAT0), Interrupt Status One (ISTAT1), Chip Test Zero (CTEST0), 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 IRQ/ 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 to be cleared. This interrupt must be cleared before servicing any other interrupts.

See Register 0x14, Interrupt Status Zero (ISTAT0) register, Bit 5 Signal process in Chapter 4, “Registers,” for additional information.

The host (C Code) or the SCRIPTS code could potentially try to access the mailbox bits at the same time.

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.

SIST0 and SIST1 – The SCSI Interrupt Status Zero (SIST0) and SCSI Interrupt Status One (SIST1) registers contain SCSI-type interrupt bits. Reading these registers determines which condition or conditions caused the SCSI-type interrupt, and clears that SCSI interrupt condition.

If the LSI53C895A 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.

If the LSI53C895A is sending data to the SCSI bus and a fatal SCSI interrupt condition occurs, data could be left in the DMA FIFO. Because of this the DMA FIFO Empty (DFE) bit in DMA Status (DSTAT) should be checked.

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). The CSF bit is bit 1 in SCSI Test Three (STEST3).

DSTAT – The DMA Status (DSTAT) register contains the DMA-type interrupt bits. Reading this register determines which condition or conditions caused the DMA-type interrupt, and clears that DMA interrupt condition. Bit 7 in DSTAT, DFE, is purely a status bit; it will not generate an interrupt under any circumstances and will not be cleared when read. DMA interrupts flush neither the DMA nor SCSI FIFOs before generating the interrupt, so the DFE bit in the DMA Status (DSTAT) register should be checked after any DMA interrupt.

If the DFE bit is cleared, then the FIFOs must be cleared by setting the CLF (Clear DMA FIFO) and CSF (Clear SCSI FIFO) bits, or flushed by setting the FLF (Flush DMA FIFO) 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).

DIEN – The DMA Interrupt Enable (DIEN) register is the interrupt enable register for DMA interrupts in DMA Status (DSTAT).

DCNTL – When bit 1 in the DMA Control (DCNTL) register is set, the IRQ/ pin is not asserted when an interrupt condition occurs. The interrupt is not lost or ignored, but is merely masked at the pin. Clearing this bit when an interrupt is pending immediately causes the IRQ/ pin to assert. As with any register other than ISTAT, this register cannot be accessed except by a SCRIPTS instruction during SCRIPTS execution.

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 (indicated by the DIP bit in ISTAT and one or more bits in DMA Status (DSTAT) being set) are fatal.

Some SCSI interrupts (indicated by the SIP bit in the Interrupt Status Zero (ISTAT0) and one or more bits in SCSI Interrupt Status Zero (SIST0) or SCSI Interrupt Status One (SIST1) being set) are nonfatal.

When the LSI53C895A is operating in the Initiator mode, only the 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, CMP, SEL, RSL, Target mode: SATN/ active (M/A), GEN, and HTH are nonfatal. Refer to the description for the Disable Halt on a Parity Error or SATN/ active (Target Mode Only) (DHP) bit in the SCSI Control One (SCNTL1) register to configure the

chip's behavior when the SATN/ interrupt is enabled during Target mode operation. The Interrupt-on-the-Fly interrupt is also nonfatal, since SCRIPTS can continue when it occurs.

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 LSI53C895A 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. Interrupts can be masked by clearing bits in the SCSI Interrupt Enable Zero (SIEN0) and SCSI Interrupt Enable One (SIEN1) (for SCSI interrupts) registers or DMA Interrupt Enable (DIEN) (for DMA interrupts) register. 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 is masked and that condition occurs, the SCRIPTS do not stop, 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 IRQ/ pin is not asserted.

If a fatal interrupt is masked and that condition occurs, then the SCRIPTS still stop, the appropriate bit in the DMA Status (DSTAT), SCSI Interrupt Status Zero (SIST0), or SCSI Interrupt Status One (SIST1) register is set, and the SIP or DIP bit in the Interrupt Status Zero (ISTAT0) register is set, but the IRQ/ pin is not asserted.

Interrupts can be disabled by setting SYNC_IRQD bit 0 in the Interrupt Status One (ISTAT1) register. If an interrupt is already asserted and SYNC_IRQD is then set, the interrupt will remain asserted until serviced.

At this point, the IRQ/ pin is blocked for future interrupts until this bit is cleared. When the LSI53C895A is initialized, enable all fatal interrupts if you are using hardware interrupts. If a fatal interrupt is disabled and that

interrupt condition occurs, the SCRIPTS halt and the system never knows it unless it times out and checks the ISTAT register after a certain period of inactivity.

If you are polling the ISTAT 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 IRQ/ pin.

Masking an interrupt after IRQ/ is asserted does not cause deassertion of IRQ/.

2.2.16.5 Stacked Interrupts

The LSI53C895A will stack interrupts if they occur one after the other. If the SIP or DIP bits in the ISTAT register are set (first level), then there is already at least one pending interrupt, and any future interrupts 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 SCSI Interrupt Status Zero (SIST0), SCSI Interrupt Status One (SIST1), and DMA Status (DSTAT). When the first level of interrupts are cleared, all the interrupts that came in afterward move into SIST0, SIST1, and DSTAT. After the first interrupt is cleared by reading the appropriate register, the IRQ/ pin is deasserted for a minimum of three CLKs; the stacked interrupts move into SIST0, SIST1, or DSTAT; and the IRQ/ 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 SIST0, but does not assert the IRQ/ pin. Since no interrupt is generated, future interrupts move into SCSI Interrupt Status Zero (SIST0) or SCSI Interrupt Status One (SIST1) instead of being stacked behind another interrupt. When another condition occurs that generates an interrupt, the bit corresponding to the earlier masked nonfatal interrupt is still 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 LSI53C895A 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 DMA SCRIPTS Pointer (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 LSI53C895A 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) register should be checked to see if any data remains in the DMA FIFO.
- SCSI SREQ/SACK handshakes that have begun are completed before halting.
- The LSI53C895A 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.
- If the instruction is a JUMP/CALL WHEN/IF <phase>, 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 for the LSI53C895A. It can be repeated during polling or should be called when the IRQ/ pin is asserted during hardware interrupts.

1. Read Interrupt Status Zero (ISTAT0).
2. If the INTF bit is set, it must be written 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 get the DMA interrupt status. The bits in DSTAT tells 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 get the interrupt status. If using 8-bit reads of the SIST0, SIST1, and DSTAT registers to clear interrupts, insert a 12 CLK 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 interrupt service routine. 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 interrupt service routine, in case any stacked interrupts moved in when the first interrupt was cleared. When using hardware interrupts, the IRQ/ pin is asserted again if there are any stacked interrupts. This should cause the system to re-enter the interrupt service routine.

2.2.17 Interrupt Routing

This section documents the recommended approach to RAID ready interrupt routing for the LSI53C895A. In order to be compatible with AMI RAID upgrade products and the LSI53C895A, the following requirements must be met:

- When a RAID upgrade card is installed in the upgrade slot, interrupts from the mainboard SCSI controller(s) assigned to the RAID upgrade card must be routed to INTB/, INTC/ and INTD/ of the upgrade slot and isolated from the mainboard interrupt controller. The system processor must not see interrupts from the SCSI controllers that are to be serviced by the RAID upgrade card. An upgrade slot is one that is connected to the interrupt routing logic for mainboard SCSI device(s). When a PCI RAID upgrade board is installed into the system, it would be plugged into this slot if it is to control mainboard SCSI device(s).
- 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.

The LSI53C895A supports four different interrupt routing modes. Additional information for these modes may be found in the Register 0x4D SCSI Test One (STEST1) description in Chapter 4, "Registers." The interrupt routing mode is selected using bits [1:0] in the STEST1 register. Mode 0 is the default mode and is compatible with AMI RAID upgrade products. In this mode, interrupts are presented on IRQ/ and ALR_IRQ/.

If INTB/, INTC/ or INTD/ of the PCI RAID upgrade slot is used in the interrupt routing scheme, it 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 mainboard. As long as the interrupt routing requirements stated above are satisfied, a mainboard designer could implement this design with external logic.

There can only be one entity controlling a mainboard LSI53C895A or conflicts will occur. Typically, SCSI BIOS and an operating system driver control the LSI53C895A. When allocated to a RAID adapter, however, a mechanism is implemented to prevent the SCSI BIOS and operating system driver from trying to access the chip. The mainboard designer has several options, listed below.

The first option is to have the LSI53C895A load its PCI Subsystem ID using a serial EPROM on power-up. If bit 15 in this ID is set, the LSI Logic BIOS and operating system drivers (not all versions support this capability) will ignore the chip. This makes it possible to control the assignment of the mainboard SCSI controller using a configuration utility.

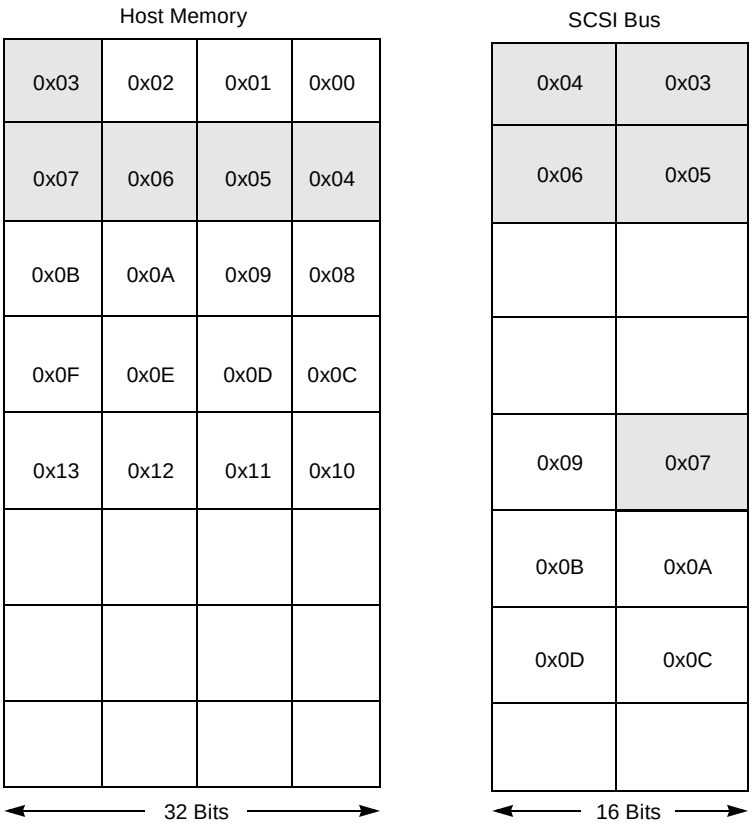
The second option is to provide mainboard and system BIOS support for NVS. You can then enable or disable the LSI53C895A using the SCSI BIOS configuration utility. Not all versions of the LSI Logic drivers support this capability.

The third option is to have the system BIOS not report the existence of the SCSI controller 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 LSI53C895A 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 2.8.

Figure 2.8 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 low-order byte of the SCSI Output Data Latch (SODL) register and is married with the first byte of the following MOVE instruction.

MOVE 5, 9 when Data_Out

Moves five bytes from address 0x09 in the host memory to the SCSI bus.

2.2.18.1 Wide SCSI Send Bit

The WSS bit is set whenever the SCSI controller is sending data (Data-Out for initiator or Data-In for target) and 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 and the WSS flag is set. The hardware uses the WSS flag to determine what behavior must occur at the start of the next data send transfer. When the WSS flag 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 stored low-order byte in the SODL register; and the two bytes are sent out across the bus, regardless of the type of Block Move instruction (normal or chained). The flag is automatically cleared when the "married" word is sent. The flag is alternately cleared through SCRIPTS or by the microprocessor. Also, the microprocessor or SCRIPTS can use this bit for error detection and recovery purposes.

2.2.18.2 Wide SCSI Receive Bit

The WSR bit is set whenever the SCSI controller is receiving data (Data-In for initiator or Data-Out for target) and the controller detects a partial transfer at the end of a block move or chained block move SCRIPTS instruction. When WSR is set, the high-order byte of the last SCSI bus transfer is not transferred to memory. Instead, the byte is temporarily stored in the SCSI Wide Residue (SWIDE) register. The hardware uses the WSR bit to determine what behavior must occur at the start of the next data receive transfer. The bit is automatically cleared at the start of the next data receive transfer. The bit can alternatively be cleared by the microprocessor or through SCRIPTS. Also, the microprocessor or SCRIPTS can use this bit for error detection and recovery purposes.

2.2.18.3 SWIDE Register

This register stores data for partial byte data transfers. For receive data, 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 send data, 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 command.

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 initiator or Data-Out for 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 contents of the SWIDE register should be the first byte transferred to memory at the start of the chained Block Move data stream. Since the byte count always represents data transfers to/from memory (as opposed to the SCSI bus), the byte transferred out of the SCSI Wide Residue (SWIDE) register is one of the bytes in the byte 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. Whether the WSR bit is set or cleared, when a normal block move instruction is executed, the contents of the SCSI Wide Residue (SWIDE) register are ignored and the transfer takes place normally. For "N" consecutive wide data receive Block Move instructions, the 2nd through the Nth Block Move instructions should be chained block moves.

For send data (Data-Out for initiator or Data-In for target), a chained Block Move instruction indicates that if a partial transfer terminates the chained block move instruction, the last low-order byte (the partial memory transfer) should be stored in the lower byte of the SCSI Output Data Latch (SODL) register 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 in the lower byte of the SCSI Output Data Latch (SODL) register waiting to be married with the first byte of the next Block Move instruction. Regardless of whether a chained Block Move or normal Block Move instruction is used, if the WSS bit is set at the start of a data send command, the first byte of the data send command is assumed to be the high-order byte and is "married" with the low-order byte stored in the lower byte of the SCSI Output Data Latch (SODL) register before the two bytes are sent across the SCSI bus. For "N" consecutive wide data send Block Move commands, the first through the (N^{th} – 1) Block Move instructions should be Chained Block Moves.

2.3 Parallel ROM Interface

The LSI53C895A supports up to one megabyte of external memory in binary increments from 16 Kbytes, to allow the use of expansion ROM for add-in PCI cards. 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 LSI53C895A, 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. If in-system Flash ROM updates are required, a 7406 (high voltage open collector inverter), a MTD4P05, and several passive components are also needed. The memory size and speed is determined by pull-up resistors on the 8-bit bidirectional memory bus at power-up. The LSI53C895A senses this bus shortly after the release of the Reset signal and configures the Expansion ROM Base Address register and the memory cycle state machines for the appropriate conditions.

The external memory interface works with a variety of ROM sizes and speeds. An example set of interface drawings is in Appendix B, "External Memory Interface Diagram Examples."

The LSI53C895A supports a variety of sizes and speeds of expansion ROM, using pull-down resistors on the MAD[3:0] pins. The encoding of pins MAD[3:1] allows the user to define how much external memory is available to the LSI53C895A. Table 2.7 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 2.7 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

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. For example, to connect to a 64 Kbyte external ROM, use a pull-up on MAD2. If the external memory interface is not used, MAD[3:1] should be pulled HIGH.

Note: There are internal pull-downs on all of the MAD bus signals.

The LSI53C895A allows the system to determine the size of the available external memory using the Expansion ROM Base Address 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."

MAD0 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

The LSI53C895A implements an interface that allows attachment of a serial EEPROM device to the GPIO0 and GPIO1 pins. There are two modes of operation relating to the serial EEPROM and the Subsystem ID and Subsystem Vendor ID registers. These modes are programmable through the MAD7 pin which is sampled at power-up.

Also, the LSI53C895A implements a method for programming the Subsystem ID and Subsystem Vendor ID registers without a serial EEPROM download. Please see Section 2.5, “Alternative SSVID/SSID Loading Mechanism,” for additional information.

2.4.1 Default Download Mode

In this mode, MAD7 is pulled down internally, GPIO0 is the serial data signal (SDA) and GPIO1 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 2.8. If the download is enabled and an EEPROM is not present, or the checksum fails, the Subsystem ID and Subsystem Vendor ID registers read back all zeros. At power-up, only 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.

Table 2.8 Mode A Serial EEPROM Data Format

Byte	Name	Description
0xFB	SVID(0)	Subsystem Vendor ID, 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, 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, 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, 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. This 8 bit checksum is formed by adding, bytewise, each byte contained in locations 0x00–0x03 to the seed value 0x55, and then taking the 2's complement of the result.
0x100–0xEOM	UD	User Data.

2.4.2 No Download Mode

When MAD7 is pulled up through an external resistor, the automatic download is disabled and no data is automatically loaded into chip registers at power-up. The Subsystem ID and Subsystem Vendor ID registers are read only, per the PCI specification, with a default value of 0x1000 and 0x1000 respectively.

2.5 Alternative SSVID/SSID Loading Mechanism

Programming the PCI Subsystem ID and Subsystem Vendor ID registers can be accomplished in the LSI53C895A without the use of a serial EEPROM. This alternative loading mechanism is the only way to set the SSVID/SSID registers to something other than the default value, except through Serial EEPROM download.

Please see Section 2.4, "Serial EEPROM Interface," for additional information.

An additional register, the Subsystem ID Access, is located in the PCI configuration space at offset 0x48–0x4B. This is a 32-bit write only register that always reads back a value of 0x00000000. Once enabled and unlocked using a write of three specific byte values to offset 0x48, a write to this register is shadowed into the PCI Subsystem register at offset 0x2C. Any data written to the register cannot be read back through the register, it always reads back 0x00000000.

To disable the Subsystem ID Access, the MAD4 pin must be pulled HIGH. (Note: There is an internal pull-down on the MAD4 pin.) A logical zero (0) on this pin will enable the Subsystem ID Access register, allowing it to be unlocked by writing the proper three byte sequence to offset 0x48. A logical one (1) on this pin will disable the Subsystem ID Access register.

Once the Subsystem ID Access register is enabled, a sequence of three byte writes to offset 0x48 will allow a 32-bit subsystem value to be written to offset 0x48–0x4B which will then be shadowed into the PCI Subsystem register at offset 0x2C–0x2F. The three byte values that must be written are 0x53, 0x59, 0x4D (ASCII) in this order. Once this sequence is written, the next write to offset 0x48–0x4B is shadowed into the Subsystem register. At no time can any of the data written to the Subsystem ID register be read back. The register always reads back zeros. In addition, any reads to offset 0x48 between writes of the unlock code or the actual subsystem value will reset the internal state machine requiring the sequence be restarted from the beginning. Byte, word, or Dword writes are allowed during the unlock sequence with the other byte lanes (0x49, 0x4A, and 0x4B) being don't cares. Once the subsystem value is written into the Subsystem ID Access register, the register will again lock itself and the three byte sequence must be repeated to allow further writes to this register to be shadowed into the Subsystem register at offset 0x2C–0x2F.

If the Subsystem ID Access register writes a new value to the Subsystem register (0x2C), the Subsystem register retains that value until the Alternative SSVID/SSID Loading Mechanism is used again to change the value. Prior to the first unlock of the Subsystem ID Access register after a power-on, the Subsystem register presents the value determined by the Subsystem ID and Subsystem Vendor ID described in the section above. This allows an autodownload from a serial EEPROM to change the value of the Subsystem register. Additionally, that value may be overridden by writing to the Subsystem ID Access register. The serial

EEPROM value is always the first value loaded (if that mechanism is enabled). The system would then have the opportunity to override the value loaded from the serial EEPROM.

Below is an example of how the enabling sequence occurs:

1. Ensure that the MAD4 pin is at a logical zero during power-up of the LSI53C895A. This enables the Subsystem ID Access register.
2. Write value 0x53 to PCI offset 0x48 using a PCI Configuration Write.
3. Write value 0x59 to PCI offset 0x48 using a PCI Configuration Write.
4. Write value 0x4D to PCI offset 0x48 using a PCI Configuration Write. The Subsystem ID Access register is now unlocked for a single write.
5. Write the desired subsystem value to offset 0x48–0x4B using a PCI Configuration Write.
6. Read back the Subsystem register at PCI offset 0x2C–0x2F to verify the new value written in Step 5.
7. Return to Step 2 to change the subsystem value at offset 0x2C.

   Note: During the unlock sequence byte, word, or Dword writes are allowed, but with the other byte lanes being don't cares.

The following events will reset the lock mechanism:

- A PCI Reset.
- Any reads to offset 0x48–0x4B between Steps 2 through 5.
- Any writes other than the specified data values between Steps 2 through 4.
- The write of the subsystem value in Step 5.

2.6 Power Management

The LSI53C895A 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 LSI53C895A power states shown in Table 2.9 are independently controlled through two power state bits that are located in the PCI Power Management Control/Status (PMCSR) register 0x44.

Table 2.9 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 LSI53C895A hardware places no restriction on transitions between power states.

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. Power state actions are separate for each function.

2.6.1 Power State D0

Power state D0 is the maximum power state and is the power-up default state. The LSI53C895A is fully functional in this state.

2.6.2 Power State D1

Power state D1 is a lower power state than D0. In this state, the LSI53C895A core is placed in the snooze mode and the SCSI CLK is disabled. In the snooze mode, a SCSI reset does not generate an IRQ/ signal. However, the SCSI CLK is still disabled.

2.6.3 Power State D2

Power state D2 is a lower power state than D1. In this state the LSI53C895A core is placed 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 memory and I/O spaces cannot be accessed, and the LSI53C895A cannot be a PCI bus master. Furthermore, all interrupts are disabled when in power state D2. If changed from power state D2 to power state D0, the previous values of the PCI command register are restored. Also, any pending interrupts before the function entered power state D2 are asserted.

2.6.4 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 LSI53C895A 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. Furthermore, the device'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 device's PCI command register is cleared and the Clock Quadrupler is disabled, which results in additional power savings.

Chapter 3
Signal Descriptions

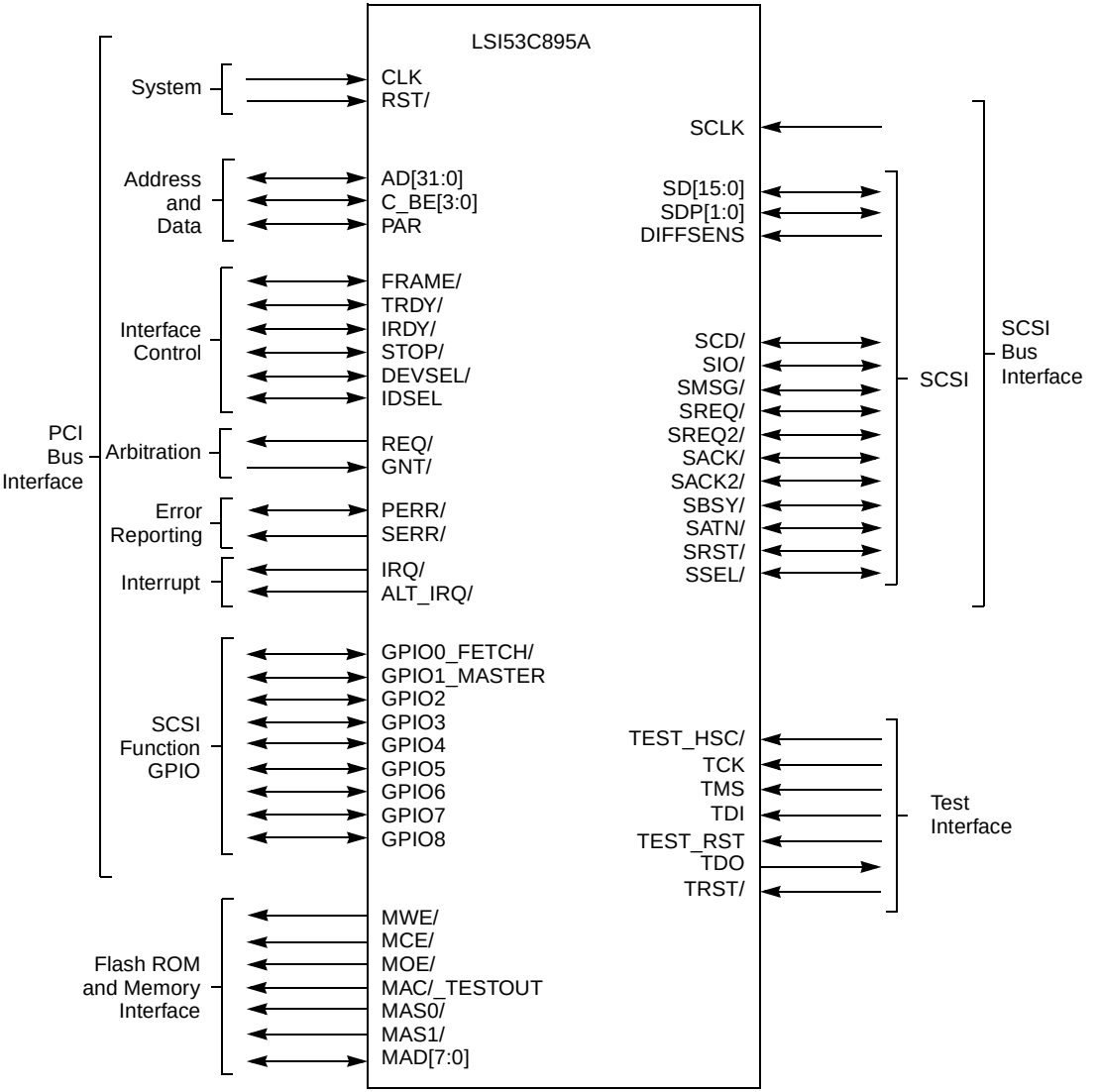
This chapter presents the LSI53C895A pin configuration and signal definitions using tables and illustrations. The LSI53C895A comes in a 208 PQFP and a 272 BGA package. Definitions in the signal description tables are for both the 208 PQFP and the 272 BGA. This chapter contains the following sections:

A slash (/) at the end of a signal name indicates that the active state occurs when the signal is at a LOW voltage. When the slash is absent, the signal is active at a HIGH voltage.

3.1 LSI53C895A Functional Signal Grouping

Figure 3.1 presents the LSI53C895A signals by functional group.

Figure 3.1 LSI53C895A Functional Signal Grouping



3.2 Signal Descriptions

The Signal Descriptions are divided into PCI Bus Interface Signals, SCSI Bus Interface Signals, Flash ROM and Memory Interface Signals, Test Interface Signals, and Power and Ground Signals.

The PCI Bus Interface Signals are subdivided into System Signals, Address and Data Signals, Interface Control Signals, Arbitration Signals, Error Reporting Signals, Interrupt Signals, and SCSI GPIO Signals.

The SCSI Bus Interface Signals are subdivided into SCSI Bus Interface Signals, SCSI Signals, and SCSI Control Signals.

Signals are assigned a Type. There are five signal types:

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.

3.2.1 Internal Pull-ups on LSI53C895A Signals

Several signals in the LSI53C895A have internal pull-up resistors. Table 3.1 describes the conditions that enable these pull-ups.

Table 3.1 LSI53C895A Internal Pull-ups

Signal Name	Pull-up Current	Conditions for Pull-up
IRQ/, ALT_IRQ/	25 µA	Pull-up enabled when the IRQ mode bit (bit 3 of DMA Control (DCNTL) (0x3B)) is cleared.
GPIO[1:0]	25 µA	Pull-up enabled when bits [1:0] of General Purpose Pin Control Zero (GPCNTL0) are not set.
TEST_HSC/	25 µA	Pull-up enabled all the time.
TEST_RST/	25 µA	Pull-up enabled all the time.
TRST, TCK, TMS, TDI	25 µA	Pull-up enabled all the time.

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, and SCSI GPIO Signals.

3.3.1 System Signals

Table 3.2 describes the System signals.

Table 3.2 System Signals

Name	PQFP	BGA Pos	Type	Strength	Description
CLK	195	T1	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, and other timing parameters are defined with respect to this edge. Clock can optionally serve as the SCSI core clock, but this may effect fast SCSI-2 (or faster) transfer rates.
RST/	194	R2	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. The CLK input must be active while RST/ is active to properly reset the device.

3.3.2 Address and Data Signals

Table 3.3 describes Address and Data signals.

Table 3.3 Address and Data Signals

Name	PQFP	BGA Pos	Type	Strength	Description
AD[31:0]	199, 201–204, 3, 5, 6, 10–12, 14–17, 19, 33–35, 37–40, 42, 44, 45, 47, 48, 50, 51, 57, 58	U2, V1, V2, W1, V3, Y3, V5, W5, W6, Y6, V7, W7, Y7, V8, W8, Y8, V12, Y13, W13, V13, Y14, W14, Y15, W15, Y17, W17, Y18, V17, Y19, V18, U18, V20	T/S	8 mA PCI	Physical Dword **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[31:0] contain a 32-bit physical byte address. If the command is a DAC, implying a 64-bit address, a second address phase is required. During the first phase, AD[31:0] will contain the lower 32 bits of the address followed by a second phase with AD[31:0] containing the upper 32 bits of the address. During subsequent clocks, AD[31:0] contain data. PCI supports both read and write bursts. AD[7:0] define the least significant byte, and AD[31:24] define the most significant byte.
C_BE[3:0]/	7, 20, 32, 43	Y5, U9, W12, Y16	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. During the data phase, C_BE[3:0]/ 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[3]/ to byte 3.

Table 3.3 Address and Data Signals (Cont.)

Name	PQFP	BGA Pos	Type	Strength	Description
PAR	30	Y12	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 data phase, both data and byte enables are covered.

3.3.3 Interface Control Signals

Table 3.4 describes the Interface Control signals.

Table 3.4 Interface Control Signals

Name	PQFP	BGA Pos	Type	Strength	Description
FRAME/	21	V9	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/	24	W10	S/T/S	8 mA PCI	**Target Ready** indicates the target agent's (selected device'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 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 AD[31:0]. During a write, it indicates that the target is prepared to accept data. Wait cycles are inserted until both IRDY/ and TRDY/ are asserted together.

Table 3.4 Interface Control Signals (Cont.)

Name	PQFP	BGA Pos	Type	Strength	Description
IRDY/	22	W9	S/T/S	8 mA PCI	**Initiator Ready** indicates the initiating agent's (bus master'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 AD[31:0]. During a read, it indicates that the master is prepared to accept data. Wait cycles are inserted until both IRDY/ and TRDY/ are asserted together.
STOP/	27	W11	S/T/S	8 mA PCI	**Stop** indicates that the selected target is requesting the master to stop the current transaction.
DEVSEL/	25	Y10	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	9	V6	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 3.5 describes Arbitration signals.

Table 3.5 Arbitration Signals

Name	PQFP	BGA Pos	Type	Strength	Description
REQ/	198	U1	O	8 mA PCI	**Request** indicates to the system arbiter that this agent desires use of the PCI bus. This is a point-to-point signal. Every master has its own REQ/ signal.
GNT/	196	T2	I	N/A	**Grant** indicates to the 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 3.6 describes the Error Reporting signals.

Table 3.6 Error Reporting Signals

Name	PQFP	BGA Pos	Type	Strength	Description
PERR/	28	V11	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. However, on detection of a PERR/ pulse, the central resource may generate a nonmaskable interrupt to the host CPU, which often implies the system is unable to continue operation once error processing is complete.
SERR/	29	U11	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 3.7 describes the Interrupt signals.

Table 3.7 Interrupt Signals

Name	PQFP	BGA Pos	Type	Strength	Description
IRQ/	59	U20	O	8 mA PCI	**Interrupt Request.** This signal, when asserted LOW, indicates that an interrupting condition has occurred and that service is required from the host CPU. The output drive of this pin is open drain.
ALT_IRQ/	1	Y2	O	8 mA PCI	**Alt Interrupt Request.** When asserted LOW, it indicates that an interrupting condition has occurred and that service is required from the host CPU. The output drive of this pin is open drain.

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

3.3.7 SCSI GPIO Signals

Table 3.8 describes the SCSI GPIO signals.

Table 3.8 SCSI GPIO Signals

Name	PQFP	BGA Pos	Type	Strength	Description
GPIO0_ FETCH/	61	T19	I/O	8 mA	**SCSI General Purpose I/O pin.** Optionally, when driven LOW, indicates that the next bus request will be for an opcode fetch. This pin is programmable at power-up through the MAD7 pin to serve as the data signal for the serial EEPROM interface. This signal can also be programmed to be driven LOW when the LSI53C895A is active on the SCSI bus.
GPIO1_ MASTER/	63	R18	I/O	8 mA	**SCSI General Purpose I/O pin.** Optionally, when driven LOW, indicates that the LSI53C895A is bus master. This pin is programmable at power-up through the MAD7 pin to serve as the clock signal for the serial EEPROM interface.
GPIO2	65	R20	I/O	8 mA	**SCSI General Purpose I/O pin.** This pin powers up as an input.
GPIO3	66	P18	I/O	8 mA	**SCSI General Purpose I/O pin.** This pin powers up as an input.
GPIO4	67	P19	I/O	8 mA	**SCSI General Purpose I/O pin.** GPIO4 powers up as an output. (This pin may be used as the enable line for VPP, the 12 V power supply to the external flash memory interface.)
GPIO5	52	W19	I/O	8 mA	**SCSI General Purpose I/O pin.** This pin powers up as an input.
GPIO6	54	W20	I/O	8 mA	**SCSI General Purpose I/O pin.** This pin powers up as an input.
GPIO7	55	V19	I/O	8 mA	**SCSI General Purpose I/O pin.** This pin powers up as an input.
GPIO8	205	W2	I/O	8 mA	**SCSI General Purpose I/O pin.** This pin powers up as an input.

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 Signals, SCSI Control Signals.

3.4.1 SCSI Bus Interface Signal

Table 3.9 describes the SCSI Bus Interface signal.

Table 3.9 SCSI Bus Interface Signal

Name	PQFP	BGA Pos	Type	Strength	Description
SCLK	80	J20	I	N/A	**SCSI Clock** is used to derive all SCSI-related timings. The speed of this clock is determined by the application's requirements. In some applications, SCLK may be sourced internally from the PCI bus clock (CLK). If SCLK is internally sourced, then the SCLK pin should be tied LOW. For Ultra2 SCSI operations, the clock supplied to SCLK must be 40 MHz. The clock frequency will be quadrupled to create the 160 MHz clock required by the SCSI core.

3.4.2 SCSI Signals

Table 3.10 describes the SCSI signals.

Table 3.10 SCSI Signals

Name	PQFP	BGA Pos	Type	Strength	Description
SD[15:0]– SDP[1:0]–	167, 170, 172, 175, 87, 89, 92, 94, 135, 137, 140, 142, 145, 147, 149, 162 165, 132	F2, G2, H2, J3, G20, F20, E20, D20, A9, A8, A7, B6, B5, B4, B3, C1 E1, B10	I/O	48 mA SCSI	SCSI Data. SCSI Parity.
LVD Mode: Negative half of LVDlink pair for SCSI data and parity lines. SD[15:0]– are the 16-bit SCSI data bus, and SDP[1:0]– are the SCSI data parity lines. SE Mode: SD[15:0]– are the 16-bit SCSI data bus, and SDP[1:0]– are the SCSI data parity lines. HVD Mode: SD[15:0]– and SDP1:0]– are the SCSI data bus and parity lines.					
SD[15:0]+ SDP[1:0]+	168, 171, 173, 176, 88, 90, 93, 95, 136, 138, 141, 143, 146, 148, 150, 163 166, 133	F1, G1, H1, J2, G19, F19, E19, D19, B9, B8, B7, A5, A4, A3, B2, D1 F3, C10	I/O	48 mA SCSI	SCSI Data. SCSI Parity.
LVD Mode: Positive half of LVDlink pair for SCSI data lines. SD[15:0]+ are the 16-bit data bus, and SDP[1:0]+ are the SCSI data parity lines. SE Mode: SD[15:0]+ and SDP[1:0]+ are at 0 Volts. HVD Mode: SD[15:0]+ and SDP[1:0]+ are driver directional control for SCSI data bus and parity lines.					
DIFFSENS	84	H20	I	N/A	**SCSI Differential Sense** pin detects the present mode of the SCSI bus when connected to the DIFFSENS signal on the physical SCSI bus.
LVD Mode: When a voltage between 0.7 V and 1.9 V is present on this pin, the LSI53C895A will operate in the LVD mode. SE Mode: When this pin is driven LOW (below 0.5 V) indicating SE bus operation, the LSI53C895A will operate in the SE mode. HVD Mode: When this pin is detected HIGH (above 2.4 V) indicating a HVD bus, the LSI53C895A will 3-state its SCSI drivers. Set the DIF bit in SCSI Test Two (STEST2) to enable HVD drivers.					

3.4.3 SCSI Control Signals

Table 3.11 describes the SCSI Control signals.

Table 3.11 SCSI Control Signals

Name	PQFP	BGA Pos	Type	Strength	Description
SCSI Control includes the following signals:					
SCD– SCD+	111 112	C17 D16	I/O	48 mA SCSI	SCSI phase line, command/data.
SIO– SIO+	97 98	C20 E17			SCSI phase line, input/output.
SMSG– SMSG+	116 117	B16 A16			SCSI phase line, message.
SREQ– SREQ+	99 100	D18 C19			Data handshake line from target device.
SREQ2– SREQ2+	101 102	B20 B19			Data handshake line from target device. Duplicate of SREQ– and SREQ+ enabled by pulling MAD5 HIGH at reset.
SACK– SACK+	121 122	B14 A14			Data handshake signal from the initiator device.
SACK2– SACK2+	160 161	D2 D3			Data handshake signal from the initiator device. Duplicate of SACK– and SACK+ enabled by pulling MAD5 HIGH at reset.
SBSY– SBSY+	123 124	B13 A13			SCSI bus arbitration signal, busy.
SATN– SATN+	126 127	B12 A12			SCSI Attention, the initiator is requesting a message out phase.
SRST– SRST+	118 119	B15 A15			SCSI bus reset.
SSEL– SSEL+	113 114	A18 A17			SCSI bus arbitration signal, select device.

For all SCSI control signals:
LVD Mode: Negative and positive halves of LVDlink signal pairs shown for the SCSI Control.
SE Mode: SCSI Control signals shown. + signals are at 0 Volts.
HVD Mode: SCSI Control signals shown. + signals become direction control.

3.5 Flash ROM and Memory Interface Signals

Table 3.12 describes the Flash ROM and Memory Interface signals.

Table 3.12 Flash ROM and Memory Interface Signals

Name	PQFP	BGA Pos	Type	Strength	Description
MWE/	188	N2	O	4 mA	**Memory Write Enable.** This pin is used as a write enable signal to an external flash memory.
MCE/	191	R1	O	4 mA	**Memory Chip Enable.** This pin is used as a chip enable signal to an external EEPROM or flash memory device.
MOE/	189	N3	O	4 mA	**Memory Output Enable.** This pin is used as an output enable signal to an external EEPROM or flash memory during read operations. It is also used to test the connectivity of the LSI53C895A signals in test mode.
MAC/_ TESTOUT	79	K19	O	16 mA	**Memory Access Control.** This pin can be programmed to indicate local or system memory accessed (non-PCI applications). It is also used to test the connectivity of the LSI53C895A signals in test mode.
MAS0/	186	M2	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 EEPROM or flash memory. Since the LSI53C895A moves addresses eight bits at a time, this pin connects to the clock of an external bank of flip-flops which are used to assemble up to a 20-bit address for the external memory.

Table 3.12 Flash ROM and Memory Interface Signals (Cont.)

Name	PQFP	BGA Pos	Type	Strength	Description
MAS1/	185	M1	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 EEPROM or flash memory. Since the LSI53C895A moves addresses eight bits at a time, this pin connects to the clock of an external bank of flip-flops which assemble up to a 20-bit address for the external memory.
MAD[7:0]	69, 70, 71, 72, 74, 75, 76, 77	N19, N20, M18, M19, M20, L19, L20, K20	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 EEPROM or flash memory. This bus will put out the least significant byte first 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 EEPROM/flash memory. These pins have static pull-downs.

3.6 Test Interface Signals

Table 3.13 describes Test Interface signals.

Table 3.13 Test Interface Signals

Name	PQFP	BGA Pos	Type	Strength	Description
TEST_HSC/	82	J19	I	N/A	**Test Halt SCSI Clock.** For LSI Logic pulled HIGH internally. This signal can also cause a full chip reset.
TCK	180	K1	I	N/A	**Test Clock.** This pin provides the clock for the JTAG test logic.
TMS	181	L1	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-down.
TDI	183	L3	I	N/A	**Test Data In.** Serial test instructions are received by the JTAG test logic at this pin. This pin has a static pull-down.
TEST_RST/	178	K2	I	N/A	**Test Reset.** For test purposes only. Pulled HIGH internally.
TDO	182	L2	O	4 mA	**Test Data Out.** This pin is the serial output for test instructions and data from the JTAG test logic.
TRST/	206	Y1	I	N/A	**Test Reset**. This pin provides a reset for JTAG Test Logic. Pulled HIGH internally.

3.7 Power and Ground Signals

Table 3.14 describes the Power and Ground signals.

Table 3.14 Power and Ground Signals

Name	PQFP	BGA Pos	Type	Strength	Description
VSS_I/O	8, 18, 31, 41, 56, 78, 91, 110, 120, 128, 131, 139, 151, 169, 179, 193, 200	A1, D4, D8, D13, D17, H4, H17, J9–12, K9–12, L9–12, M9–12, N4, N17, U4, U8, U13, U17	G	N/A	Ground for PCI bus drivers/receivers, SCSI bus drivers/receivers, local memory interface drivers, and other I/O pins.
VDD_I/O	2, 13, 23, 26, 36, 46, 60, 73, 81, 86, 96, 115, 125, 134, 144, 164, 174, 184, 197	D6, D11, D15, F4, F17, K4, L17, R4, R17, U6, U10, U15	P	N/A	Power for PCI bus drivers/receivers, SCSI bus drivers/receivers, local memory interface drivers/receivers, and other I/O pins.
VDD_CORE	64, 190	P1–2, P17, R19	P	N/A	Power for core logic.
VSS_CORE	68, 187	M3, N18, P20	G	N/A	Ground for core logic.
VSS_CORE2		N1	G	N/A	Ground for core logic.
VDDA	85	H19	P	N/A	Power for analog cells (clock quadrupler and diffsense logic).
VSSA	83	J18	G	N/A	Ground for analog cells (clock quadrupler and diffsense logic).
VDD_RBIAS RBIAS	129 130	A11 A10	I	N/A	Used to connect an external resistor to generate the bias current used by LVDlink pads. Resistor value should be 9.76 kΩ. Connect other end of resistor to V_{DD}.

Table 3.14 Power and Ground Signals (Cont.)

Name	PQFP	BGA Pos	Type	Strength	Description
NC	4, 49, 53, 62, 103–109, 152–159, 177, 192, 207, 208	A2, A6, A19–A20, B1, B11, B17–18, C2–9, C11–16, C18, D5, D7, D9–10, D12, D14, E2–4, E18, F18, G3–4, G17–18, H3, H18, J1, J4, J17, K3, K17–18, L4, L18, M4, M17, P3–4, R3, T3–4, T17–18, T20, U3, U5, U7, U12, U14, U16, U19, V4, V10, V14–16, W3–4, W16, W18, Y4, Y9, Y11, Y20	N/A	N/A	These pins have NO internal connection.

Note:

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. These power and ground pins should be connected directly to the primary power and ground planes of the circuit board. Bypass capacitors of 0.01 µF should be applied 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.

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_{SS}. The pull-down resistors require that HC or HCT external components are used for the memory interface. The MAD[7:0] pins are sensed by internal circuitry three PCI clock cycles after RST/ is deasserted.

- **MAD[7]** – Serial EEPROM programmable option. When allowed to be 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. Please see Section 2.4, "Serial EEPROM Interface," in Chapter 2 and Subsystem ID and Subsystem Vendor ID registers in Chapter 4 for additional information.
- **MAD[6]** – This signal is Reserved and may be left floating.
- **MAD[5]** – Enables duplicate SCSI SREQ/ and SACK/ signals. When pulled LOW by the internal pull-down current sink, the duplicate SCSI SREQ/ and SACK/ signals are disabled. When pulled HIGH by an external resistor, the duplicate SCSI SREQ/ and SACK/ signals are enabled. If these duplicate signals are enabled, they must also be terminated.
- **MAD[4]** – Enables the alternative SSVID/SSID loading mechanism for Subsystem ID and Subsystem Vendor ID. When allowed to be pulled LOW by the internal pull-down current sink, the alternative SSVID/SSID loading mechanism for the Subsystem ID and Subsystem Vendor ID is enabled. When pulled HIGH by an external resistor, the alternative loading mechanism for the Subsystem ID and Vendor ID is disabled.

  For additional information, see the two topics: Section 2.5, "Alternative SSVID/SSID Loading Mechanism," and Section 2.4, "Serial EEPROM Interface," in Chapter 2.

- **MAD[3:1]** – These pins are used to set the size of the external expansion ROM device attached. Encoding for these pins are listed in Table 3.15 (“0” indicates a pull-down resistor is attached, “1” indicates a pull-up resistor is attached).

Table 3.15 Decode of MAD Pins

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

- **MAD[0]** – This pin is the slow ROM pin. When pulled up, it enables two extra cycles of data access time to allow use of slower memory devices.

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

Chapter 4
Registers

This chapter describes all LSI53C895A registers and is divided into the following sections:

- Section 4.1 “PCI Configuration Registers”
- Section 4.2 “SCSI Registers”
- Section 4.3 “64-Bit SCRIPTS Selectors”
- Section 4.4 “Phase Mismatch Jump Registers”

In the register descriptions, the term “set” is used to refer to bits that are programmed to a binary one. Similarly, the term “cleared” is used to refer to bits that are programmed to a binary zero. Write any bits marked as reserved to zero; mask all information read from them. Reserved bit functions may change at any time. Unless otherwise indicated, all bits in registers are active HIGH, that is, the feature is enabled by setting the bit. The bottom row of every register diagram shows the default register values, which are enabled after the chip is powered on or reset. Reserved registers and bits are shaded in the register tables.

4.1 PCI Configuration Registers

The PCI Configuration registers are accessed by performing a configuration read/write to the device with its IDSEL pin asserted and the appropriate value in AD[10:8] during the address phase of the transaction. The LSI53C895A will respond to a binary value of 000b.

Table 4.1 shows the PCI configuration registers implemented in the LSI53C895A.

All PCI-compliant devices, such as the LSI53C895A, must support the Vendor ID, Device ID, Command, and Status registers. Support of other PCI-compliant registers is optional. In the LSI53C895A, 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 LSI53C895A are described in this chapter. Reserved bits should not be accessed.

Table 4.1 PCI Configuration Register Map

31		16 15	0
Device ID		Vendor ID	0x00
Status		Command	0x04
Class Code			Revision ID (Rev ID)	0x08
Not Supported	Header Type	Latency Timer	Cache Line Size	0x0C
Base Address Register Zero (I/O)				0x10
Base Address Register One (MEMORY) bits [31:0]				0x14
Base Address Register Two (SCRIPTS RAM)				0x18
Not Supported				0x1C
Not Supported				0x20
Not Supported				0x24
Reserved				0x28
Subsystem ID		Subsystem Vendor ID		0x2C
Expansion ROM Base Address				0x30
Reserved			Capabilities Pointer	0x34
Reserved				0x38
Max_Lat	Min_Gnt	Interrupt Pin	Interrupt Line	0x3C
Power Management Capabilities (PMC)		Next Item Pointer	Capability ID	0x40
Data	Bridge Support Extensions (PMCSR_BSE)	Power Management Control/Status (PMCSR)		0x44
Subsystem ID Access				0x48

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	0	1	0	0	1	0

DID **Device ID** **[15:0]**

This 16-bit register identifies the particular device. The LSI53C895A Device ID is 0x0012.

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
x	x	x	x	x	x	x	0	x	0	x	0	x	0	0	0

The 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 LSI53C895A is logically disconnected from the PCI bus for all accesses except configuration accesses.

R	**Reserved**	**[15:9]**
SE	**SERR/ Enable** This bit 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** This bit allows the LSI53C895A to detect parity errors on the PCI bus and report these errors to the system. Only data parity checking is enabled and disabled with this bit. The LSI53C895A always generates parity for the PCI bus.	**6**
R	**Reserved**	**5**
WIE	**Write and Invalidate Enable** This bit allows the LSI53C895A to generate write and invalidate commands on the PCI bus. The WIE bit in the DMA Control (DCNTL) register must also be set for the device to generate Write and Invalidate commands.	**4**
R	**Reserved**	**3**
EBM	**Enable Bus Mastering** This bit controls the ability of the LSI53C895A 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 LSI53C895A to behave as a bus master. The device must be a bus master in order to fetch SCRIPTS instructions and transfer data.	**2**
EMS	**Enable Memory Space** This bit controls the ability of the LSI53C895A to respond to Memory space accesses. A value of zero disables the device response. A value of one allows the LSI53C895A to respond to Memory Space accesses at the address range specified by Base Address Register One (MEMORY) and Base Address Register Two (SCRIPTS RAM) registers in the PCI configuration space.	**1**

EIS **Enable I/O Space** **0**
This bit controls the LSI53C895A response to I/O space accesses. A value of zero disables the device response. A value of one allows the LSI53C895A to respond to I/O Space accesses at the address range specified by the Base Address Register Zero (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	2	1	0
DPE	SSE	RMA	RTA	R	DT[1:0]		DPR	R			NC	R			
0	0	0	0	x	0	1	0	x	x	x	1	x	x	x	x

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 instance, 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 LSI53C895A whenever it detects 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 whenever 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/. These are encoded as:

0b00	fast
0b01	medium
0b10	slow
0b11	reserved

These bits are read only and should indicate the slowest time that a device asserts DEVSEL/ for any bus command except Configuration Read and Configuration Write. The LSI53C895A supports a value of 0b01.

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

- The bus agent asserted PERR/ itself or observed PERR/ asserted.
- The agent setting this bit acted as the bus master for the operation in which the error occurred.
- The Parity Error Response bit in the Command register is set.

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

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 (Rev ID)
Read Only

7							0
RID							
x	x	x	x	x	x	x	x

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

Registers: 0x09–0x0B

Class Code
Read Only

23																						0
CC																						
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CC **Class Code** **[23:0]**
This 24-bit register is used to identify 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
Read/Write

7							0
CLS							
0	0	0	0	0	0	0	0

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 know when to retry burst accesses at cache line boundaries. These devices can ignore the PCI cache support lines (SDONE and SB0/) when this register is cleared to 0. If this register is programmed to a number which is not a power of 2, the device will not use PCI performance commands to perform data transfers.

Register: 0x0D

Latency Timer
Read/Write

7							0
LT							
0	0	0	0	0	0	0	0

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 LSI53C895A supports 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 LSI53C895A.
Latency = 2 + (Burst Size x (typical wait states + 1))
Values greater than optimum are also acceptable.

Register: 0x0E

Header Type
Read Only

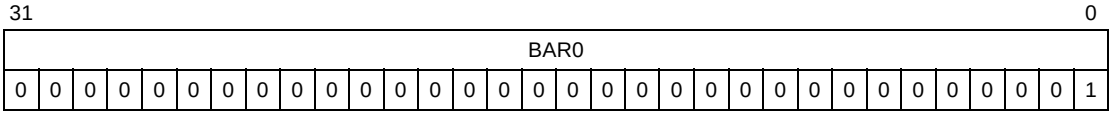
7							0
HT							
0	0	0	0	0	0	0	0

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

This register identifies the layout of bytes 0x10 through 0x3F in configuration space and also whether or not the device contains multiple functions. Since the LSI53C895A is not a multifunction controller the value of this register is 0x00.

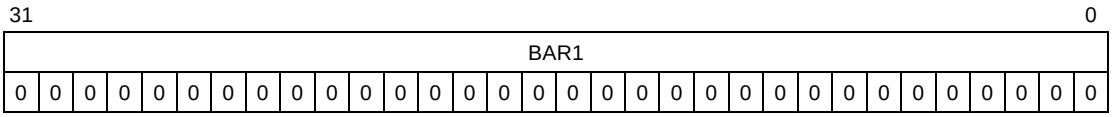
Register: 0x0F
Not Supported

Registers: 0x10–0x13
Base Address Register Zero (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 LSI53C895A requires 256 bytes of I/O space for this base address register. It has bit zero hardwired to one. Bit 1 is reserved and returns a zero on all reads, and the 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 (MEMORY)
Read/Write



BAR1 **Base Address Register One** **[31:0]**
This base address register maps SCSI operating registers into memory space. This device requires 1024 bytes of address space for this base register. This register has bits [9:0] hardwired to 0b0000000000. The default value of this register is 0x00000000. For detailed information on the operation of this register, refer to the PCI 2.2 specification.

Registers: 0x18–0x1B

Base Address Register Two (SCRIPTS RAM)
Read/Write

31																													0
BAR2																													
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

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

This base register is used to map the SCRIPTS RAM into memory space. The default value of this register is 0x00000000. The LSI53C895A points to 8192 bytes of address space with this register. This register has bits [12:0] hardwired to 0b0000000000000. For detailed information on the operation of this register, refer to the PCI 2.2 specification.

Registers: 0x1C–0x27

Not Supported

Registers: 0x28–0x2B

Reserved

Registers: 0x2C–0x2D

Subsystem Vendor ID
Read Only

15															0
SVID															
If MAD[7] HIGH Default															
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
If MAD[7] LOW See Description Default															
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 will contain the value downloaded from the serial EEPROM or a value of 0x0000 if the download fails.

If the external serial EEPROM interface is disabled (MAD[7] HIGH), this register returns a value of 0x1000 (LSI Logic Vendor ID). The 16-bit value that should be stored in the external serial EEPROM for this register is the vendor's PCI Vendor ID and must be obtained from the PCI Special Interest Group (SIG). Please see Section 2.4, "Serial EEPROM Interface," in Chapter 2 for more information on downloading a value for this register.

In addition, if the Subsystem ID Access register (0x48–0x4B) is enabled and unlocked then this register will contain the value that is written into the Subsystem ID Access register. Please see Section 2.5, "Alternative SSVID/SSID Loading Mechanism," in Chapter 2 for additional information.

Registers: 0x2E–0x2F

Subsystem ID

Read Only

15															0
SID															
If MAD[7] HIGH Default															
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
If MAD[7] LOW See Description Default															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

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).

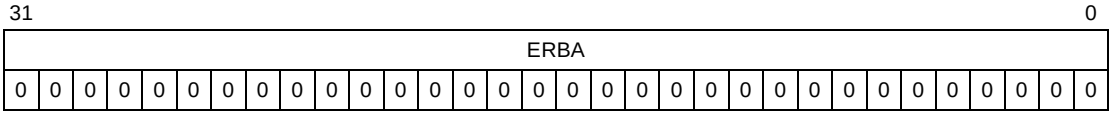
If the external serial EEPROM interface is enabled (MAD[7] is LOW), this register is automatically loaded at power-up from the external serial EEPROM and will contain the value downloaded from the serial EEPROM or a value of 0x0000 if the download fails.

If the external serial EEPROM is disabled (MAD[7] pulled HIGH), the register returns a value of 0x1000. The 16-bit value that should be stored in the external serial EEPROM is vendor specific. Please see the Section 2.4, "Serial EEPROM Interface," in Chapter 2 for additional information on downloading a value for this register.

In addition, if the Subsystem ID Access register (0x48–0x4B) is enabled and unlocked then this register will contain the value that is written into the Subsystem ID Access register. Please see Section 2.5, "Alternative SSVID/SSID Loading Mechanism," in Chapter 2 for additional information.

Registers: 0x30–0x33

Expansion ROM Base Address
Read/Write



ERBA **Expansion ROM Base Address** **[31:0]**

This four-byte register handles the base address and size information for the expansion ROM. It functions exactly like the Base Address Register Zero (I/O) and One (Memory) registers, except that the encoding of the bits is different. The upper 21 bits correspond to the upper 21 bits of the expansion ROM base address.

The expansion ROM Enable bit, bit 0, is the only bit defined in this register. This bit 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. To access the external memory interface, also set the Memory Space bit in the Command register.

The host system detects the size of the external memory by first writing the Expansion ROM Base Address register with all ones and then reading back the register. The LSI53C895A responds with zeros in all don't care locations. The ones in the remaining bits represent 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 the section on MAD Bus Programming for the possible size encodings available.

Register: 0x34

Capabilities Pointer
Read Only

7							0
CP							
0	1	0	0	0	0	0	0

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

Registers: 0x35–0x3B

Reserved

Register: 0x3C

Interrupt Line
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
Read Only

7							0
IP							
0	0	0	0	0	0	0	1

IP **Interrupt Pin** **[7:0]**
This register indicates which interrupt pin the device uses. Its value is set to 0x01 for the INTA/ signal.

Register: 0x3E

Min_Gnt
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 this register is in units of 0.25 microseconds. The LSI53C895A sets this register to 0x11.

Register: 0x3F

Max_Lat
Read Only

7							0
ML							
0	1	0	0	0	0	0	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 LSI53C895A sets this register to 0x40.

Register: 0x40

Capability ID
Read Only

7							0
CID							
0	0	0	0	0	0	0	1

CID **Cap_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
Read Only

7							0
NIP							
0	0	0	0	0	0	0	0

NIP **Next_Item_Ptr** **[7:0]**
Bits [7:0] contain the offset location of the next item in the controller's capabilities list. The LSI53C895A 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	R			DSI	APS	PMEC	VER[2:0]		
0	0	0	0	0	1	1	x	x	x	0	0	0	0	1	0

PMES **PME_Support** **[15:11]**
Bits [15:11] define the power management states in which the LSI53C895A will assert the PME pin. These bits are all set to zero because the LSI53C895A does not provide a PME signal.

D2S **D2_Support** **10**
The LSI53C895A sets this bit to indicate support for power management state D2.

D1S **D1_Support** **9**
The LSI53C895A sets this bit to indicate support for power management state D1.

R **Reserved** **[8:6]**

DSI **Device Specific Initialization** **5**
This bit is cleared to indicate that the LSI53C895A requires no special initialization before the generic class device driver is able to use it.

APS **Auxiliary Power Source** **4**
Because the LSI53C895A does not provide a PME signal, this bit is cleared, indicating that no auxiliary power source is required to support the PME signal in the D3cold power management state.

PMEC **PME Clock** **3**
Bit 3 is cleared because the LSI53C895A does not provide a PME pin.

VER[2:0] **Version** **[2:0]**
These three bits are set to 010 to indicate that the LSI53C895A complies with Revision 1.1 of the PCI Power Management Interface Specification.

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	x	x	x	x	x	x	0	0

PST **PME Status** **15**
The LSI53C895A always returns a zero for this bit, indicating that PME signal generation is not supported from D3cold.

DSCL **Data_Scale** **[14:13]**
The LSI53C895A does not support the data register. Therefore, these two bits are always cleared.

DSLT **Data_Select** **[12:9]**
The LSI53C895A does not support the data register. Therefore, these four bits are always cleared.

PEN **PME_Enable** **8**
The LSI53C895A always returns a 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 LSI53C895A. They are used to place the LSI53C895A in a new power state. Power states are defined as:

0b00	D0
0b01	D1
0b10	D2
0b11	D3hot

See Section 2.6, “Power Management,” in Chapter 2 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 LSI53C895A does not support extensions and 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 LSI53C895A does not use this register and always returns 0x00.

Registers: 0x48–0x4B

Subsystem ID Access
Write Only

31																													0
SIDA																													
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	1

SIDA **Subsystem ID Access** **[31:0]**
The Subsystem ID Access register is the PCI configuration space at offset 0x48–0x4B. Once enabled and unlocked, this register can be used to shadow Subsystem Vendor ID/Subsystem ID data into the appropriate PCI Subsystem registers, thus eliminating the need for a serial EEPROM to store the data. For

additional information about using this register refer to the Section 2.5, “Alternative SSVID/SSID Loading Mechanism,” topic in Chapter 2.

4.2 SCSI Registers

The control registers for the SCSI core are directly accessible from the PCI bus using Memory or I/O mapping. The address map of the SCSI registers is shown in Table 4.2.

Note: The only registers that the host CPU can access while the LSI53C895A is executing SCRIPTS are the Interrupt Status Zero (ISTAT0), Interrupt Status One (ISTAT1) and Mailbox Zero (MBOX0), 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.

Table 4.2 SCSI Register Address Map

31 16	15 0			
SCNTL3	SCNTL2	SCNTL1	SCNTL0	0x00
GPREG0	SDID	SXFER	SCID	0x04
SBCL	SSID	SOCL	SFBR	0x08
SSTAT2	SSTAT1	SSTAT0	DSTAT	0x0C
DSA				0x10
MBOX1	MBOX0	ISTAT1	ISTAT0	0x14
CTEST3	CTEST2	CTEST1	CTEST0	0x18
TEMP				0x1C
CTEST6	CTEST5	CTEST4	DFIFO	0x20
DCMD	DBC			0x24
DNAD				0x28
DSP				0x2C
DSPS				0x30
SCRATCH A				0x34
DCNTL	SBR	DIEN	DMODE	0x38
ADDER				0x3C
SIST1	SIST0	SIEN1	SIEN0	0x40
GPCNTL0	MACNTL	SWIDE	SLPAR	0x44
RESPID1	RESPID0	STIME1	STIME0	0x48
STEST3	STEST2	STEST1	STEST0	0x4C
Reserved	STEST4	SIDL		0x50
CCNTL1	CCNTL0	SODL		0x54
GPREG1	GPCNTL1	SBDL		0x58
SCRATCH B				0x5C
SCRATCH C–SCRATCH R				0x60–0x9F
MMRS				0xA0
MMWS				0xA4
SFS				0xA8
DRS				0xAC
SBMS				0xB0
DBMS				0xB4
DNAD64				0xB8
Reserved				0xBC
PMJAD1				0xC0
PMJAD2				0xC4
RBC				0xC8
UA				0xCC
ESA				0xD0
IA				0xD4
Reserved	SBC			0xD8
CSBC				0xDC
Reserved				0xE0–0xFF

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

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

ARB[1:0] Arbitration Mode Bits 1 and 0 [7:6]

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

Simple Arbitration

1. The LSI53C895A 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 LSI53C895A 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 LSI53C895A wins arbitration.
4. Once the LSI53C895A wins arbitration, SSEL/ must be asserted using the SCSI Output Control Latch (SOCL) for a bus clear plus a bus settle delay (1.2 µs) before a low level selection is performed.

Full Arbitration, Selection/Reselection

1. The LSI53C895A waits for a bus free condition.
2. It asserts SBSY/ and its SCSI ID (the highest priority ID stored in the SCSI Chip ID (SCID) register) onto the SCSI bus.
3. If the SSEL/ signal is asserted by another SCSI device or if the LSI53C895A detects a higher priority ID, the LSI53C895A deasserts SBSY, deasserts its ID, and waits until the next bus free state to try arbitration again.
4. The LSI53C895A 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 LSI53C895A performs selection by asserting the following onto the SCSI bus: SSEL/, the target's ID (stored in the SCSI Destination ID (SDID) register), and the LSI53C895A's ID (stored in the SCSI Chip ID (SCID) register).
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 LSI53C895A 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, indicates that the LSI53C895A 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 SCNTL1 register to verify that the LSI53C895A is not connected to the SCSI bus.

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

When this bit is set and the LSI53C895A 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 LSI53C895A 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 low level mode.

EPC **Enable Parity Checking** **3**

When this bit is set, the SCSI data bus is checked for odd parity when data is received from the SCSI bus in either the initiator or 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.

If the LSI53C895A is operating in the initiator mode and a parity error is detected, assertion of SATN/ is optional, but the transfer continues until the target changes phase. When this bit is cleared, parity errors are not reported.

R **Reserved** **2**

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

When this bit is set, the LSI53C895A automatically asserts the SATN/ signal upon detection of a parity 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 Checking bit for the LSI53C895A to assert SATN/ in this manner. A parity error is detected on data received from the SCSI bus.

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

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

This bit determines the default operating mode of the LSI53C895A. 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 by default. When this bit is cleared, the LSI53C895A is an initiator device by default.

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
EXC	ADB	DHP	CON	RST	AESP	IARB	SST
0	0	0	0	0	0	0	0

EXC **Extra Clock Cycle of Data Setup** **7**

When this bit is set, an extra clock period of data setup is added to each SCSI data transfer. The extra data setup time can provide additional system design margin, though it affects the SCSI transfer rates. Clearing this bit disables the extra clock cycle of data setup time. Setting this bit only affects SCSI send operations.

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

When this bit is set, the LSI53C895A drives the contents of the SCSI Output Data Latch (SODL) register onto the SCSI data bus. When the LSI53C895A is an initiator, the SCSI I/O signal must be inactive to assert the SODL contents onto the SCSI bus. When the LSI53C895A is a target, the SCSI I/O signal must be active to assert the SODL contents onto the SCSI bus. The contents of the SCSI Output Data Latch (SODL) register can be asserted at any time, even before the LSI53C895A is connected to the SCSI bus. Clear this bit when executing SCSI SCRIPTS. It is normally used only for diagnostic testing or operation in low level mode.

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

The DHP bit is only defined for target mode. When this bit is cleared, the LSI53C895A halts the SCSI data transfer when a parity error is detected or when the SATN/ signal is asserted. If SATN/ or a parity error is received in the middle of a data transfer, the LSI53C895A

may transfer up to three additional bytes before halting to synchronize between internal core cells. During synchronous operation, the LSI53C895A transfers data until there are no outstanding synchronous offsets. If the LSI53C895A is receiving data, any data residing in the DMA FIFO is sent to memory before halting.

When this bit is set, the LSI53C895A does not halt the SCSI transfer when SATN/ or a parity error is received.

CON **Connected** **4**

This bit is automatically set any time the LSI53C895A is connected to the SCSI bus as an initiator or as a target. It is set after the LSI53C895A 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 LSI53C895A is not connected to the SCSI bus.

The CPU can force a connected or disconnected condition by setting or clearing this bit. This feature is used primarily during loopback mode.

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 LSI53C895A 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 LSI53C895A checks data received for odd parity. This bit is used for diagnostic testing and is cleared for normal operation. It is useful to generate parity errors to test error handling functions.

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 the SCSI Control Zero (SCNTL0) register are set for full arbitration and selection before setting this bit.

Arbitration is retried until won. At that point, the LSI53C895A 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 Bus Free phase immediately following the arbitration phase, it is possible to perform a low level selection instead.
- The abort completes because the LSI53C895A 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.

SST **Start SCSI Transfer** **0**

This bit is automatically set during SCRIPTS execution and should not be used. It causes the SCSI core to begin a SCSI transfer, including SREQ/ and SACK/ handshaking. The determination of whether the transfer is a send or receive is made according to the value written to the I/O bit in SCSI Output Control Latch (SOCL). This bit is self-clearing. Do not set it for low level operation.

Caution: Writing to this register while not connected may cause the loss of a selection/reselection by clearing the Connected bit.

Register: 0x02

SCSI Control Two (SCNTL2)

Read/Write

7	6	5	4	3	2	1	0
SDU	CHM	SLPMD	SLPHBEN	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 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).

Chained mode is primarily used to transfer consecutive wide data blocks. Using chained mode facilitates partial receive transfers and allows correct partial send behavior. When this bit is set and a data transfer ends on an odd byte boundary, the LSI53C895A 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 is completed.

SLPMD **SLPAR Mode** **5**
If this bit is cleared, the SCSI Longitudinal Parity (SLPAR) register functions as a byte-wide longitudinal parity register. If this bit is set, the SLPAR functions as a word-wide longitudinal parity function. The high or low byte of the SLPAR word is accessible through the SLPAR register. Which byte is accessible is controlled by the SLPHBEN bit.

SLPHBEN **SLPAR High Byte Enable** **4**
If this bit is cleared, the low byte of the SLPAR word is accessible through the SCSI Longitudinal Parity (SLPAR) register. If this bit is set, the high byte of the SLPAR word is present in the SLPAR register.

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.

When the WSS flag is HIGH following a wide SCSI send operation, the SCSI core is holding a byte of “chain” data in the SCSI Output Data Latch (SODL) register. This data becomes the first low-order byte sent when married with a high-order byte during a subsequent data send transfer.

Performing a SCSI receive operation clears this bit. Also, performing any nonwide transfer clears this bit.

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 clearing function is self-clearing.

The WSR flag indicates that the SCSI core received data from the SCSI bus, detected a possible partial transfer at the end of a chained or nonchained block move command, and temporarily stored the high-order byte in the SCSI Wide Residue (SWIDE) register rather than passing the byte out the DMA channel. The hardware uses the WSR status flag to determine what behavior must occur at the start of the next data receive transfer. When the flag is set, the stored data in SWIDE may be “residue” data, valid data for a subsequent data transfer, or overrun data. The byte is read as normal data by starting a data receive transfer.

Performing a SCSI send operation clears this bit. Also, performing any nonwide transfer clears this bit.

Register: 0x03

SCSI Control Three (SCNTL3)

Read/Write

7	6		4	3	2		0
USE	SCF[2:0]			EWS	CCF[2:0]		
0	0	0	0	0	0	0	0

USE **Ultra SCSI Enable** **7**

Setting this bit enables Ultra SCSI or Ultra2 SCSI synchronous transfers. The default value of this bit is 0. This bit should remain cleared if the LSI53C895A is not operating in Ultra SCSI mode or faster.

When this bit is set, the signal filtering period for SREQ/ and SACK/ automatically changes to 8 ns for Ultra2 SCSI or 15 ns for Ultra SCSI, regardless of the value of the Extend REQ/ACK Filtering bit in the SCSI Test Two (STEST2) register.

Note: Set this bit to achieve Ultra SCSI transfer rates in legacy systems that use an 80 MHz clock.

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. Write these to the same value as the Clock Conversion Factor bits below unless fast SCSI operation is desired. See the SCSI Transfer (SXFER) register description for examples of how the SCF bits are used to calculate synchronous transfer periods. See the table under the description of bits [7:5] of the SXFER register for the valid combinations.

EWS **Enable Wide SCSI** **3**
When this bit is clear, 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 done 16 bits at a time, with the least significant byte on SD[7:0]/ and SDP0/ and the most significant byte on SD[15:8]/, SDP1/. Command, Status, and Message phases are not affected by this bit.

CCF[2:0] **Clock Conversion Factor** **[2:0]**
These bits select a factor by which the frequency of SCLK is divided before being presented to the SCSI core. The synchronous portion of the SCSI core can be run at a different clock rate for fast SCSI, using the Synchronous Clock Conversion Factor bits. The bit encoding is displayed in the table below. All other combinations are reserved.

SCF2 CCF2	SCF1 CCF1	SCF0 CCF0	Factor Frequency	SCSI Clock (MHz)
0	0	0	SCLK/3	50.01–75.0
0	0	1	SCLK/1	16.67–25.0
0	1	0	SCLK/1.5	25.01–37.5
0	1	1	SCLK/2	37.51–50.0
1	0	0	SCLK/3	50.01–75.0
1	0	1	SCLK/4	75.01–80.00
1	1	0	SCLK/6	120
1	1	1	SCLK/8	160

Note: It is important that these bits are set to the proper values to guarantee that the LSI53C895A meets the SCSI timings as defined by the ANSI specification.

Register: 0x04

SCSI Chip ID (SCID)

Read/Write

7	6	5	4	3			0
R	RRE	SRE	R	ENC			
x	0	0	x	0	0	0	0

R **Reserved** **7**

RRE **Enable Response to Reselection** **6**

When this bit is set, the LSI53C895A 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 LSI53C895A is able to respond to bus-initiated selection at the chip ID in the RESPID0 and RESPID1 registers. Note that the chip does not automatically reconfigure itself to target mode as a result of being selected.

R **Reserved** **4**

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

These bits are used to store the LSI53C895A encoded SCSI ID. This is the ID which the chip asserts when arbitrating for the SCSI bus. The IDs that the LSI53C895A 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			5	4			0
TP[2:0]			MO[4:0]				
0	0	0	0	0	0	0	0

Note: When using Table Indirect I/O commands, bits [7:0] of this register are loaded from the I/O data structure.

TP[2:0] **SCSI Synchronous Transfer Period [7:5]**
These bits determine the SCSI synchronous transfer period used by the LSI53C895A when sending synchronous SCSI data in either the initiator or target mode. These bits control the programmable dividers in the chip.

TP2	TP1	TP0	XFERP
0	0	0	4
0	0	1	5
0	1	0	6
0	1	1	7
1	0	0	8
1	0	1	9
1	1	0	10
1	1	1	11

The synchronous transfer period the LSI53C895A should use when transferring SCSI data is determined in the following example:

The LSI53C895A is connected to a hard disk which can transfer data at 10 Mbytes/s synchronously. The LSI53C895A's SCLK is running at 40 MHz. The synchronous transfer period (SCSI Transfer (SXFER)) is found as follows:

SXFERP = Period/SSCP + ExtCC
Period = 1 ÷ Frequency = 1 ÷ 10 Mbytes/s = 100 ns
SSCP = 1 ÷ SSCF = 1 ÷ 40 MHz = 25 ns

(This SCSI synchronous core clock is determined in SCNTL3 bits [6:4], ExtCC = 1 if SCNTL1 bit 7 is asserted and the LSI53C895A is sending data. ExtCC = 0 if the LSI53C895A is receiving data.)

SXFERP = 100 ÷ 25 = 4

Where:

SXFERP Synchronous transfer period.

SSCP SCSI synchronous core period.

SSCF SCSI synchronous core frequency.

ExtCC Extra clock cycle of data setup.

Table 4.3 shows examples of synchronous transfer periods and rates for SCSI-1.

Table 4.3 Examples of Synchronous Transfer Periods and Rates for SCSI-1

CLK (MHz)	SCSI CLK ÷ SCNTL3 Bits [6:4]	XFERP (SXFER Bits [7:5])	Synch. Transfer Period (ns)	Synch. Transfer Rate (Mbytes/s)
66.67	3	4	180	5.55
66.67	3	5	225	4.44
50	2	4	160	6.25
50	2	5	200	5
40	2	4	200	5
37.50	1.5	4	160	6.25
33.33	1.5	4	180	5.55
25	1	4	160	6.25
20	1	4	200	5
16.67	1	4	240	4.17

Table 4.4 shows example transfer periods and rates for fast SCSI-2 and Ultra SCSI.

Table 4.4 Example Transfer Periods and Rates for Fast SCSI-2, Ultra, and Ultra2

CLK (MHz)	SCSI CLK ÷ SCNTL3 Bits [6:4]	XFERP	Synch. Transfer Period (ns)	Synch. Transfer Rate (Mbytes/s)
160	1	4	25	40
160	2	4	50	20
160	4	4	100	10
80	1	4	50	20
50	1	4	80	12.5
50	1	5	100	10.0
40	1	4	100	10.0
37.50	1	4	106.67	9.375
33.33	1	4	120	8.33
25	1	4	160	6.25
20	1	4	200	5
16.67	1	4	240	4.17

MO[4:0] **Max SCSI Synchronous Offset [4:0]**

These bits describe the maximum SCSI synchronous offset used by the LSI53C895A when transferring synchronous SCSI data in either the initiator or target mode. Table 4.5 describes the possible combinations and their relationship to the synchronous data offset used by the LSI53C895A. These bits determine the LSI53C895A's method of transfer for Data-In and Data-Out phases only; all other information transfers occur asynchronously.

Table 4.5 Maximum Synchronous Offset

MO4	MO3	MO2	MO1	MO0	Synchronous Offset
0	0	0	0	0	0-Asynchronous
0	0	0	0	1	1
0	0	0	1	0	2
0	0	0	1	1	3
0	0	1	0	0	4
0	0	1	0	1	5
0	0	1	1	0	6
0	0	1	1	1	7
0	1	0	0	0	8
0	1	0	0	1	9
0	1	0	1	0	10
0	1	0	1	1	11
0	1	1	0	0	12
0	1	1	0	1	13
0	1	1	1	0	14
0	1	1	1	1	15
1	0	0	0	0	16
1	0	0	0	1	17
1	0	0	1	0	18
1	0	0	1	1	19
1	0	1	0	0	20
1	0	1	0	1	21
1	0	1	1	0	22
1	0	1	1	1	23
1	1	0	0	0	24
1	1	0	0	1	25
1	1	0	1	0	26
1	1	0	1	1	27
1	1	1	0	0	28
1	1	1	0	1	29
1	1	1	1	0	30
1	1	1	1	1	31

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 set 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 (GPREG0)
Read/Write

7			5	4			0
R			GPIO				
x	x	x	0	x	x	x	x

Reads to this register will always yield the same values. A write to this register will cause the data written to be output to the appropriate GPIO pin if it is set to the output mode in the General Purpose Pin Control Zero (GPCNTL0) register.

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

GPIO **General Purpose I/O** **[4:0]**
These bits are programmed through the General Purpose Pin Control Zero (GPCNTL0) 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. GPIO4 may be used to enable or disable V_{PP}, the 12 Volt power supply to the external flash memory. This bit powers up with the power to external memory disabled. GPIO[3:0] default as inputs and GPIO4 defaults as an output pin. When configured as inputs, an internal pull-down is enabled for GPIO[4:2]. For GPIO[1:0], internal pull-ups are enabled.

LSI Logic 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 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

7							0
1B							
0	0	0	0	0	0	0	0

This register contains the first byte received in any asynchronous information transfer phase. For example, when a LSI53C895A is operating in the initiator mode, this register contains the first byte received in the Message-In, Status phase, 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 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 SFBR as the destination. This allows bit testing after an operation.

The 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 LSI53C895A register (such as a 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 nondeterminate operations may occur. This includes SCRIPTS Read/Write operations and conditional transfer control instructions that initialize the 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

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**

This register is used primarily for diagnostic testing or programmed I/O operation. It is controlled by the SCRIPTS processor when executing SCSI SCRIPTS. SOCL is used only when transferring data using programmed I/O. Some bits are set (1) or cleared (0) when executing SCSI SCRIPTS. Do not write to the register once the LSI53C895A starts executing normal SCSI SCRIPTS.

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 LSI53C895A 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 diagnostic testing or operation in low level mode.

REQ	**SREQ/ Status**	**7**
ACK	**SACK/ Status**	**6**
BSY	**SBSY/ Status**	**5**
SEL	**SSEL/ Status**	**4**
ATN	**SATN/ Status**	**3**
MSG	**SMSG/ Status**	**2**
C_D	**SC_D/ Status**	**1**
I_O	**SI_O/ Status**	**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 LSI53C895A stacks 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 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 LSI53C895A as a master detects a data parity error, or 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 LSI53C895A 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 is 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** **0**

This status bit is set any time an illegal or reserved instruction opcode is detected, whether the LSI53C895A is operating in single step mode or automatically executing SCSI SCRIPTS.

Any of the following conditions during instruction execution also set this bit:

- The LSI53C895A is executing a Wait Disconnect instruction and the SCSI REQ line is asserted without a disconnect occurring.
- A Block Move instruction is executed with 0x000000 loaded into the DMA Byte Counter (DBC) register, indicating there are zero bytes to move.

- 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 LSI53C895A is in target mode.
- 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.
- A Transfer Control instruction is executed with the reserved bit 22 set.
- A Transfer Control instruction is executed with the Wait for Valid phase bit (bit 16) set while the chip is in target mode.
- A Load/Store instruction is issued with the memory address mapped to the operating registers of the chip, not including ROM or RAM.
- A Load/Store instruction is issued when the register address is not aligned with the memory address.
- A Load/Store instruction is issued with bit 5 in the DMA Command (DCMD) register cleared or bits 3 or 2 set.
- A Load/Store instruction when the count value in the DMA Byte Counter (DBC) register is not set at 1 to 4.
- A Load/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 24 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	ORF	OLF	AIP	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) register contains data. Data is transferred from the SCSI bus to the SCSI Input Data Latch register before being sent to the DMA FIFO and then to the host bus. The SIDL register contains SCSI data received asynchronously. Synchronous data received does not flow through this register.

ORF — **SODR Least Significant Byte Full** — **6**

This bit is set when the least significant byte in the SCSI Output Data Register (SODR, a hidden buffer register which is not accessible) contains data. The SODR is used by the SCSI logic as a second storage register when sending data synchronously. It is not readable or writable by the user. It is possible to use this bit to determine how many bytes reside in the chip when an error occurs.

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. In synchronous mode, data is transferred from the host bus to the SODL register, and then to the SCSI Output Data Register (SODR, a hidden buffer register which is not accessible) before being sent to the SCSI bus. In asynchronous mode, data is transferred from the host bus to the SODL register, and then to the SCSI bus. The SODR buffer register is not used for asynchronous transfers. It is possible to use this bit to determine how many bytes reside in the chip when an error occurs.

AIP **Arbitration in Progress** **4**

Arbitration in Progress (AIP = 1) indicates that the LSI53C895A 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 LSI53C895A 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 LSI53C895A 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 SRST signal (bit 6) in the Interrupt Status Zero (ISTAT0) register. This bit is not latched and may change as it is read.

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

This bit represents the active HIGH current status 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
FF[3:0]				SDP0L	MSG	C_D	I_O
0	0	0	0	x	x	x	x

FF[3:0] **FIFO Flags** **[7:4]**

These four bits, along with SCSI Status Two (SSTAT2) bit 4, define the number of bytes or words that currently reside in the LSI53C895A's SCSI synchronous data FIFO as shown in Table 4.6. These bits are not latched and they will change as data moves through the FIFO. The

SCSI FIFO can hold up to 31 bytes for narrow SCSI synchronous data transfers, or up to 31 words for wide. Values over 31 will not occur.

Table 4.6 SCSI Synchronous Data FIFO Word Count

FF4 (SSTAT2 bit 4)	FF3	FF2	FF1	FF0	Bytes or Words in the SCSI FIFO
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	1	0	2
0	0	0	1	1	3
0	0	1	0	0	4
0	0	1	0	1	5
0	0	1	1	0	6
0	0	1	1	1	7
0	1	0	0	0	8
0	1	0	0	1	9
0	1	0	1	0	10
0	1	0	1	1	11
0	1	1	0	0	12
0	1	1	0	1	13
0	1	1	1	0	14
0	1	1	1	1	15
1	0	0	0	0	16
1	0	0	0	1	17
1	0	0	1	0	18
1	0	0	1	1	19
1	0	1	0	0	20
1	0	1	0	1	21
1	0	1	1	0	22
1	0	1	1	1	23
1	1	0	0	0	24
1	1	0	0	1	25
1	1	0	1	0	26
1	1	0	1	1	27
1	1	1	0	0	28
1	1	1	0	1	29
1	1	1	1	0	30
1	1	1	1	1	31

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**

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

I_O **SCSI I_O/ Signal** **0**
These three SCSI phase status bits (MSG, C_D, and I_O) are latched on the asserting edge of SREQ/ when operating in either the initiator or target mode. These bits are set when the corresponding signal is active. They are 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	ORF1	OLF1	FF4	SPL1	DM	LDSC	SDP1
0	0	0	0	x	x	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 and then to the host bus. The SIDL register contains SCSI data received asynchronously. Synchronous data received does not flow through this register.

ORF1 **SODR Most Significant Byte Full** **6**
This bit is set when the most significant byte in the SCSI Output Data register (SODR, a hidden buffer register which is not accessible) contains data. The SODR register is used by the SCSI logic as a second storage register when sending data synchronously. It is not accessible to the user. This bit is used to determine how many bytes reside in the chip when an error occurs.

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. In synchronous mode, data is transferred from the host bus to the SODL register, and then to the SCSI Output Data register (SODR, a hidden buffer register which is not accessible) before being sent to the SCSI bus. In asynchronous mode, data is transferred from the host bus to the SODL register, and then to the SCSI bus. The SODR buffer register is not used for asynchronous transfers. It is possible to use this bit to determine how many bytes reside in the chip when an error occurs.

FF4 **FIFO Flags, Bit 4** **4**

This is the most significant bit in the SCSI FIFO Flags field, when concatenated with bits [7:4] (FF[3:0]) in SCSI Status One (SSTAT1). For a complete description of this field, see the definition for SCSI Status One (SSTAT1) bits [7:4].

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.

DM **Diffsens Mismatch** **2**

This bit is set when the DIFFSENS pin detects a SE or LVD SCSI operating voltage level while the LSI53C895A is operating in HVD mode (by setting the DIF bit in the SCSI Test Two (STEST2) register). If this bit is cleared, the DIFFSENS value matches the DIF bit setting.

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 some SCSI device selects or reselects the LSI53C895A. If the Connected bit is asserted and the LDSC bit is 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 on.

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

This bit represents the active HIGH current state of the SCSI SDP1 parity signal. It is unlatched and may change as it is read.

Registers: 0x10–0x13

Data Structure Address (DSA)

Read/Write

31																													0
DSA																													
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

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 register is accessible by the host CPU while a LSI53C895A 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 LSI53C895A. 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) and Interrupt Status One (ISTAT1) registers.
4. If the SCSI Interrupt Pending bit is set, then read the SCSI Interrupt Status Zero (SIST0) or SCSI Interrupt Status One (SIST1) register to determine the cause of the SCSI Interrupt and go back to Step 2.
5. If the SCSI Interrupt Pending bit is clear, and the DMA Interrupt Pending bit is set, then write 0x00 value to this register.
6. Read the DMA Status (DSTAT) register to verify the aborted interrupt and to see if any other interrupting conditions have occurred.

SRST **Software Reset** **6**

Setting this bit resets the LSI53C895A. 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).

SIGP **Signal Process** **5**

SIGP is a R/W bit that is writable at any time, and 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 this bit causes that 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 LSI53C895A is executing a SCRIPTS operation. This bit enables the LSI53C895A to notify an external processor of a predefined condition while SCRIPTS are running. The external processor may also notify the LSI53C895A 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 LSI53C895A is connected to the SCSI bus as an initiator or as a target. It is set after successfully completing selection or when the LSI53C895A responds to a bus-initiated selection or reselection. It is also set after the LSI53C895A wins arbitration when operating in low level mode. When this bit is clear, the LSI53C895A 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) or Interrupt Status One (ISTAT1) registers are read they are not automatically cleared. To clear this bit, write it to a one. The reset operation is self-clearing.

Note: If the INTF bit is set but SIP or DIP are not set, do not attempt to read the other chip status registers. An Interrupt-on-the-Fly interrupt must be cleared before servicing any other interrupts indicated by SIP or DIP.

This bit must be written to one in order to clear it after it has been set.

SIP **SCSI Interrupt Pending** **1**

This status bit is set when an interrupt condition is detected in the SCSI portion of the LSI53C895A. 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 LSI53C895A is selected
- The LSI53C895A 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 is expired
- The general purpose timer is expired
- A SCSI bus mode change is detected

To determine exactly 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 LSI53C895A. 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 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
x	x	x	x	x		0	0	0

R **Reserved** **[7:3]**

FLSH **Flushing** **2**
Reading this bit monitors if the chip is currently flushing data. If set, the chip is flushing data from the DMA FIFO. If cleared, no flushing is occurring. This bit is read only and writes will have no effect on the value of this bit.

SRUN **SCRIPTS Running** **1**
This bit indicates whether or not the SCRIPTS engine is currently fetching and executing SCRIPTS instructions. If this bit is set, the SCRIPTS engine is active.

If it is cleared, the SCRIPTS engine is not active.

This bit is read only and writes will have no effect on the value of this bit.

SI **SYNC_IRQD** **0**
Setting this bit disables the IRQ/ pin. Clearing this bit enables normal operation of the IRQ/ pin. The function of this bit is nearly identical to bit 1 of the DMA Control (DCNTL) (0x3B) register except that if the IRQ/ is already asserted and this bit is set, IRQ/ will remain asserted until the interrupt is serviced. At this point the IRQ/ line will be 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 potentially attempt to access the same mailbox byte at the same time. Using one mailbox register as a read only and the other as a write only will prevent this type of 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 potentially attempt to access the same mailbox byte at the same time. Using one mailbox register as a read only and the other as a write only will prevent this type of 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 bottom bytes in the DMA FIFO that are empty. Each bit corresponds to a byte lane in the DMA FIFO. For example, if byte lane three is empty, then FMT3 will be set. Since the FMT flags indicate the status of bytes at the bottom of the FIFO, 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 top bytes in the DMA FIFO that are full. Each bit corresponds to a byte lane in the DMA FIFO. For example, if byte lane three is full then FFL3 is set. Since the FFL flags indicate the status of bytes at the top of the FIFO, 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	1	0
DDIR	SIGP	CIO	CM	PCICIE	TEOP	DREQ	DACK
0	0	x	x	0	0	0	1

DDIR **Data Transfer Direction** **7**

This status bit indicates which direction data is being transferred. When this bit is set, the data is transferred from the SCSI bus to the host bus. When this bit is clear, the data is transferred from the host bus to the SCSI bus.

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 Into Enable** **3**

This bit controls the shadowing of the PCI Base Address Register One (MEMORY), PCI Base Address Register Two (SCRIPTS RAM), PCI Device ID, and PCI Revision ID into the Scratch Register A (SCRATCHA), Scratch Register B (SCRATCHB), and SCRIPTS Fetch Selector (SFS) registers.

When it is set, the SCRATCHA register contains bits [31:0] of the Memory Base Address value from the PCI Base Address Register One (MEMORY). This is the memory mapped operating register base address. Bits [9:0] will be 0. The SCRATCHB register contains bits [31:13] of the RAM Base Address value from the PCI Base Address Register Two (SCRIPTS RAM). This is the base address for the internal 8 Kbytes RAM. Bits [12:0] will be 0. Bits [23:16] of SCRIPTS Fetch Selector (SFS) contain the PCI Revision ID (Rev ID) register value and bits [15:0] contain the PCI Device ID register value. When this bit is set, writes to this register have no effect.

When this bit is cleared, the Scratch Register A (SCRATCHA), Scratch Register B (SCRATCHB), and SCRIPTS Fetch Selector (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.

TEOP **SCSI True End of Process** **2**

This bit indicates the status of the LSI53C895A's TEOP signal. The TEOP signal acknowledges the completion of a transfer through the SCSI portion of the LSI53C895A. When this bit is set, TEOP is active. When this bit is clear, TEOP is inactive.

DREQ **Data Request Status** **1**

This bit indicates the status of the LSI53C895A's internal Data Request signal (DREQ). When this bit is set, DREQ is active. When this bit is clear, DREQ is inactive.

DACK **Data Acknowledge Status** **0**

This bit indicates the status of the LSI53C895A's internal Data Acknowledge signal (DACK/). When this bit is set, DACK/ is inactive. When this bit is clear, DACK/ is active.

Register: 0x1B

Chip Test Three (CTEST3)

Read/Write

7			4	3	2	1	0
V				FLF	CLF	FM	WRIE
x	x	x	x	0	0	0	0

V **Chip Revision Level** **[7:4]**

These bits identify the chip revision level for software purposes. It should have the same value as the lower nibble of the PCI Revision ID (Rev ID) register, at address 0x08 in the configuration space.

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 64 (DNAD64) 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 LSI53C895A.

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 LSI53C895A 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.

FM **Fetch Pin Mode** **1**

When set, this bit causes the FETCH/ pin to deassert during indirect and table indirect read operations. FETCH/ is only active during the opcode portion of an instruction fetch. This allows the storage of SCRIPTS in a PROM while data tables are stored in RAM.

If this bit is not set, FETCH/ is asserted for all bus cycles during instruction fetches.

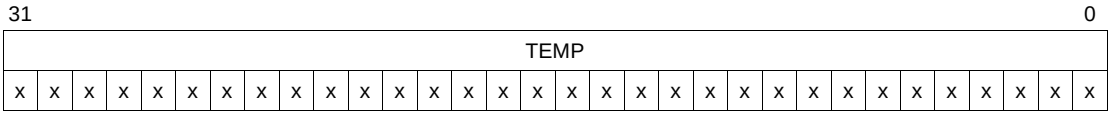
WRIE **Write and Invalidate Enable** **0**

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 in order for the chip to generate Write and Invalidate commands.

Registers: 0x1C–0x1F

Temporary (TEMP)

Read/Write



TEMP **Temporary** **[31:0]**

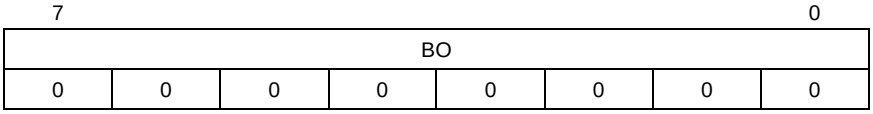
This 32-bit register stores the Return instruction address pointer from the Call instruction. The address pointer stored in this register is loaded into the DMA SCRIPTS Pointer (DSP) register when a Return instruction is executed. This address points to the next instruction to execute. Do not write to this register while the LSI53C895A 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

DMA FIFO (DFIFO)

Read/Write



BO **Byte Offset Counter** **[7:0]**

These bits, along with bits [1:0] in the Chip Test Five (CTEST5) register, indicate the amount of data transferred between the SCSI core and the DMA core. It is used to determine the number of bytes in the DMA FIFO when an interrupt occurs. These bits are unstable while data is being transferred between the two cores.

Once the chip has stopped transferring data, these bits are stable.

The DMA FIFO (DFIFO) register counts the number of bytes transferred between the DMA core and the SCSI core. The DMA Byte Counter (DBC) register counts the number of bytes transferred across the host bus. The difference between these two counters represents the number of bytes remaining in the DMA FIFO.

The following steps determine how many bytes are left in the DMA FIFO when an error occurs, regardless of the transfer direction:

If the DFS bit (bit 5, Chip Test Five (CTEST5)) is set:

1. Subtract the ten least significant bits of the DMA Byte Counter (DBC) register from the 10-bit value of the DFBOC. The DFBOC consists of the Chip Test Five (CTEST5) register, bits 1 and 0 and the DMA FIFO (DFIFO) register, bits [7:0].
2. AND the result with 0x3FF for a byte count between zero and 944.

If the DFS bit (bit 5, Chip Test Five (CTEST5)) is cleared:

1. Subtract the seven least significant bits of the DMA Byte Counter (DBC) register from the seven bit value of the DFBOC which is made up of the DMA FIFO (DFIFO) register, bits [6:0].
2. AND the result with 0x7F for a byte count between zero and 112.

Note: If trying to calculate the total number of bytes in both the DMA FIFO and SCSI Logic, see Section 2.2.12.1, “Data Paths,” in Chapter 2, “Functional Description.”

Register: 0x21

Chip Test Four (CTEST4)

Read/Write

7	6	5	4	3	2		0
BDIS	FBL3	ZSD	SRTM	MPEE	FBL[2:0]		
0	0	0	0	0	0	0	0

BDIS **Burst Disable** **7**

When set, this bit causes the LSI53C895A to perform back-to-back cycles for all transfers. When this bit is cleared, back-to-back transfers for opcode fetches and burst transfers for data moves are performed.

FBL3 **FIFO Byte Control** **6**

This bit is used with FBL[2:0]. See Bits [2:0] description in this register.

ZSD **SCSI Data High Impedance** **5**

Setting this bit causes the LSI53C895A to place the SCSI data bus SD[15:0] and the parity lines SDP[1:0] in a high impedance state. In order to transfer data on the SCSI bus, clear this bit.

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 DSA and TEMP registers contain the base address used for table indirect calculations, and the address pointer for a call or return instruction, respectively. This bit is intended for manufacturing diagnostics only and should not be set during normal operations.

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 LSI53C895A. A parity error during a bus master write is detected by the target, and the

LSI53C895A is informed of the error by the PERR/ pin being asserted by the target. When this bit is cleared, the LSI53C895A 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
0	x	x	x	Disabled
1	0	0	0	0
1	0	0	1	1
1	0	1	0	2
1	0	1	1	3
1	1	0	0	4
1	1	0	1	5
1	1	1	0	6
1	1	1	1	7

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	4	3	2	1	0
ADCK	BBCK	DFS	MASR	DDIR	BL2	BO[9:8]	
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 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 DBC register. It is decremented based on the DMA Byte Counter (DBC) contents and the current DMA Next Address (DNAD) value. This bit automatically clears itself after decrementing the DBC register.

DFS **DMA FIFO Size** **5**

This bit controls the size of the DMA FIFO. When clear, the DMA FIFO appears as only 112 bytes deep. When set, the DMA FIFO size increases to 944 bytes. Using an 112-byte FIFO allows software written for other LSI53C8XX family chips to properly calculate the number of bytes residing in the chip after a target disconnect. The default value of this bit is zero.

MASR **Master Control for Set or Reset Pulses** **4**

This bit controls the operation of bit 3. When this bit is set, bit 3 asserts the corresponding signals. When this bit is cleared, bit 3 deasserts the corresponding signals. Do not change this bit and bit 3 in the same write cycle.

DDIR **DMA Direction** **3**

Setting this bit either asserts or deasserts the internal DMA Write (DMAWR) direction signal depending on the current status of the MASR bit in this register. Asserting the internal DMA write signal indicates that data is

transferred from the SCSI bus to the host bus. Deasserting the internal DMA write signal transfers data from the host bus to the SCSI bus.

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. This bit is disabled if a 112-byte FIFO is selected by clearing the DMA FIFO size bit.

BO[9:8] **DMA FIFO Byte Offset Counter, Bits [9:8]** **[1:0]**

These are the upper two bits of the DFBOC. The DFBOC consists of these bits, and the DMA FIFO (DFIFO) register, bits [7:0].

Register: 0x23

Chip Test Six (CTEST6)

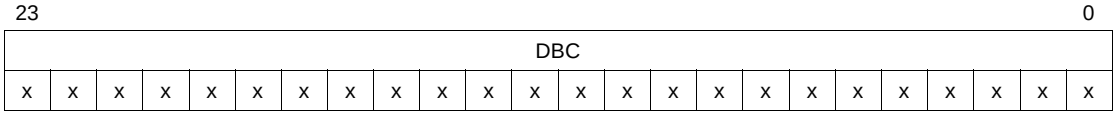
Read/Write

7							0
DF							
0	0	0	0	0	0	0	0

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 operation. Write 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 LSI53C895A. 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 that are transferred.

The maximum number of bytes that can be transferred in any one Block Move command is 16,777,215 bytes. The maximum value that can be loaded into the 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 LSI53C895A is not in target mode, Command phase.

The 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	1	x	x	x	x	x	x

DCMD **DMA Command** **[7:0]**
This 8-bit register determines the instruction for the LSI53C895A 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

31																													0
DSP																													
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

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

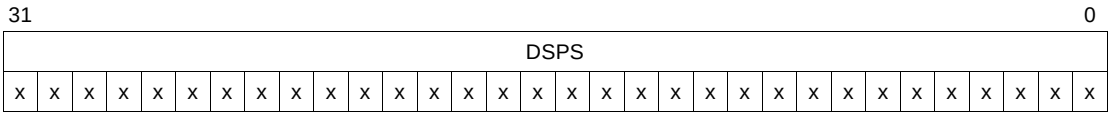
the SCRIPT is written to this register, SCRIPTS are automatically fetched and executed until an interrupt condition occurs.

In 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, but the Start DMA bit (bit 2, DMA Control (DCNTL) register) must be set each time the step interrupt occurs to fetch and execute the next SCRIPTS command. 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																													
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

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 Into Enable bit in the Chip Test Two (CTEST2) register. If this bit is set, the SCRATCH A register returns bits [31:10] of the Memory Mapped Operating register PCI base address (Base Address Register One (MEMORY)) in bits [31:10] of the Scratch Register A (SCRATCHA) when read. Bits [9:0] of SCRATCH A will always return zero in this mode. Writes to the SCRATCH A register are unaffected. Clearing the PCI Configuration Into 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. The LSI53C895A asserts the Bus Request (REQ/) output when the DMA FIFO can accommodate a transfer of at least one burst threshold of data. Bus Request (REQ/) 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 LSI53C895A inserts a “fairness delay” of four CLKs between burst transfers (as 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.

The LSI53C895A will only support burst thresholds of up to 16 Dwords in the small FIFO mode. Setting the burst threshold to higher than 16 Dwords in the small FIFO mode will yield unexpected results in burst lengths. The big FIFO mode is activated by setting bit 5 of the Chip Test Five (CTEST5) register. In the big FIFO mode, the LSI53C895A will support burst thresholds of up to 128 Dwords.

BL2 (CTEST5 bit 2)	BL1	BL0	Burst Length Transfers	Dwords
0	0	0	2	4
0	0	1	4	8
0	1	0	8	16
0	1	1	16	32[1]
1	0	0	32	64[1]
1	0	1	64	128[1]
1	1	0	64	128[1]
1	1	1	Reserved	Reserved

1. The 944-byte FIFO must be enabled for these burst sizes.

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; and if cleared, then the source address is in memory space.

This function is useful for register-to-memory operations using the Memory Move instruction when the LSI53C895A is I/O mapped. Bits 4 and 5 of the Chip Test Two (CTEST2) register are used to determine the configuration status of the LSI53C895A.

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; and if cleared, then the destination address is in memory space.

This function is useful for memory-to-register operations using the Memory Move instruction when the LSI53C895A is I/O mapped. Bits 4 and 5 of the Chip Test Two (CTEST2) register are used to determine the configuration status of the LSI53C895A.

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, then 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 LSI53C895A 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 effect. This bit also has no effect on fetches out of SCRIPTS RAM.

MAN **Manual Start Mode** **0**

Setting this bit prevents the LSI53C895A 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 LSI53C895A to

automatically begin fetching and executing SCSI SCRIPTS when the 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

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**

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 IRQ/ from being asserted for the corresponding interrupt, but the status bit is still set in the DSTAT register. Masking an interrupt does not prevent setting the Interrupt Status Zero (ISTAT0) DIP. All DMA interrupts are considered fatal, therefore SCRIPTS stops running when this condition occurs, whether or not the interrupt is masked. Setting a mask bit enables the assertion of IRQ/ 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 IRQ/ output is latched. Once asserted, it will remain asserted until the interrupt is cleared by reading the appropriate status register. Masking an interrupt after the IRQ/ output is asserted does not cause deassertion of IRQ/.

For more information on interrupts, see Chapter 2, “Functional Description.”

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 LSI53C8XX family products.

Register: 0x3B

DMA Control (DCNTL)
Read/Write

7	6	5	4	3	2	1	0
CLSE	PFF	PFEN	SSM	IRQM	STD	IRQD	COM
0	0	0	0	0	0	0	0

CLSE **Cache Line Size Enable** **7**
Setting this bit enables the LSI53C895A to sense and react to cache line boundaries set up by the DMA Mode (DMODE) or PCI Cache Line Size register, whichever contains the smaller value. Clearing this bit disables the cache line size logic and the LSI53C895A monitors the cache line size using the DMODE register.

PFF **Prefetch Flush** **6**
Setting this bit causes the prefetch unit to flush its contents. The 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, will fetch 8 Dwords of instructions and instruction operands in bursts of 4 or 8 Dwords. Prefetching instructions allows the LSI53C895A to make more efficient use of the system PCI bus, thus improving overall system performance. The unit will flush whenever the PFF bit is set, as well as on all transfer control instructions when the transfer conditions are met, on every write to the DMA SCRIPTS Pointer (DSP), on every regular MMOV instruction, and when any interrupt is generated. The unit automatically determines the maximum burst size that it is capable of performing based on the burst length as determined by the values in the DMA Mode (DMODE) register. If the burst threshold is set to 8 Dwords the prefetch unit will fetch instructions in two bursts of 4 Dwords. If the burst threshold is set to 16 Dwords or greater the prefetch unit will fetch instructions in one burst of 8 Dwords. Burst thresholds of less than 8 Dwords will cause the prefetch unit to be disabled. PCI Cache commands (Read Line and Read Multiple) will be issued appropriately 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 will have no effect on the fetch mechanism from SCRIPTS RAM.

SSM **Single Step Mode** **4**

Setting this bit causes the LSI53C895A to stop after executing each SCRIPTS instruction, and generate a single step interrupt. When this bit is cleared the LSI53C895A 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 clear. To restart the LSI53C895A after it generates a SCRIPTS Step interrupt, read the Interrupt Status Zero (ISTAT0), Interrupt Status One (ISTAT1), 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 IRQ/ pin. When cleared, this bit enables an open drain driver for the IRQ/ pin with an internal weak pull-up. The bit should remain cleared to retain full PCI compliance.

STD **Start DMA Operation** **2**

The LSI53C895A 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 LSI53C895A 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 LSI53C895A 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 LSI53C895A is in single step mode, set the Start DMA bit to restart execution of SCRIPTS after a single step interrupt.

IRQD **IRQ Disable** **1**

Setting this bit disables the IRQ pin. Clearing the bit enables normal operation. As with any other register other than Interrupt Status Zero (ISTAT0) and Interrupt Status One (ISTAT1), this register cannot be accessed except by a SCRIPTS instruction during SCRIPTS execution. For more information on the use of this bit in interrupt handling, see Chapter 2, “Functional Description.”

COM **LSI53C700 Family Compatibility** **0**

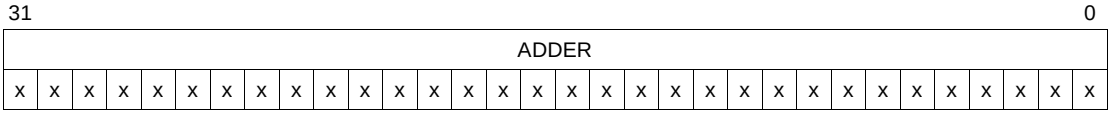
When the COM bit is cleared, the LSI53C895A behaves in a manner compatible with the LSI53C700 family; selection/reselection IDs are stored in both the SCSI Selector ID (SSID) and SCSI First Byte Received (SFBR) registers. This bit is not affected by a software reset.

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

When the COM bit is set, the ID is stored only in the SSID register, protecting the 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



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. It is used to determine if the correct memory address was calculated for a relative jump SCRIPTS instruction.

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 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**
Indicates full arbitration and selection sequence is completed.

SEL **Selected** **5**
Indicates the LSI53C895A is selected by a SCSI initiator device. Set the Enable Response to Selection bit in the SCSI Chip ID (SCID) register for this to occur.

RSL **Reselected** **4**
Indicates the LSI53C895A is reselected by a SCSI target device. Set the Enable Response to Reselection bit in the SCSI Chip ID (SCID) register for this to occur.

SGE **SCSI Gross Error** **3**
The following conditions are considered SCSI Gross Errors:

- Data underflow – reading the SCSI FIFO when no data is present.
- Data overflow – writing to the SCSI FIFO while it is full.
- Offset underflow – receiving a SACK/ pulse in target mode before the corresponding SREQ/ is sent.
- Offset overflow – receiving an SREQ/ pulse in the initiator mode, and exceeding the maximum offset (defined by the MO[3:0] bits in the SCSI Transfer (SXFER) register).
- A phase change in the initiator mode, with an outstanding SREQ/SACK/ offset.
- Residual data in SCSI FIFO – starting a transfer other than synchronous data receive with data left in the SCSI synchronous receive FIFO.

UDC **Unexpected Disconnect** **2**
This condition only occurs in the initiator mode. It happens when the target to which the LSI53C895A is connected disconnects from the SCSI bus unexpectedly. See the SCSI Disconnect Unexpected bit in the SCSI

Control Two (SCNTL2) register for more information on expected versus unexpected disconnects. Any disconnect in low level mode causes this condition.

RST **SCSI Reset Condition** **1**
Indicates assertion of the SRST/ signal by the LSI53C895A or any other SCSI device. This condition is edge-triggered, so multiple interrupts cannot occur because of a single SRST/ pulse.

PAR **SCSI Parity Error** **0**
Indicates detection by the LSI53C895A of a parity error while receiving or sending SCSI data. See the Disable Halt on Parity Error or SATN/ Condition bits in the SCSI Control One (SCNTL1) register for more information on when this condition is actually 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** **4**
Setting this bit allows the LSI53C895A 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.

R **Reserved** **3**

STO **Selection or Reselection Time-out** **2**
The SCSI device which the LSI53C895A 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**
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.

HTH **Handshake-to-Handshake Timer Expired** **0**
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.

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 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 LSI53C895A stacks 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 CLK 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 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 LSI53C895A is selected by another SCSI device. The Enable Response to Selection bit must be set in the SCSI Chip ID (SCID) register (and the Response ID Zero (RESPID0) and Response ID One (RESPID1) register must hold the chip's ID) for the LSI53C895A to respond to selection attempts.

RSL — **Reselected** — **4**

This bit is set when the LSI53C895A is reselected by another SCSI device. The Enable Response to Reselection bit must be set in the SCID register (and the Response ID Zero (RESPID0) and Response ID One (RESPID1) registers must hold the chip's ID) for the LSI53C895A to respond to reselection attempts.

SGE — **SCSI Gross Error** — **3**

This bit is set when the LSI53C895A encounters a SCSI Gross Error Condition. The following conditions can result in a SCSI Gross Error Condition:

- Data Underflow – reading the SCSI FIFO when no data is present.

- Data Overflow – writing too many bytes to the SCSI FIFO, or the synchronous offset causes overwriting the SCSI FIFO.
- Offset Underflow – the LSI53C895A is operating in target mode and a SACK/ pulse is received when the outstanding offset is zero.
- Offset Overflow – the other SCSI device sends a SREQ/ or SACK/ pulse with data which exceeds the maximum synchronous offset defined by the SCSI Transfer (SXFER) register.
- A phase change occurs with an outstanding synchronous offset when the LSI53C895A is operating as an initiator.
- Residual data in the synchronous data FIFO – a transfer other than synchronous data receive is started with data left in the synchronous data FIFO.

UDC **Unexpected Disconnect** **2**

This bit is set when the LSI53C895A is operating in the initiator mode and the target device unexpectedly disconnects from the SCSI bus. This bit is only valid when the LSI53C895A operates in the initiator mode. When the LSI53C895A operates in low level mode, any disconnect causes an interrupt, even a valid SCSI disconnect. This bit is also set if a selection time-out occurs (it may occur before, at the same time, or stacked after the STO interrupt, since this is not considered an expected disconnect).

RST **SCSI RST/ Received** **1**

This bit is set when the LSI53C895A detects an active SRST/ signal, whether the reset is generated external to the chip or caused by the Assert RST 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 Error** **0**

This bit is set when the LSI53C895A detects a parity error while receiving SCSI data. The Enable Parity Checking bit (bit 3 in the SCSI Control Zero (SCNTL0)

register) must be set for this bit to become active. The LSI53C895A always generates parity when sending SCSI data.

Register: 0x43

SCSI Interrupt Status One (SIST1)
Read Only

7		5	4	3	2	1	0
R			SBMC	R	STO	GEN	HTH
x	x	x	0	x	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 that is 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 that indicates the SCSI bus has switched between SE, LVD or HVD modes.

R **Reserved** **3**

STO **Selection or Reselection Time-out** **2**
The SCSI device which the LSI53C895A 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

SCSI Longitudinal Parity (SLPAR)

Read/Write

7							0
SLPAR							
x	x	x	x	x	x	x	x

SLPAR **SCSI Longitudinal Parity** **[7:0]**

This register performs a bytewise longitudinal parity check on all SCSI data received or sent through the SCSI core. If one of the bytes received or sent (usually the last) is the set of correct even parity bits, SLPAR should go to zero (assuming it started at zero). As an example, suppose that the following three data bytes and one check byte are received from the SCSI bus (all signals are shown active HIGH):

Data Bytes	Running SLPAR
–	00000000
1. 11001100	11001100 (XOR of word 1)
2. 01010101	10011001 (XOR of word 1 and 2)
3. 00001111	10010110 (XOR of word 1, 2 and 3) Even Parity
4. 10010110	00000000

A one in any bit position of the final SLPAR value would indicate a transmission error.

The SLPAR register is also used to generate the check bytes for SCSI send operations. If the SLPAR register contains all zeros prior to sending a block move, it

contains the appropriate check byte at the end of the block move. This byte must then be sent across the SCSI bus.

Note: Writing any value to this register clears it to zero.

The longitudinal parity checks are meant to provide an added measure of SCSI data integrity and are entirely optional. This register does not latch SCSI selection/reselection IDs under any circumstances. The default value of this register is zero.

The longitudinal parity function normally operates as a byte function. During 16-bit transfers, the high and low bytes are XORed together and then XORed into the current longitudinal parity value. By setting the SLPMD bit in the SCSI Control Two (SCNTL2) register, the longitudinal parity function is made to operate as a word-wide function. During 16-bit transfers, the high byte of the SCSI bus is XORed with the high byte of the current longitudinal parity value, and the low byte of the SCSI bus is XORed with the low byte of the current longitudinal parity value. In this mode, the 16-bit longitudinal parity value is accessed a byte at a time through the SCSI Longitudinal Parity (SLPAR) register. Which byte is accessed is controlled by the SLPHBEN bit in the SCSI Control Two (SCNTL2) register.

Register: 0x45

SCSI Wide Residue (SWIDE)

Read/Write

7							0
SWIDE							
x	x	x	x	x	x	x	x

SWIDE **SCSI Wide Residue** **[7:0]**

After a 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, or it is a residue byte which should be cleared when an Ignore

Wide Residue message is received. It may also be an overrun data byte. The power-up value of this register is indeterminate.

Register: 0x46

Memory Access Control (MACNTL)
Read/Write

7			4	3	2	1	0
TYP				DWR	DRD	PSCPT	SCPTS
1	1	1	1	0	0	0	0

TYP **Chip Type** **[7:4]**
These bits identify the chip type for software purposes.

Note: These bits no longer identify an 8XX device. These bits have been set to 0xF to indicate that the device should be uniquely identified by setting the PCI Configuration Enable bit in the Chip Test Two (CTEST2) register and using the PCI Revision ID and PCI Device ID which will be shadowed in the SCRIPTS Fetch Selector (SFS) register. Any devices that contain the value 0xF in this register should use this mechanism to uniquely identify the device.

DWR **Data Write** **3**
This bit is used to define if a data write is considered to be a local memory access.

DRD **Data Read** **2**
This bit is used to define if a data read is considered to be a local memory access.

PSCPT **Pointer SCRIPTS** **1**
This bit is used to define if a pointer to a SCRIPTS indirect or table indirect fetch is considered to be a local memory access.

SCPTS **SCRIPTS** **0**
This bit is used to define if a SCRIPTS fetch is considered to be a local memory access.

Register: 0x47

General Purpose Pin Control Zero (GPCNTL0)
Read/Write

7	6	5	4		2	1	0
ME	FE	LEDC	GPIO			GPIO	
0	0	0	0	1	1	1	1

This register is used to determine if the pins controlled by the General Purpose (GPREG0) register are inputs or outputs. Bits [4:0] in GPCNTL0 correspond to bits [4:0] in the GPREG0 register. When the bits are enabled as inputs, internal pull-downs are enabled for GPIO[4:2] and internal pull-ups are enabled for GPIO[1:0].

The data written to each bit of the GPREG0 register is output to the appropriate GPIO pin if it is set to the output mode in the GPCNTL0 register.

ME **Master Enable** **7**
The internal bus master signal is presented on GPIO1 if this bit is set, regardless of the state of bit 1 (GPIO1).

FE **Fetch Enable** **6**
The internal opcode fetch signal is presented on GPIO0 if this bit is set, regardless of the state of bit 0 (GPIO0).

LEDC **LED_CNTL** **5**
The internal connected signal (bit 3 of the Interrupt Status Zero (ISTAT0) register) will be presented on GPIO0 if this bit is set and bit 6 of GPCNTL0 is cleared and the chip is not in progress of performing an EEPROM autodownload regardless of the state of bit 0 (GPIO0). This provides a hardware solution to driving a SCSI activity LED in many implementations of LSI Logic SCSI chips.

GPIO **GPIO Enable** **[4:2]**
General purpose control, corresponding to bits [4:2] in the GPREG0 register and pins GPIO[4:2]. GPIO4 powers up as a general purpose output, and GPIO[3:2] power-up as general purpose inputs.

GPIO **GPIO Enable** **[1:0]**

These bits power-up set, causing the GPIO1 and GPIO0 pins to become inputs. Clearing these bits causes 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, 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] GEN [3:0]	Minimum Time-out (80 MHz Clock) With Scale Factor Bit Cleared[1]	Minimum Time-out (80 MHz Clock) With Scale Factor Bit Set
0000	Disabled	Disabled
0001	100 µs	1.6 ms
0010	200 µs	3.2 ms
0011	400 µs	6.4 ms
0100	800 µs	12.8 ms
0101	1.6 ms	25.6 ms
0110	3.2 ms	51.2 ms
0111	6.4 ms	102.4 ms
1000	12.8 ms	204.8 ms
1001	25.6 ms	409.6 ms
1010	51.2 ms	819.2 ms
1011	102.4 ms	1.6 s
1100	204.8 ms	3.2 s
1101	409.6 ms	6.4 s
1110	819.2 ms	12.8 s
1111	1.6 + s	25.6 s

1. These values are correct if the CCF bits in the SCSI Control Three (SCNTL3) register are set according to the valid combinations in the bit description. A quadrupled 40 MHz clock is required for Ultra2 SCSI operation.

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/, 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.

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

7							0
ID							
x	x	x	x	x	x	x	x

RESPID0 and Response ID One (RESPID1) contain the selection or reselection IDs. In other words, these two 8-bit registers contain the ID that the chip responds to on the SCSI bus. Each bit represents one possible ID with the most significant bit of RESPID1 representing ID 15 and the least significant bit of RESPID0 representing 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: 0x4B

Response ID One (RESPID1)
Read/Write

15							8
ID							
x	x	x	x	x	x	x	x

Response ID Zero (RESPID0) and RESPID1 contain the selection or reselection IDs. In other words, these two 8-bit registers contain the ID that the chip responds to on the SCSI bus. Each bit represents one possible ID with the most significant bit of RESPID1 representing ID 15 and the least significant bit of RESPID0 representing 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				SLT	ART	SOZ	SOM
x	x	x	x	0	x	1	1

SSAID **SCSI Selected As ID** **[7:4]**

These bits contain the encoded value of the SCSI ID that the LSI53C895A is selected during a SCSI selection phase. These bits work in conjunction with the Response ID Zero (RESPID0) and Response ID One (RESPID1) registers, which contain the allowable IDs that the LSI53C895A can respond to. During a SCSI selection phase, when a valid ID is put on the bus, and the LSI53C895A responds to that ID, the ID that the chip was selected as will be written into the SSAID[3:0] bits.

SLT **Selection Response Logic Test** **3**

This bit is set when the LSI53C895A 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 LSI53C895A exhibits the highest priority ID asserted on the SCSI bus during arbitration. It is primarily used for chip level testing, but it may be used during low level mode operation to determine if the LSI53C895A 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, the LSI53C895A functioning as an initiator, is waiting for the target to request data transfers. If the LSI53C895A is 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 [3:0] 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. When this bit is set, the LSI53C895A, as a target, is waiting for the initiator to acknowledge the data transfers. If the LSI53C895A is an initiator, then 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
SCLK	ISO	R		QEN	QSEL	ISEL[1:0]	
0	0	x	x	0	0	0	0

SCLK **SCSI Clock** **7**

When set, this bit disables the external SCLK (SCSI Clock) pin, and the chip uses the PCI clock as the internal SCSI clock. When set, it will also select the PCI clock as the internal SCSI clock if the internal clock quadrupler is enabled and selected. If a transfer rate of 10 Mbytes/s, 20 Mbytes/s, or 40 Mbytes/s (20 Mbytes/s, 40 Mbytes/s, or 80 Mbytes/s on a wide SCSI bus) is desired on the SCSI bus, this bit must be cleared and a 40 MHz external SCLK must be provided.

ISO **SCSI Isolation Mode** **6**

This bit allows the LSI53C895A to put the SCSI bidirectional and input pins into a low power mode when the SCSI bus is not in use. When this bit is set, the SCSI bus inputs are logically isolated from the SCSI bus.

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

QEN **SCLK Quadrupler Enable** **3**
This bit, when set, powers up the internal clock quadrupler circuit, which quadruples the SCLK 40 MHz clock to an internal 160 MHz SCSI clock required for Ultra SCSI and Ultra2 SCSI operation. When cleared, this bit powers down the internal quadrupler circuit.

QSEL **SCLK Quadrupler Select** **2**
This bit, when set, selects the output of the internal clock quadrupler for use as the internal SCSI clock. When cleared, this bit selects the clock presented on SCLK for use as the internal SCSI clock.

ISEL[1:0] **Interrupt Select** **[1:0]**
The LSI53C895A supports different interrupt routing modes. These modes are described in the following table. For additional information on the LSI53C895A interrupt routing modes see Section 2.2.17, “Interrupt Routing,” in Chapter 2.

Mode	ISEL[1:0]	Interrupt Routing
0	00	Interrupts are signaled on IRQ/ and ALT_IRQ/.
1	01	Interrupts are only signaled on IRQ/.
2	10	Interrupts are only signaled on ALT_IRQ/.
3	11	Reserved.

Register: 0x4E
SCSI Test Two (STEST2)
Read/Write

7	6	5	4	3	2	1	0
SCE	ROF	DIF	SLB	SZM	AWS	EXT	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 LSI53C895A 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 diagnostic purposes only.

ROF **Reset SCSI Offset** **6**
Setting this bit clears any outstanding synchronous SREQ/SACK offset. Set this bit if a SCSI gross error condition occurs and to clear the offset when a synchronous transfer does not complete successfully. The bit automatically clears itself after resetting the synchronous offset.

DIF **HVD or SE/LVD** **5**
Setting this bit allows the LSI53C895A to interface to external HVD transceivers. Clearing this bit enables SE or LVD operation. Set this bit in the initialization routine if the differential pair interface is used.

SLB **SCSI Loopback Mode** **4**
Setting this bit allows the LSI53C895A to perform SCSI loopback diagnostics. That is, it enables the SCSI core to simultaneously perform as both the initiator and the target.

SZM **SCSI High Impedance Mode** **3**
Setting this bit places all the open drain 48 mA SCSI drivers into a high impedance state. This is to allow internal loopback mode operation without affecting the SCSI bus.

AWS **Always Wide SCSI** **2**
When this bit is set, all SCSI information transfers are done in 16-bit wide mode. This includes data, message, command, status and reserved phases. Normally, deassert this bit since 16-bit wide message, command, and status phases are not supported by the SCSI specifications.

EXT **Extend SREQ/SACK/ Filtering** **1**
LSI Logic TolerANT SCSI receiver technology includes a special digital filter on the SREQ/ and SACK/ pins which causes the disregarding of glitches on deasserting edges. Setting this bit increases the filtering period from 30 ns to 60 ns on the deasserting edge of the SREQ/ and SACK/ signals.

Note: Never set this bit during fast SCSI (greater than 5 Mbyte transfers per second) operations, because a valid assertion could be treated as a glitch.

LOW **SCSI Low level Mode** **0**

Setting this bit places the LSI53C895A 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 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	STR	HSC	DSI	S16	TTM	CSF	STW
0	0	0	0	x	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 LSI53C895A is driving these signals. Active deassertion of these signals occurs only when the LSI53C895A is in an information transfer phase. When operating in a differential environment or at fast SCSI timings, TolerANT Active negation 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, “General Description.”

Note: Set this bit if the Enable Ultra SCSI bit in SCSI Control Three (SCNTL3) is set.

STR **SCSI FIFO Test Read** **6**

Setting this bit places the SCSI core into a test mode in which the SCSI FIFO is easily read. Reading the least significant byte of the SCSI Output Data Latch (SODL) register causes the FIFO to unload. The functions are summarized in the table below.

Register Name	Register Operation	FIFO Bits	FIFO Function
SODL	Read	[15:0]	Unload
SODL0	Read	[7:0]	Unload
SODL1	Read	[15:8]	None

HSC **Halt SCSI Clock** **5**

Asserting 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.

DSI **Disable Single Initiator Response** **4**

If this bit is set, the LSI53C895A ignores all bus-initiated selection attempts that employ the single initiator option from SCSI-1. In order to select the LSI53C895A while this bit is set, the LSI53C895A'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.

S16 **16-Bit System** **3**

If this bit is set, all devices in the SCSI system implementation are assumed to be 16-bit. This causes the LSI53C895A to always check the parity bit for SCSI IDs [15:8] during bus-initiated selection or reselection, assuming parity checking has been enabled. If an 8-bit SCSI device attempts to select the LSI53C895A while this bit is set, the LSI53C895A will ignore the selection attempt. This is because the parity bit for IDs [15:8] will not be driven. See the description of the Enable Parity Checking bit in the SCSI Control Zero (SCNTL0) register for more information.

TTM **Timer Test Mode** **2**

Asserting 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 and if the respective bits in the SCSI Interrupt Enable One (SIEN1) register are asserted, the LSI53C895A generates interrupts at time-out. This bit is intended for internal manufacturing diagnosis and should not be used.

CSF **Clear SCSI FIFO** **1**

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 SCSI Input Data Latch (SIDL), SCSI Output Data Latch (SODL), and (SODR, a hidden buffer register which is not accessible) full bits in the SCSI Status Zero (SSTAT0) and SCSI Status Two (SSTAT2) are cleared.

STW **SCSI FIFO Test Write** **0**

Setting this bit places the SCSI core into a test mode in which the FIFO is easily read or written. While this bit is set, writes to the least significant byte of the SCSI Output Data Latch (SODL) register cause the entire word contained in the SODL to be loaded into the FIFO. These functions are summarized in the table below.

Register Name	Register Operation	FIFO Bits	FIFO Function
SODL	Write	[15:0]	Load
SODL0	Write	[7:0]	Load
SODL1	Write	[15:8]	None

Registers: 0x50–0x51
SCSI Input Data Latch (SIDL)
Read Only

15															0
SIDL															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

SIDL **SCSI Input Data Latch** **[15:0]**

This register is used primarily for diagnostic testing, programmed I/O operation, or error recovery. Data received from the SCSI bus can be read from this register. Data can be written to the SCSI Output Data Latch (SODL) register and then read back into the LSI53C895A by reading this register to allow loopback testing. When receiving 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; SIDL contains latched data and the 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 not valid. The power-up values are indeterminate.

Register: 0x52
SCSI Test Four (STEST4)
Read Only

7	6	5	4				0
SMODE[1:0]		LOCK	R				
1	1	0	x	x	x	x	x

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.

Bit [7:6]	Operating Mode
00	Not Possible
01	HVD or powered down (for HVD mode, the DIF bit must also be set)
10	SE
11	LVD SCSI

LOCK **Frequency Lock** **5**

This bit is used when enabling the SCSI clock quadrupler, which allows the LSI53C895A to transfer data at Ultra2 SCSI rates. Poll this bit for a 1 to determine that the clock quadrupler has locked to 160 MHz. For more information on enabling the clock quadrupler, refer to the descriptions of SCSI Test One (STEST1), bits 2 and 3.

R **Reserved** **[4:0]**

Register: 0x53

Reserved

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 diagnostic testing or programmed I/O operation. Data written to this register is asserted onto the SCSI data bus by setting the Assert Data Bus bit in the 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 any 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 0 (CCNTL0)

Read/Write

7	6	5	4	3	2	1	0
ENPMJ	PMJCTL	ENNDJ	DISFC	R		DILS	R
0	0	0	0	x	x	0	x

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 will be calculated such that they can be easily stored to the appropriate memory location with SCRIPTS Store instruction.

In the case of a SCSI send, any data in the part will be 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 LSI53C895A will use Phase Mismatch Jump Address 1 (PMJAD1) when the WSR bit is cleared and Phase Mismatch Jump Address 2 (PMJAD2) when the WSR bit is set. When this bit is set the LSI53C895A will use jump address one (PMJAD1) on data out (data out, command, message out) transfers and jump address two (PMJAD2) on data in (data in, status, message in) transfers. 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.

ENNDJ **Enable Jump On Nondata Phase Mismatches** **5**

This bit controls whether or not a jump is taken during a nondata phase mismatch (i.e. message in, message out, status, or command). When this bit is clear, 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 as well as data in the SCSI Output Data Latch (SODL) and SODR, a hidden buffer register which is not accessible, will not be cleared after calculations on them are complete. When cleared, the DMA FIFO and SODL and SODR will automatically be cleared. This bit also disables the enhanced flushing mechanism.

R **Reserved** **[3:2]**

DILS **Disable Internal Load and Store** **1**

This bit controls whether or not Load and Store data transfers in which the source/destination is located in SCRIPTS RAM generate external PCI cycles.

If cleared, Load and Store data transfers of this type will NOT generate PCI cycles, but will stay internal to the chip.

If set, Load and Store data transfers of this type will generate PCI cycles.

R **Reserved** **0**

Register: 0x57
Chip Control 1 (CCNTL1)
Read/Write

7	6		4	3	2	1	0
ZMODE	R			DDAC	64TIMOD	EN64TIBMV	EN64DBMV
0	x	x	x	0	0	0	0

ZMODE **High Impedance Mode** **7**

Setting this bit causes the LSI53C895A to place all output and bidirectional pins except MAC/_TESTOUT, into a high impedance state. Also, setting this bit causes all I/O pins to become inputs, and all pull-ups and pull-downs to be disabled. When this bit is set, the MAC/_TESTOUT pin becomes the output pin for the connectivity test of the LSI53C895A signals in the “AND-tree” test mode. In order to read data out of the LSI53C895A, this bit must be cleared. This bit is intended for board-level testing only. Do not set this bit during normal system operation.

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

DDAC **Disable Dual Address Cycle** **3**

When this bit is set, all 64-bit addressing as a master will be disabled. No dual address cycles will be generated by the LSI53C895A.

When this bit is cleared, the LSI53C895A will generate dual address cycles based on the master operation being performed and the value of its associated selector register.

64TIMOD **64-Bit Table Indirect Indexing Mode** **2**

When this bit is cleared, bits [24:28] 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 [24:31] of the first table entry Dword will be copied directly into 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:

[31:29]	[28:24]	[23:0]
Reserved	Sel Index	Byte Count
Source/Destination Address [31:0]		

Index Mode 1 (64TIMOD set) table entry format:

[31:24]	[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 (bit [24:31]) of the first Dword of the table entry. When this bit is cleared table indirect BMOVs will 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 will 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															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

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 diagnostic testing or in low level mode. The power-up value of this register is indeterminate.

If the chip is in the wide mode (SCSI Control Three (SCNTL3), bit 3 and SCSI Test Two (STEST2), bit 2 are set) and 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 upper byte lane parity is invalid.

Register: 0x5A

General Purpose Pin Control One (GPCNTL1)
Read/Write

7			4	3			0
R				GPIOEN[8:5]			
0	x	x	x	1	1	1	1

This register is used to determine if the signals controlled by the General Purpose One register are inputs or outputs. Bits [3:0] in GPCNTL1 correspond to bits [3:0] in General Purpose Pin Control One (GPCNTL1) register. When the bits are enabled as inputs, an internal pull-down is also enabled. The data written to each bit of the GPREG1 register is output to the appropriate GPIO pin if it is set to the output mode in this register.

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

GPIOEN[8:5] **GPIO Enable** **[3:0]**
General purpose control, corresponding to bits [3:0] in the General Purpose One (GPREG1) register. GPIO[8:5] power-up as general purpose inputs.

Register: 0x5B

General Purpose One (GPREG1)
Read/Write

7			4	3			0
R				GPIO[8:5]			
x	x	x	x	x	x	x	x

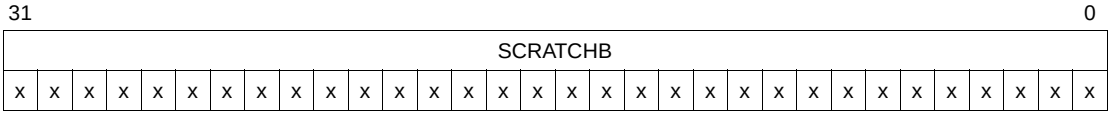
R **Reserved** **[7:4]**

GPIO[8:5] **General Purpose I/O** **[3:0]**
These bits can be programmed through the General Purpose Pin Control One (GPCNTL1) register to become inputs or outputs. As outputs, these pins can be used to enable or disable external terminators. These signals can also be programmed as live inputs and sensed through a SCRIPTS register to Register Move instruction.
GPIO[8:5] default as inputs. When configured as inputs, an internal pull-down is enabled.

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 Into Enable bit in the Chip Test Two (CTEST2) register. If this bit is set, the SCRATCH B register returns bits [31:13] of the SCRIPTS RAM PCI Base Address Register Two (SCRIPTS RAM) in bits [31:13] of the SCRATCH B register when read. When read, bits [12:0] of SCRATCH B will always return zeros in this mode. Writes to the SCRATCH B register are unaffected. Resetting the PCI Configuration Into Enable bit causes the SCRATCH B register to return to normal operation.

Registers: 0x60–x9F

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 and Stores directed at a SCRATCH register will alter its contents. The power-up values are indeterminate.

4.3 64-Bit SCRIPTS Selectors

The following registers are used to hold the upper 32-bit addresses for various SCRIPTS operations. When a particular type of SCRIPTS operation is performed, one of the six selector registers below will be used to generate a 64-bit address.

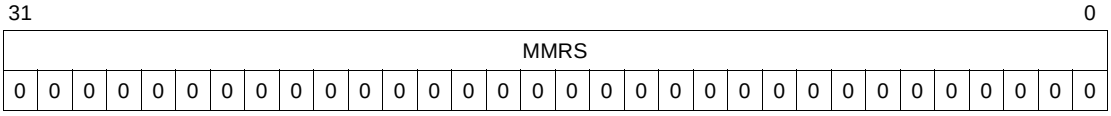
If the selector for a particular device operation is zero, then a standard 32-bit address cycle will be generated. If the selector value is nonzero, then a DAC will be issued and the 64-bit address will be presented in two address phases.

All selectors default to 0 (zero) with the exception of the 16 SCRATCH registers, these power-up in an indeterminate state and should be initialized before they are used.

All selectors can be read/written using the Load and Store SCRIPTS instruction, Memory-to-Memory Move, Read/Write SCRIPTS instruction, or CPU with SCRIPTS not running.

Note: Crossing of selector boundaries in one memory operation is not supported.

Registers: 0xA0–0xA3
Memory Move Read Selector (MMRS)
Read/Write



MMRS **Memory Move Read Selector** **[31:0]**
Supplies the upper Dword of a 64-bit address during data read operations for Memory-to-Memory Moves and absolute address LOAD operations.

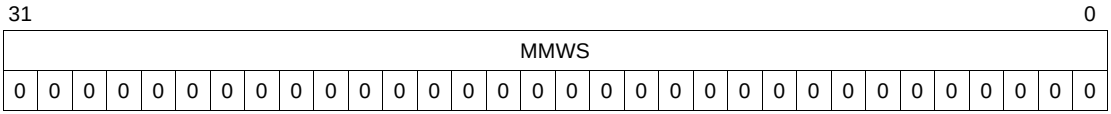
A special mode of this register can be enabled by setting the PCI Configuration Enable bit in the Chip Test Two (CTEST2) register. Because the LSI53C895A supports only a 32-bit memory mapped PCI base address, the MMRS register is always read as 0x00000000 when in the special mode.

Writes to the MMRS register are unaffected. Clearing the PCI Configuration Into 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]**

Supplies the upper Dword of a 64-bit address during data write operations during Memory-to-Memory Moves and absolute address STORE operations.

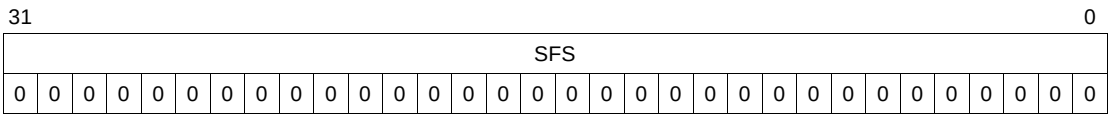
A special mode of this register can be enabled by setting the PCI Configuration Into Enable bit in the Chip Test Two (CTEST2) register. Because the LSI53C895A supports only a 32-bit SCRIPTS RAM PCI base address, the MMWS register is always read as 0x00000000 when in the special mode.

Writes to the MMWS register are unaffected. Clearing the PCI Configuration Enable bit causes the MMWS register to return to normal operation.

Registers: 0xA8–0xAB

SCRIPTS Fetch Selector (SFS)

Read/Write



SFS **SCRIPTS Fetch Selector** **[31:0]**

Supplies the upper Dword of a 64-bit address during SCRIPTS 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 Into Enable bit in the Chip Test Two (CTEST2) register. If this bit is set, bits [23:16] of the SFS

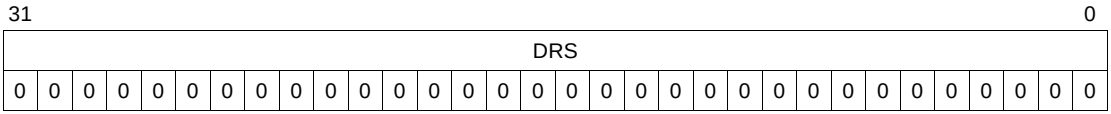
register return the PCI Revision ID (Rev ID) register value and bits [15:0] return the PCI Device ID register value when read.

Writes to the SFS register are unaffected. Clearing the PCI Configuration Into Enable bit causes the SFS register to return to normal operation.

Registers: 0xAC–0xAF

DSA Relative Selector (DRS)

Read/Write



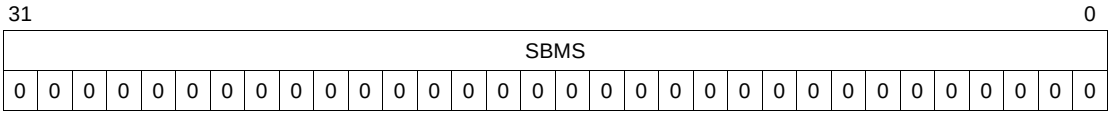
DRS **DSA Relative Selector** **[31:0]**

Supplies the upper Dword of a 64-bit address during table indirect fetches and Load and Store Data Structure Address (DSA) relative operations.

Registers: 0xB0–0xB3

Static Block Move Selector (SBMS)

Read/Write

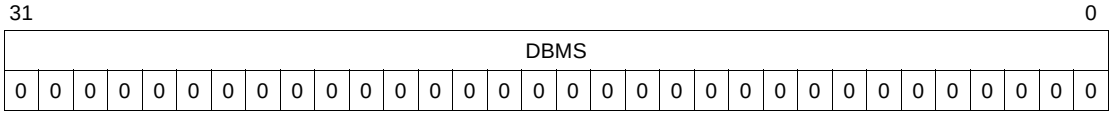


SBMS **Static Block Move Selector** **[31:0]**

Supplies the upper Dword of a 64-bit address during block move operations, reads or writes. This register is static and will not be changed when a 64-bit direct BMOV is used.

Registers: 0xB4–0xB7

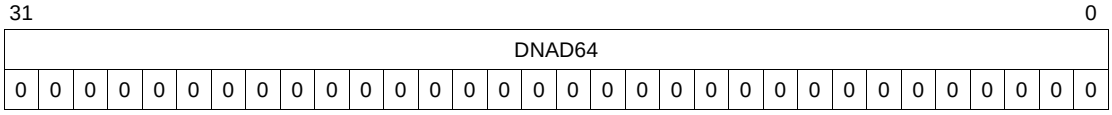
Dynamic Block Move Selector (DBMS)
Read/Write



DBMS **Dynamic Block Move Selector** **[31:0]**
Supplies the upper Dword of a 64-bit address during block move operations, reads or writes. This register is used only during 64-bit direct BMOV instructions and will be reloaded with the upper 32-bit data address upon execution of a 64-bit direct BMOVs.

Registers: 0xB8–0xBB

DMA Next Address 64 (DNAD64)
Read/Write



DNAD64 **DMA Next Address 64** **[31:0]**
This register holds the current selector being used in a given host transaction. The appropriate selector is copied to this register prior to beginning a host transaction.

Registers: 0xBC–0xBF

Reserved

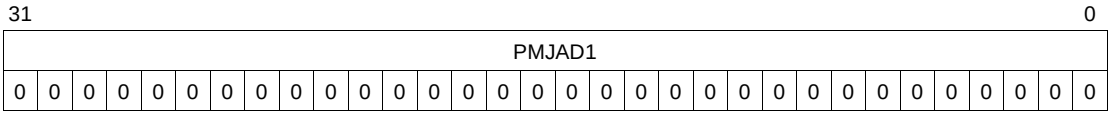
4.4 Phase Mismatch Jump Registers

Eight 32-bit 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 eight register descriptions follow.

All 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.

Registers: 0xC0–0xC3

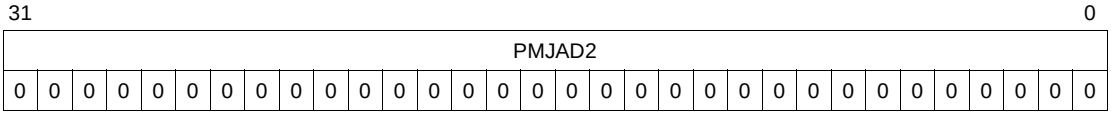
Phase Mismatch Jump Address 1 (PMJAD1)
Read/Write



PMJAD1 **Phase Mismatch Jump Address 1** **[31:0]**
This register contains the 32-bit address that will be jumped to upon a phase mismatch. Depending upon the state of the PMJCTL bit in register Chip Control 0 (CCNTL0) this address will either be used during an outbound (data out, command, message out) phase mismatch (PMJCTL = 0) or when the WSR bit is cleared (PMJCTL = 1). It should be loaded with an address of a SCRIPTS routine that will handle the updating of memory data structures of the BMOV that was executing when the phase mismatch occurred.

Registers: 0xC4–0xC7

Phase Mismatch Jump Address 2 (PMJAD2)
Read/Write

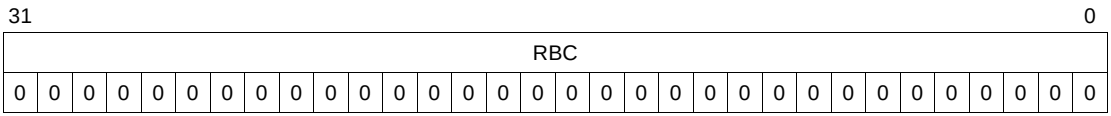


PMJAD2 **Phase Mismatch Jump Address 2** **[31:0]**

This register contains the 32-bit address that will be jumped to upon a phase mismatch. Depending upon the state of the PMJCTL bit in register Chip Control 0 (CCNTL0) this address will either be used during an inbound (data in, status, message in) phase mismatch (PMJCTL = 0) or when the WSR bit is set (PMJCTL = 1). It should be loaded with an address of a SCRIPTS routine that will handle the updating of 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 (RBC)** **[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 will also contain the opcode of the BMOV that was executing. In the case of a table indirect BMOV instruction, the upper byte will contain the upper byte of the table indirect entry that was fetched.

In the case of a SCSI data receive, this byte count will reflect all data received from the SCSI bus, including any byte in SCSI Wide Residue (SWIDE). There will be no data remaining in the part that must be flushed to

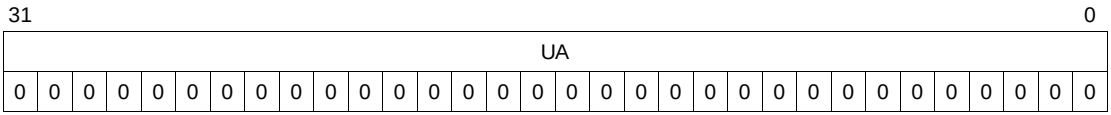
memory with the exception of a possible byte in the SWIDE register. That byte must be flushed to memory manually in SCRIPTS.

In the case of a SCSI data send, this byte count will reflect all data sent out onto the SCSI bus. Any data left in the part from the phase mismatch will be ignored and automatically cleared from the FIFOs.

Registers: 0xCC–0xCF

Updated Address (UA)

Read/Write



UA **Updated Address** **[31:0]**

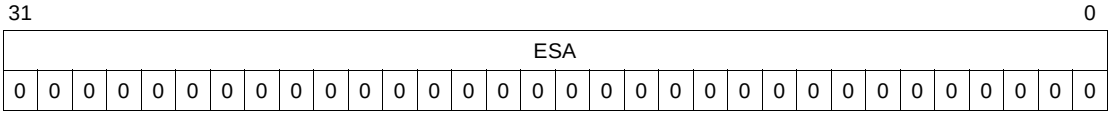
This register will contain the updated data address for the BMOV that was executing when the phase mismatch occurred.

In the case of a SCSI data receive, if there is a byte in the SCSI Wide Residue (SWIDE) register then this address will point 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 will be 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 will be ignored and will be automatically cleared from the FIFOs.

Registers: 0xD0–0xD3
Entry Storage Address (ESA)
Read/Write



ESA **Entry Storage Address** **[31:0]**

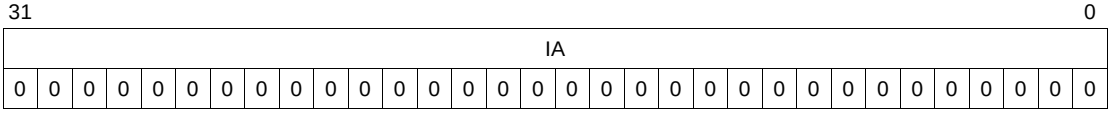
This register's value depends on the type of BMOV being executed. The three types of BMOVs are listed below.

Direct BMOV: In the case of a direct BMOV, this register will contain the address the BMOV was fetched from when the phase mismatch occurred.

Indirect BMOV: In the case of an indirect BMOV, this register will contain 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 will contain the address of the table indirect entry being used when the phase mismatch occurred.

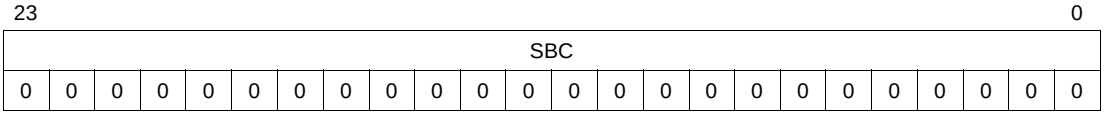
Registers: 0xD4–0xD7
Instruction Address (IA)
Read/Write



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 will point to the address of the BMOV instruction.

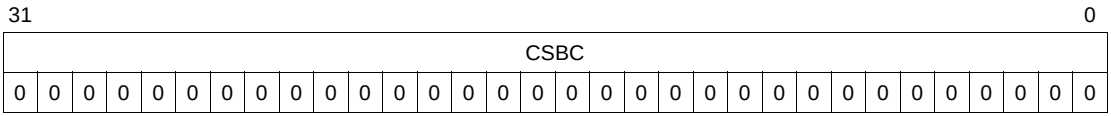
Registers: 0xD8–0xDA
SCSI Byte Count (SBC)
Read only



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) register and should not need to be used in normal operations. There are two conditions in which this byte count will not match the number of bytes transferred exactly. If a BMOV is executed to transfer an odd number of bytes across a wide bus then the byte count at the end of the BMOV will be greater than the number of bytes sent by one. This will also happen in an odd byte count wide receive case. Also, in the case of a wide send in which there is a chain byte from a previous transfer, the count will not reflect the chain byte sent across the bus during that BMOV. The reason for this is due to the fact that to determine the correct address to start fetching data from after a phase mismatch this byte cannot be counted for this BMOV as it was actually part of the byte count for the previous BMOV.

Register: 0xDB
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 actual number of bytes that have been transferred across

the SCSI bus during data phases, i.e. it will not count bytes sent in command, status, message in or message out phases. It will count bytes as long as the phase mismatch enable (ENPMJ) bit in the Chip Control 0 (CCNTL0) register is set. Unlike the SCSI Byte Count (SBC) this count will not be cleared on each BMOV instruction but will continue to count across multiple BMOV instructions. This register can be loaded with any arbitrary start value.

Registers: 0xE0–0xFF

Reserved

Chapter 5
SCSI SCRIPTS Instruction Set

The LSI53C895A contains a SCSI SCRIPTS processor that permits both DMA and SCSI commands to be fetched from host memory or internal SCRIPTS RAM. Algorithms written in SCSI SCRIPTS control the actions of the SCSI and DMA cores. The SCRIPTS processor executes complex SCSI bus sequences independently of the host CPU. This chapter describes the SCSI SCRIPTS Instruction Set used to write these algorithms. The following sections describe the benefits and use of SCSI SCRIPTS Instructions.

After power-up and initialization, the LSI53C895A can be operated in the low level register interface mode or in the high level SCSI SCRIPTS mode.

5.1 Low Level Register Interface Mode

With the low level register interface mode, 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 allows 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. Another feature allowed at the low level is loopback testing. In loopback mode, the SCSI core can be directed to talk to the DMA core to test internal data paths all the way out to the chip's pins.

5.2 High Level SCSI SCRIPTS Mode

To operate in the SCSI SCRIPTS mode, the LSI53C895A requires only a SCRIPTS start address. The start address must be at a Dword (four byte) boundary. This aligns subsequent 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 LSI53C895A halts all operations until the interrupt is serviced. Then, the start address of the next SCRIPTS instruction may be written to the DMA SCRIPTS Pointer (DSP) register to restart the automatic fetching and execution of instructions.

The SCSI SCRIPTS mode of execution allows the LSI53C895A to make decisions based on the status of the SCSI bus, which offloads 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 low level mode for error recovery should never be required.

The following types of SCRIPTS instructions are implemented in the LSI53C895A, as shown in Table 5.1:

Table 5.1 SCRIPTS Instructions

Instruction	Description
Block Move	Block Move instruction moves data between the SCSI bus and memory.
I/O or Read/Write	I/O or Read/Write instructions cause the LSI53C895A to trigger common SCSI hardware sequences, or to move registers.
Transfer Control	Transfer Control instruction allows SCRIPTS instructions to make decisions based on real time SCSI bus conditions.
Memory Move	Memory Move instruction causes the LSI53C895A to execute block moves between different parts of main memory.
Load and Store	Load and Store instructions provide 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 is followed by one or two more 32-bit fetch cycles.

5.2.1 Sample Operation

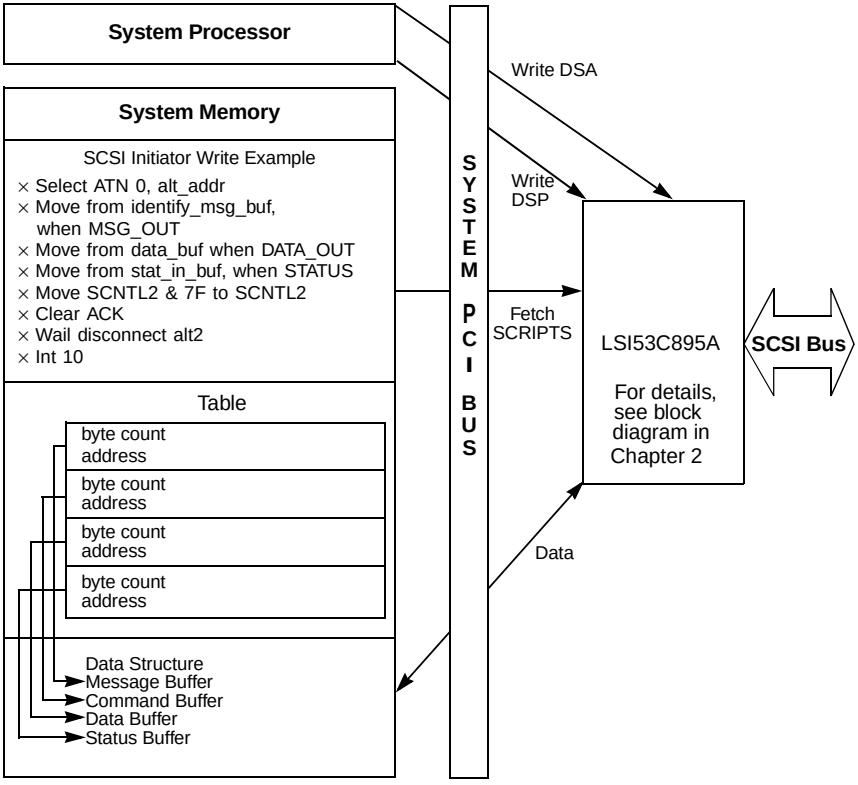
This sample operation describes execution of a SCRIPTS instruction for a Block Move instruction.

- 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 LSI53C895A to fetch its first instruction at the address just loaded. This is from main memory or the internal RAM, depending on the address.

- The LSI53C895A typically fetches two Dwords (64 bits) and decodes the high order byte of the first longword as a SCRIPTS instruction. If the instruction is a Block Move, the lower three bytes of the first longword are stored and interpreted as the number of bytes to be moved. The second longword 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 LSI53C895A 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 LSI53C895A requests use of the PCI bus again to transfer the data.
- When the LSI53C895A is granted the PCI bus, it executes (as a bus master) a burst transfer (programmable size) of data, decrement the internally stored remaining byte count, increment the address pointer, and then release the PCI bus. The LSI53C895A 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 LSI53C895A 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 LSI53C895A 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 LSI53C895A performs scatter/gather operations on data without requiring help from the host program, generating a host interrupt, or requiring an external DMA controller to be programmed. An overview of this process is presented in Figure 5.1.

Figure 5.1 SCRIPTS Overview



5.3 Block Move Instruction

Performing a Block Move instruction, bit 5, Source I/O - Memory Enable (SIOM) and bit 4, Destination I/O - Memory Enable (DIOM) in the DMA Mode (DMODE) register determines whether the source/destination address resides in memory or I/O space. When data is being moved onto the SCSI bus, SIOM controls whether that data comes from I/O or memory space. When data is being moved off of the SCSI bus, DIOM controls whether that data goes to I/O or memory space.

5.3.1 First Dword

31 30 29 28 27 26 24	23 16	15 8	7 0
DMA Command (DCMD) Register	DMA Byte Counter (DBC) Register		
IT[1:0] IA TIA OPC SCSIP[2:0]	Transfer Counter [23:0]		
0 0 x x x x x x	x x x x x x x x	x x x x x x x x	x x x x x x x x

IT[1:0] **Instruction Type - Block Move** **[31:30]**
The IT bit configuration (00) defines a Block Move Instruction Type.

IA **Indirect Addressing** **29**
This bit determines if addressing is direct or indirect. If IA bit is (0), use destination field as an address (direct addressing). If IA bit is (1), use destination field as a pointer to an address (indirect addressing).

When this bit is zero, 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 this bit is one, 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 longword fetch (4-byte transfer across the host computer bus).

Direct Addressing

The byte count and absolute address are:

Command	Byte Count
Address of Data	

Indirect Addressing

Use the fetched byte count, but fetch the data address from the address in the instruction.

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 a table of data buffer addresses to be specified. Using the LSI Logic SCSI SCRIPTS assembler, the table offset is placed in the script at compile time. Then at the actual data transfer time, the offsets are added to the base address of the data address table by the external processor. The logical I/O driver builds a structure of addresses for an I/O rather than treating each address individually. This feature makes it possible to locate SCSI SCRIPTS in a PROM.

Note: Do not use indirect and table indirect addressing simultaneously; 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: Do not use indirect and table indirect addressing simultaneously; use only one addressing method at a time.

Prior to the start of an I/O, the Data Structure Address (DSA) register should be loaded with the base address of the I/O data structure. The address may be any address on a longword boundary.

After a Table Indirect opcode is fetched, the 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 LSI53C895A. 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 longword 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, as shown below, and
- Indirect data fetches are not available during execution of a Memory-to-Memory DMA operation.

00	Byte Count
Physical Data Address	

OPC **OpCode** **27**

This 1-bit OpCode field defines the type of Block Move (MOVE) Instruction to be preformed in Target and Initiator mode.

Target Mode

In Target mode, the OpCode bit defines the following operations:

OPC	Instruction Defined
0	MOVE/MOVE64
1	CHMOV/CHMOV64

These instructions perform the following steps:

1. The LSI53C895A verifies that it is connected to the SCSI bus as a Target before executing this instruction.
2. The LSI53C895A asserts the SCSI phase signals (SMSG/, SC_D/, and SI_O/) as defined by the Phase Field bits in the instruction.
3. If the instruction is for the command phase, the LSI53C895A 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, and if the Vendor Unique Enhancement 1 (VUE1) bit (SCNTL2 bit 1) is clear, then the LSI53C895A 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 1 (VUE1) bit (SCNTL2 bit 1) is set, the LSI53C895A receives the number of bytes in the byte count regardless of the group code.
   - If the Vendor Unique Enhancement 1 bit is clear and group code is vendor unique, the LSI53C895A receives the number of bytes in the count.
   - If any other Group Code is received, the DMA Byte Counter (DBC) register is not modified and the LSI53C895A 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.
4. The LSI53C895A transfers the number of bytes specified in the 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 LSI53C895A 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 be completed.

5. If the SATN/ signal is asserted by the Initiator or a parity error occurred during the transfer, the transfer can optionally be halted and an interrupt generated. The Disable Halt on Parity Error or ATN bit in the SCSI Control One (SCNTL1) register controls whether the LSI53C895A halts on these conditions immediately, or waits until completion of the current Move.

Initiator Mode

In Target mode, the OpCode bit defines the following operations:

OPC	Instruction Defined
0	CHMOV
1	MOVE

These instructions perform the following steps:

1. The LSI53C895A verifies that it is connected to the SCSI bus as an Initiator before executing this instruction.
2. The LSI53C895A waits for an unserviced phase to occur. An unserviced phase is any phase (with SREQ/ asserted) for which the LSI53C895A has not yet transferred data by responding with a SACK/.
3. The LSI53C895A 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.
4. If the SCSI phase bits match the value stored in the SCSI SCSI Status One (SSTAT1) register, the LSI53C895A 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 LSI53C895A 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.

5. If the SCSI phase bits do not match the value stored in the SCSI Status One (SSTAT1) register, the LSI53C895A generates a phase mismatch interrupt and the instruction is not executed.
6. During a Message-Out phase, after the LSI53C895A has performed a select with Attention (or SATN/ is manually asserted with a Set ATN instruction), the LSI53C895A deasserts SATN/ during the final SREQ/SACK/ handshake.
7. When the LSI53C895A 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 3-bit field defines the SCSI information transfer phase. When the LSI53C895A operates in Initiator mode, these bits are compared with the latched SCSI phase bits in the SCSI Status One (SSTAT1) register. When the LSI53C895A operates in Target mode, it asserts the phase defined in this field. Table 5.2 describes the possible combinations and the corresponding SCSI phase.

Table 5.2 SCSI Information Transfer Phase

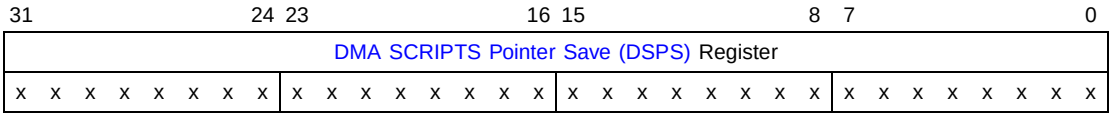
MSG	C_D	I_O	SCSI Phase
0	0	0	Data-Out
0	0	1	Data-In
0	1	0	Command
0	1	1	Status
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Message-Out
1	1	1	Message-In

TC[23:0] **Transfer Counter** **[23:0]**

This 24-bit field specifies the number of data bytes to be moved between the LSI53C895A and system memory. The field is stored in the DMA Byte Counter (DBC) register. When the LSI53C895A 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 LSI53C895A 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.

5.3.2 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 LSI53C895A 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.

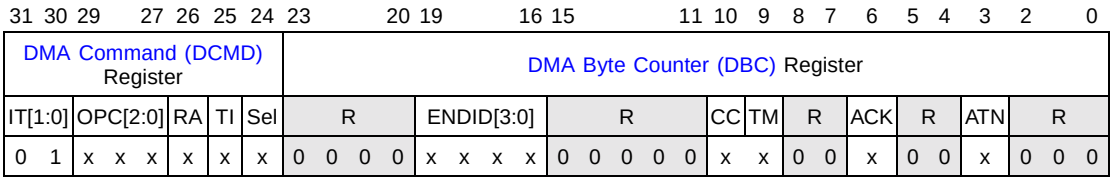
5.4 I/O Instruction

I/O Instructions perform the following SCSI operations in Target and Initiator mode. These I/O operations are chosen with the opcode bits in the DMA Command (DCMD) register.

OPC2	OPC1	OPC0	Target Mode	Initiator Mode
0	0	0	Reselect	Select
0	0	1	Disconnect	Wait Disconnect
0	1	0	Wait Select	Wait Reselect
0	1	1	Set	Set
1	0	0	Clear	Clear

This section describes these I/O operations.

5.4.1 First Dword



IT[1:0] **Instruction Type - I/O Instruction** **[31:30]**
The IT bit configuration (01) defines an I/O Instruction Type.

OPC[2:0] **OpCode** **[29:27]**
The OpCode bit configurations define the I/O operation performed but the OpCode bit meanings change in Target mode compared to Initiator mode. OpCode bit configurations (101, 110, and 111) are considered Read/Write instructions, and are described in Section 5.5, “Read/Write Instructions.” This section describes Target mode operations.

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 LSI53C895A 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 LSI53C895A wins arbitration, it attempts to reselect the SCSI device whose ID is defined in the destination ID field of the instruction. Once the LSI53C895A 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 LSI53C895A 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 LSI53C895A to Initiator mode if it is reselected, or to Target mode if it is selected.

Disconnect Instruction

The LSI53C895A disconnects from the SCSI bus by deasserting all SCSI signal outputs.

Wait Select Instruction

If the LSI53C895A is selected, it fetches the next instruction from the address pointed to by the DMA SCRIPTS Pointer (DSP) register.

If reselected, the LSI53C895A 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 LSI53C895A to Initiator mode when it is reselected.

If the CPU sets the SIGP bit in the Interrupt Status Zero (ISTAT0) register, the LSI53C895A 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 LSI53C895A 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 LSI53C895A wins arbitration, it attempts to select the SCSI device whose ID is defined in the destination ID field of the instruction. Once the LSI53C895A 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 LSI53C895A 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 LSI53C895A to Initiator mode if it is reselected, or to 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 LSI53C895A 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 LSI53C895A receives a Disconnect Message or a Command Complete Message.

Wait Reselect Instruction

If the LSI53C895A 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 LSI53C895A to Target mode when it is selected.

If the LSI53C895A 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 LSI53C895A 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. Use bits 25 and 26 individually or in combination to produce the following conditions:

Bit 25	Bit 26	Addressing Mode
0	0	Direct
0	1	Table Indirect
1	0	Relative
1	1	Table Relative

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 LSI53C895A is executing a Select instruction. When operating in 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]**

CC **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 Target mode. Setting this bit with a Set instruction configures the LSI53C895A as a Target device (this sets bit 0 of the SCSI Control Zero (SCNTL0) register). Clearing this bit with a Clear instruction configures the LSI53C895A as an Initiator device (this clears bit 0 of the SCNTL0 register).

R **Reserved** **[8:7]**

ACK **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.

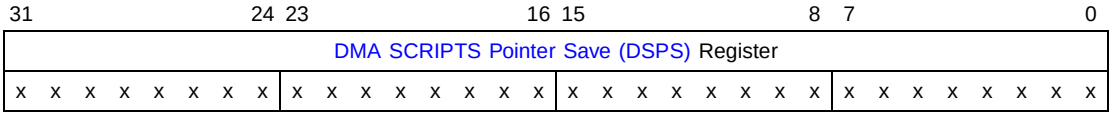
The Set instruction is used to assert SACK/ and/or SATN/ on the SCSI bus. The Clear instruction is used to deassert SACK/ and/or SATN/ on the SCSI bus. The corresponding bit in the SCSI Output Control Latch (SOCL) register is set or cleared depending on the instruction used.

Since SACK/ and SATN/ are Initiator signals, they are not asserted on the SCSI bus unless the LSI53C895A is operating as an Initiator or the SCSI Loopback Enable bit is set in the SCSI Test Two (STEST2) register.

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]**

5.4.2 Second Dword



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.

5.5 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, then stores the result back to the specified register or the SFBR. If the COM bit DMA Control (DCNTL bit 0) is cleared, Read/Write instructions cannot be used.

5.5.1 First Dword

31 30 29 27 26 24	23 22 16 15 8	7 6 0
DMA Command (DCMD) Register	DMA Byte Counter (DBC) Register	

IT[1:0]	OPC[2:0]	O[2:0]	D8	A[6:0]	ImmD	A7	Reserved - Must be 0
0 1	x x x	x x x	x	x x x x x x x	x x x x x x x x	x	0 0 0 0 0 0 0

IT[1:0] **Instruction Type - Read/Write Instruction** **[31:30]**
The configuration of the IT bits, the OpCode bits and the Operator bits define the Read/Write Instruction Type. The configuration of all these bits determine which instruction is currently selected.

OPC[2:0] **OpCode** **[29:27]**
The combinations of 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]**
These 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, 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.

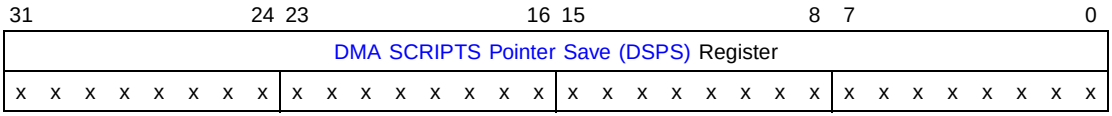
A[6:0] **Register Address - A[6:0]** **[22:16]**
It is possible to change register values from SCRIPTS in read-modify-write cycles or move to/from SFBR cycles. A[6:0] selects an 8-bit source/destination register within the LSI53C895A.

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]**

5.5.2 Second Dword



Destination Address **[31:0]**
This field contains the 32-bit destination address where the data is to move.

5.5.3 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 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.

5.5.4 Move To/From SFBR Cycles

All operations are read-modify-writes. However, two registers are involved, one of which is always the SFBR. Table 5.3 shows the possible read-modify-write operations. The possible functions of this instruction are:

- Write one byte (value contained within the SCRIPTS instruction) into any chip register.
- Move to/from the 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 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 5.3 Read/Write Instructions

Operator	OpCode 111 Read-Modify-Write	OpCode 110 Move to SFBR	OpCode 101 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[1]	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”
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”

Table 5.3 Read/Write Instructions (Cont.)

Operator	OpCode 111 Read-Modify-Write	OpCode 110 Move to SFBR	OpCode 101 Move from SFBR
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[1]	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. Data is shifted through the Carry bit and the Carry bit is shifted into the data byte.

Miscellaneous Notes:

- 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.

5.6 Transfer Control Instructions

This section describes the Transfer Control Instructions. The configuration of the OpCode bits define which Transfer Control Instruction to perform.

5.6.1 First Dword

31 30 29 27 26 24	23 22 21 20 19 18 17 16 15 8 7 0																													
DMA Command (DCMD) Register									DMA Byte Counter (DBC) Register																					
IT[1:0]	OPC[2:0]			SCSIP[2:0]			RA	R	CT	IF	JMP	CD	CP	WVP	DCM-Data Compare Mask									DCV-Data Compare Value							
1 0	x	x	x	x	x	x	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

IT[1:0] **Instruction Type - Transfer Control Instruction** **[31:30]**

The IT bit configuration (10) defines the Transfer Control Instruction Type.

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 Initiator or Target mode. Transfer Control Instructions are shown in Table 5.4.

Table 5.4 Transfer Control Instructions

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 LSI53C895A 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, then 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 LSI53C895A fetches the next instruction from the address pointed to by the DMA SCRIPTS Pointer (DSP) register, leaving the instruction pointer unchanged.

Call Instruction

The LSI53C895A 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 LSI53C895A executes a Call instruction, the instruction pointer contained in the 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 LSI53C895A fetches the next instruction from the address pointed to by the DSP register and the instruction pointer is not modified.

Return Instruction

The LSI53C895A 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 DSP register. The LSI53C895A 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 LSI53C895A fetches the next instruction from the address pointed to by the DSP register and the instruction pointer is not modified.

Interrupt Instruction

The LSI53C895A 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 LSI53C895A 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 Interrupt Service Routine to quickly identify the point at which the interrupt occurred.

The LSI53C895A halts and the DMA SCRIPTS Pointer (DSP) register must be written to before starting any further operation.

Interrupt-on-the-Fly Instruction

The LSI53C895A 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 One (ISTAT1), bit 2) is set, the LSI53C895A 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. Table 5.5 describes the possible combinations and their corresponding SCSI phase. These bits are only

valid when the LSI53C895A is operating in Initiator mode. Clear these bits when the LSI53C895A is operating in Target mode.

Table 5.5 SCSI Phase Comparisons

MSG	C/D	I/O	SCSI Phase
0	0	0	Data-Out
0	0	1	Data-In
0	1	0	Command
0	1	1	Status
1	0	0	Reserved
1	0	1	Reserved
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, 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 LSI53C895A. 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.

R **Reserved** **22**

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 One (ISTAT1), bit 2) is asserted.

JMP **Jump If True/False** **19**
This bit determines whether the LSI53C895A 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 LSI53C895A is in 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 LSI53C895A is operating in Target mode and this bit is set it tests for an active SCSI SATN/ signal.

WVP **Wait For Valid Phase** **16**

If the Wait for Valid Phase bit is set, the LSI53C895A waits for a previously unserviced phase before comparing the SCSI phase and data.

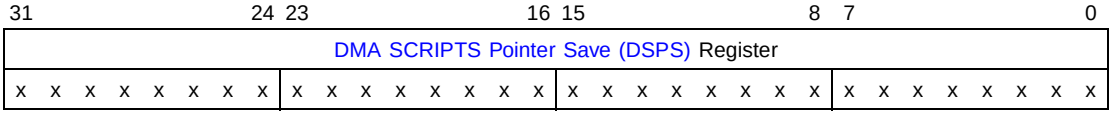
If the Wait for Valid Phase bit is cleared, the LSI53C895A compares the SCSI phase and data immediately.

DCM **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.

DCV **Data Compare Value** **[7:0]**
This 8-bit field is the data compared against the register. These bits are used in conjunction with the Data Compare Mask Field to test for a particular data value.

5.6.2 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 LSI53C895A 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.

5.7 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.

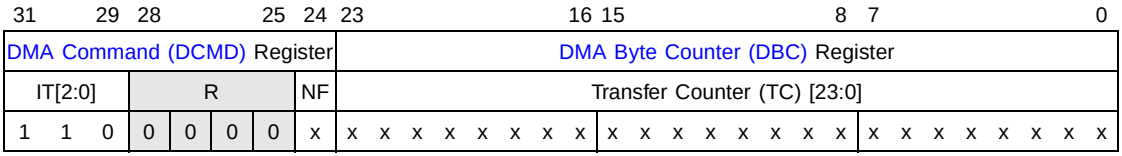
Allowing the LSI53C895A to perform memory moves frees the system processor for other tasks and moves data at higher speeds than available from current DMA controllers. Up to 16 Mbytes may be transferred with one instruction. There are two restrictions:

- Both the source and destination addresses must start with the same address alignment A[1:0]. If the source and destination are not aligned, then an illegal instruction interrupt occurs. For the PCI Cache Line Size 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 SCRIPTS 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.

5.7.1 First Dword



IT[2:0] **Instruction Type - Memory Move** **[31:29]**
The IT bit configuration (110) defines a Memory Move Instruction Type.

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 LSI53C895A 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.

TC[23:0] **Transfer Counter** **[23:0]**
The number of bytes to transfer is stored in the lower 24 bits of the first instruction word.

5.7.2 Read/Write System Memory from SCRIPTS

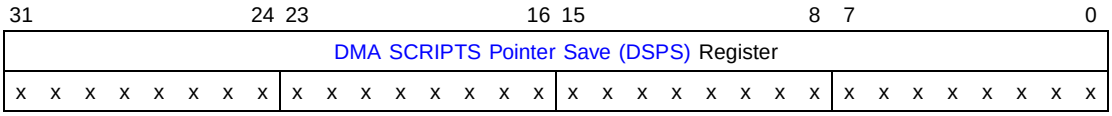
By using the Memory Move instruction, single or multiple register values are transferred to or from system memory.

Because the LSI53C895A responds to addresses as defined in the Base Address Register Zero (I/O) or Base Address Register One (MEMORY) registers, it 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 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 LSI53C895A register (for example, a SCRATCH register), and then to the SFBR.

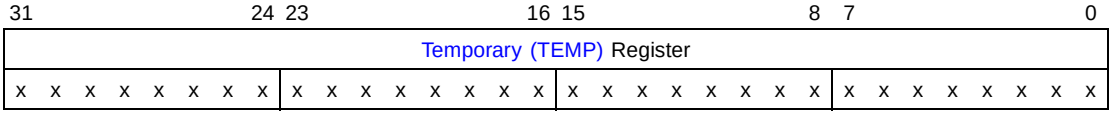
The same address alignment restrictions apply to register access operations as to normal memory-to-memory transfers.

5.7.3 Second Dword



DSPS Register **[31:0]**
These bits contain the source address of the Memory Move.

5.7.4 Third Dword



TEMP Register **[31:0]**
These bits contain the destination address for the Memory Move.

5.8 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 and Store, or the offset from the Data Structure Address (DSA), depending on the value of bit 28 (DSA Relative).

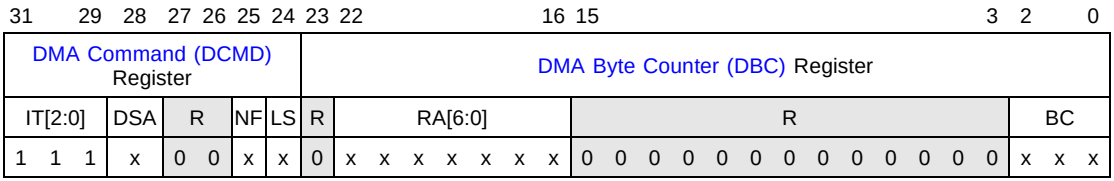
A maximum of 4 bytes may be moved with these instructions. The register address and memory address must have the same byte alignment, and the count set such that it does not cross Dword boundaries. The memory address may not map back to the chip, excluding RAM and ROM. If it does, a PCI read/write cycle occurs (the data does not actually transfer to/from the chip), and the chip issues an interrupt (Illegal Instruction Detected) immediately following.

Bit A1	Bit A0	Number of Bytes Allowed to Load and Store
0	0	One, two, three or four
0	1	One, two, or three
1	0	One or two
1	1	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 and 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

5.8.1 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 and Store to/from. When this bit is set, the chip determines the memory address to perform the Load and 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 LSI53C895A 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.

LS **Load and Store** **24**
When this bit is set, the instruction is a Load. When cleared, it is a Store.

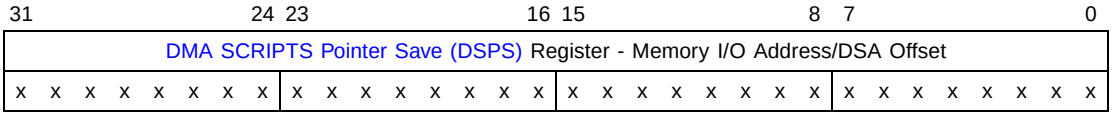
R **Reserved** **23**

RA[6:0] **Register Address** **[22:16]**
A[6:0] selects the register to Load and Store to/from within the LSI53C895A.

R **Reserved** **[15:3]**

BC **Byte Count** **[2:0]**
This value is the number of bytes to Load and Store.

5.8.2 Second Dword



Memory I/O Address / DSA Offset **[31:0]**
This is the actual memory location of where to Load and Store, or the offset from the Data Structure Address (DSA) register value.

Chapter 6
Electrical Specifications

This section specifies the LSI53C895A electrical and mechanical characteristics. It is divided into the following sections:

6.1 DC Characteristics

This section of the manual describes the LSI53C895A DC characteristics. Tables 6.1 through 6.13 give current and voltage specifications. Figures 6.1 and 6.2 are driver schematics.

Table 6.1 Absolute Maximum Stress Ratings

Symbol	Parameter	Min	Max	Unit	Test Conditions
T_{STG}	Storage temperature	–55	150	°C	–
V_{DD}	Supply voltage	–0.5	4.5	V	–
V_{IN}	Input voltage	V_{SS} –0.3	V_{DD} +0.3	V	
V_{IN5V}	Input voltage (5 V tolerant pins)	V_{SS} –0.3	5.25	V	–
I_{LP}[1]	Latch-up current	±150	–	mA	–
ESD	Electrostatic discharge	–	2 K	V	MIL-STD 883C, Method 3015.7

1. $-2\ V < V_{PIN} < 8\ V$.

Note: 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.

Table 6.2 Operating Conditions

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{DD}	Supply voltage	2.97	3.63	V	±10%
I_{DD}	Supply current (dynamic) Supply current (static)	– –	200 1	mA mA	– –
$I_{DD\text{-}I/O}$	LVD Mode Supply Current (dynamic)	–	600	mA	–
T_A	Operating free air	0	70	°C	–
θ_{JA}	Thermal resistance (junction to ambient air)	–	25 40	°C/W °C/W	272 BGA 208 PQFP

Note: Conditions that exceed the operating limits may cause the device to function incorrectly.

Table 6.3 LVD Driver SCSI Signals—SD[15:0]+, SDP[1:0]/, SREQ/, SREQ2/, SACK/, SACK2/, SMSG/, SIO/, SCD/, SATN/, SBSY/, SSEL/, SRST/

Symbol	Parameter	Min	Max	Unit	Test Conditions[1]
I_O+	Source (+) current	7	13	mA	Asserted state
I_O-	Sink (–) current	–7	–13	mA	Asserted state
I_O+	Source (+) current	3.5	–6.5	mA	Negated state
I_O-	Sink (–) current	–3.5	6.5	mA	Negated state
I_{OZ}	3-state leakage	–20	20	µA	V_{PIN} = 0 V, 3.47 V
$I_{OZ\ (SRST\text{-}only)}$	3-state leakage	–500	–50	µA	–

1. V_{CM} = 0.7–1.8 V, R_L = 0–110 Ω, R_{bias} = 9.76 kΩ.

Figure 6.1 LVD Driver

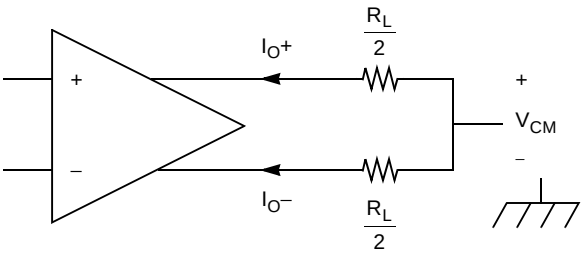


Table 6.4 LVD Receiver SCSI Signals—SD[15:0]/, SDP[1:0]/, SREQ/, SREQ2/, SACK/, SACK2/, SMSG/, SIO/, SCD/, SATN/, SBSY/, SSEL/, SRST/

Symbol	Parameter	Min	Max	Unit
V_I	LVD receiver voltage asserting	60	–	mV
V_I	LVD receiver voltage negating	–	–60	mV

Note: V_{CM} = 0.7–1.8 V.

Figure 6.2 LVD Receiver

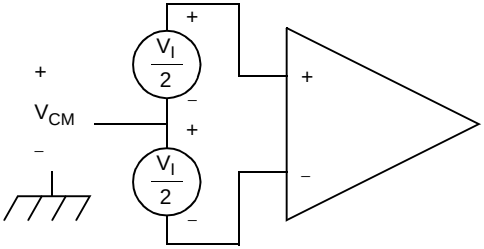


Table 6.5 DIFFSENS SCSI Signal

Symbol	Parameter	Min	Max	Unit	Test Conditions[1]
V_{IH}	HVD sense voltage	2.4	5.5	V	Note 1
V_S	LVD sense voltage	0.7	1.9	V	Note 1
V_{IL}	SE sense voltage	V_{SS} –0.3	0.5	V	Note 1
I_{IN}	Input leakage	–10	10	µA	V_{PIN} = 0 V, 5.25 V

1. Functional test specified for each mode (V_{IH}, V_S, V_{IL}).

Table 6.6 Input Capacitance

Symbol	Parameter	Min	Max	Unit	Test Conditions
C_I	Input capacitance of input pads	–	7	pF	–
C_{IO}	Input capacitance of I/O pads	–	15	pF	–

Table 6.7 Bidirectional Signals—MAD[7:0], MAS/[1:0], MCE/, MOE/, MWE/

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IH}	Input high voltage	2.0	5.25	V	–
V_{IL}	Input low voltage	V_{SS} –0.3	0.8	V	–
V_{OH}	Output high voltage	2.4	V_{DD}	V	–4 mA dynamic
V_{OL}	Output low voltage	V_{SS}	0.4	V	4 mA dynamic
I_{OZ}	3-state leakage	–10	10	µA	V_{PIN} = 0 V, 5.25 V
I_{PULL}	Pull-down current	7.5	75	µA	–

Table 6.8 Bidirectional Signals—GPIO0_FETCH/, GPIO1_MASTER/, GPIO[2:8]

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IH}	Input high voltage	2.0	5.25	V	–
V_{IL}	Input low voltage	V_{SS} –0.3	0.8	V	–
V_{OH}	Output high voltage	2.4	V_{DD}	V	–8 mA dynamic
V_{OL}	Output low voltage	V_{SS}	0.4	V	8 mA dynamic
I_{OZ}	3-state leakage	–10	10	µA	V_{PIN} = 0 V, 5.25 V
I_{PULL}	Pull-down current	7.5	75	µA	–

Table 6.9 Bidirectional Signals—AD[31:0], C_BE[3:0]/, FRAME/, IRDY/, TRDY/, DEVSEL/, STOP/, PERR/, PAR

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IH}	Input high voltage	0.5 V_{DD}	5.25	V	–
V_{IL}	Input low voltage	V_{SS} –0.3	0.3 V_{DD}	V	–
V_{OH}	Output high voltage	0.9 V_{DD}	–	V	–500 µA
V_{OL}	Output low voltage	–	0.1 V_{DD}	V	1500 µA
V_{OH}	5 V tolerant output high voltage	2.4	–	V	–16 mA
V_{OL}	5 V tolerant output low voltage	–	0.55	V	16 mA
I_{OZ}	3-state leakage	–10	10	µA	V_{PIN} = 0 V, 5.25 V
I_{PULL}	Pull-down current	7.5	75	µA	–

Table 6.10 Input Signals—CLK, GNT/, IDSEL, RST/, SCLK, TCK, TDI, TEST_HSC[1], TEST_RST, TMS, TRST/

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IH}	Input high voltage	0.5 V_{DD}	5.0	V	–
V_{IL}	Input low voltage	V_{SS} –0.3	0.3 V_{DD}	V	–
I_{IN}	Input leakage[2]	–10	10	µA	V_{PIN} = 0 V, 5.25 V
$I_{PULL-UP}$[3]	Pull-up current - only on TCK, TDI, TEST_HSC, TEST_RST, TMS, TRST/	–75	–7.5	µA	–

1. TEST_HSC has a pull-down.
2. The Input leakage test does not apply to the TEST_RST/ pin with V_{PIN} = 0 V.
3. Pull-up spec does not apply to: SCLK, CLK, GNT/, IDSEL, and RST/.

Table 6.11 Output Signal—TDO

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{OH}	Output high voltage	2.4	V_{DD}	V	–4 mA dynamic
V_{OL}	Output low voltage	V_{SS}	0.4	V	4 mA dynamic
I_{OZ}	3-state leakage	–10	10	µA	V_{PIN} = 0 V, 5.25 V

Table 6.12 Output Signals—ALT_IRQ/, IRQ/, MAC/_TESTOUT, REQ/

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{OH}	Output high voltage	0.9 V_{DD}	–	V	–500 µA
V_{OL}	Output low voltage	–	0.1 V_{DD}	V	1500 µA
V_{OH}	5 V tolerant output high voltage	2.4	–	V	–16 mA
V_{OL}	5 V tolerant output low voltage	–	0.55	V	16 mA
I_{OZ}	3-state leakage	–10	10	µA	V_{PIN} = 0 V, 5.25 V
I_{PULL}	Pull-down current - only on ALT_IRQ and IRQ/	7.5	75	µA	–

Table 6.13 Output Signal—SERR/

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{OL}	Output low voltage	V_{SS}	0.1 V_{DD}	V	16 mA
I_{OZ}	3-state leakage	–10	10	µA	–

6.2 TolerANT Technology Electrical Characteristics

The LSI53C895A features TolerANT technology, which includes active negation on the SCSI drivers and input signal filtering on the SCSI receivers. Active negation actively drives the SCSI Request, Acknowledge, Data, and Parity signals HIGH rather than allowing them to be passively pulled up by terminators. Table 6.14 provides electrical characteristics for SE SCSI signals. Figures 6.3 through 6.7 provide reference information for testing SCSI signals.

Table 6.14 TolerANT Technology Electrical Characteristics for SE SCSI Signals

Symbol	Parameter	Min[1]	Max	Unit	Test Conditions
V_{OH}[2]	Output high voltage	2.0	V_{DD}	V	I_{OH} = –7 mA
V_{OL}	Output low voltage	V_{SS}	0.5	V	I_{OL} = 48 mA
V_{IH}	Input high voltage	2.0	V_{DD} +0.3	V	–
V_{IL}	Input low voltage	V_{SS} –0.3	0.8	V	Referenced to V_{SS}
V_{IK}	Input clamp voltage	–0.66	–0.77	V	V_{DD} = 4.75; I_I = –20 mA
V_{TH}	Threshold, HIGH to LOW	1.0	1.2	V	–
V_{TL}	Threshold, LOW to HIGH	1.4	1.6	V	–
V_{TH}–V_{TL}	Hysteresis	300	500	mV	–
I_{OH}[2]	Output high current	2.5	24	mA	V_{OH} = 2.5 V
I_{OL}	Output low current	100	200	mA	V_{OL} = 0.5 V
I_{OSH}[2]	Short-circuit output high current	–	625	mA	Output driving low, pin shorted to V_{DD} supply[3]
I_{OSL}	Short-circuit output low current	–	95	mA	Output driving high, pin shorted to V_{SS} supply
I_{LH}	Input high leakage	–	20	µA	V_{DD} ±5%, V_{PIN} = 2.7 V
I_{LL}	Input low leakage	–20	–	µA	V_{DD} ±5%, V_{PIN} = 0 V
I_{PD}	Power down leakage	–	20		V_{DD} = 0 V, V_{PIN} = 1.2 V
R_I	Input resistance	20	–	MΩ	SCSI pins[4]
C_P	Capacitance per pin	–	15	pF	PQFP
t_R[2]	Rise time, 10% to 90%	4.0	18.5	ns	Figure 6.3

Table 6.14 TolerANT Technology Electrical Characteristics for SE SCSI Signals (Cont.)

Symbol	Parameter	Min[1]	Max	Unit	Test Conditions
t_F	Fall time, 90% to 10%	4.0	18.5	ns	Figure 6.3
dV_H/dt	Slew rate, LOW to HIGH	0.15	0.50	V/ns	Figure 6.3
dV_L/dt	Slew rate, HIGH to LOW	0.15	0.50	V/ns	Figure 6.3
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/.
3. Single pin only; irreversible damage may occur if sustained for one second.
4. SCSI RESET pin has 10 kΩ pull-up resistor.

Figure 6.3 Rise and Fall Time Test Condition

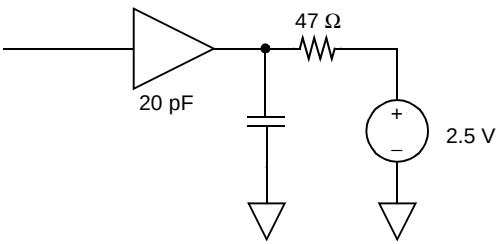
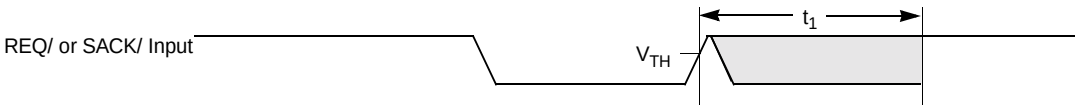


Figure 6.4 SCSI Input Filtering



Note: t_1 is the input filtering period.

Figure 6.5 Hysteresis of SCSI Receivers

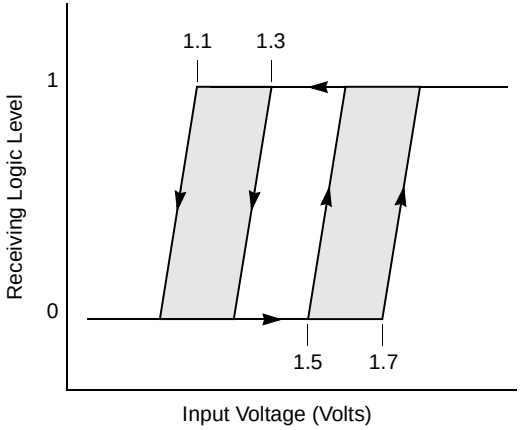


Figure 6.6 Input Current as a Function of Input Voltage

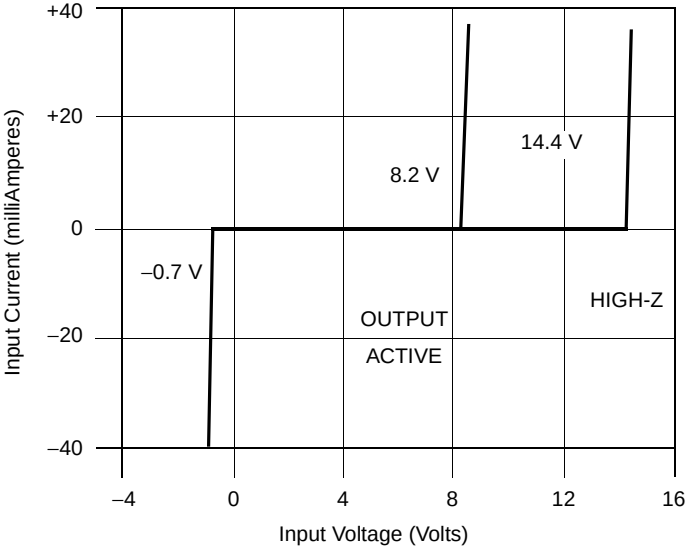
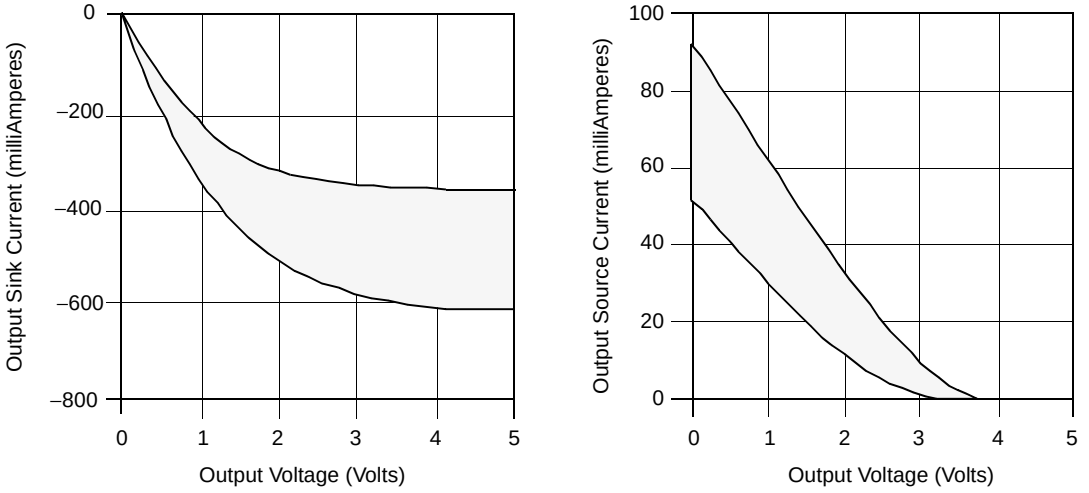


Figure 6.7 Output Current as a Function of Output Voltage



6.3 AC Characteristics

The AC characteristics described in this section apply over the entire range of operating conditions (refer to the DC Characteristics section). Chip timings are based on simulation at worst case voltage, temperature, and processing. Timing was developed with a load capacitance of 50 pF. Table 6.15 and Figure 6.8 provide External Clock timing data.

Table 6.15 External Clock[1]

Symbol	Parameter	Min	Max	Unit
t_1	Bus clock cycle time	30	DC	ns
	SCSI clock cycle time (SCLK)[2]	25	60	ns
t_2	CLK LOW time[3]	10	–	ns
	SCLK LOW time[3]	6	33	ns
t_3	CLK HIGH time[3]	12	–	ns
	SCLK HIGH time[3]	10	33	ns
t_4	CLK slew rate	1	–	V/ns
	SCLK slew rate	1	–	V/ns

1. Timings are for an external 40 MHz clock. A quadrupled 40 MHz clock is required for Ultra1 SCSI operation.
2. This parameter must be met to ensure SCSI timings are within specification.
3. Duty cycle not to exceed 60/40.

Figure 6.8 External Clock

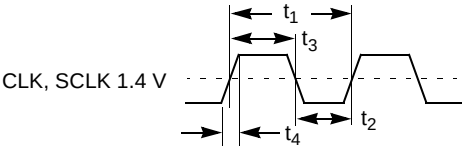
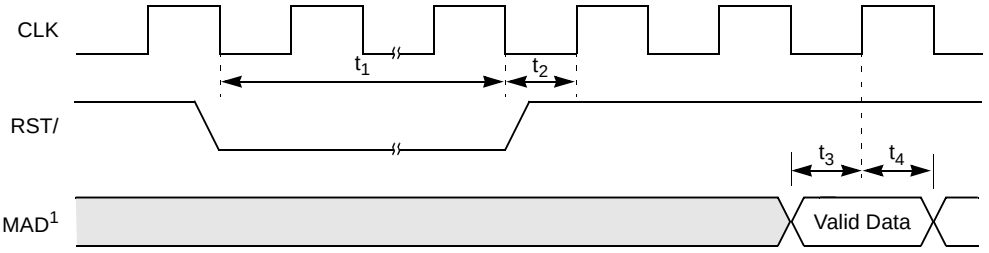


Table 6.16 and Figure 6.9 provide Reset Input timing data.

Table 6.16 Reset Input

Symbol	Parameter	Min	Max	Unit
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



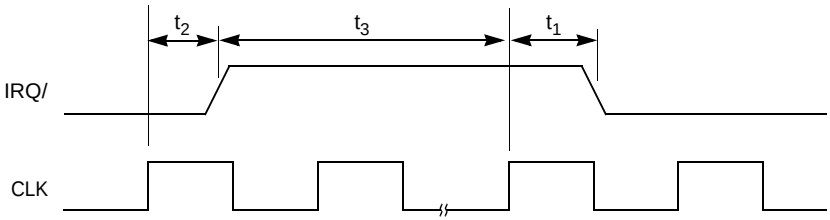
Note:
1. When enabled.

Table 6.17 and Figure 6.10 provide Interrupt Output timing data.

Table 6.17 Interrupt Output

Symbol	Parameter	Min	Max	Unit
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

Figures 6.11 through 6.34 represent signal activity when the LSI53C895A 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 Register/SCRIPTS RAM Read
  - 64-Bit Address Operating Register/SCRIPTS RAM Read

  - 32-Bit Operating Register/SCRIPTS RAM Write
  - 64-Bit Address 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 32-Bit 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) Single Byte Access Write Cycle
  - Normal/Fast Memory (≥ 128 Kbytes) Multiple Byte Access Read Cycle
  - Normal/Fast Memory (≥ 128 Kbytes) Multiple Byte Access Write Cycle
  - Slow Memory (≤ 128 Kbytes) Read Cycle
  - Slow Memory (≤ 128 Kbytes) Write Cycle
  - ≤ 64 Kbytes ROM Read Cycle
  - ≤ 64 Kbyte ROM Write Cycle

6.4.1 Target Timing

Tables 6.17 through 6.23 and Figures 6.11 through 6.16 describe Target timing.

Table 6.18 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	–	11	ns

Figure 6.11 PCI Configuration Register Read

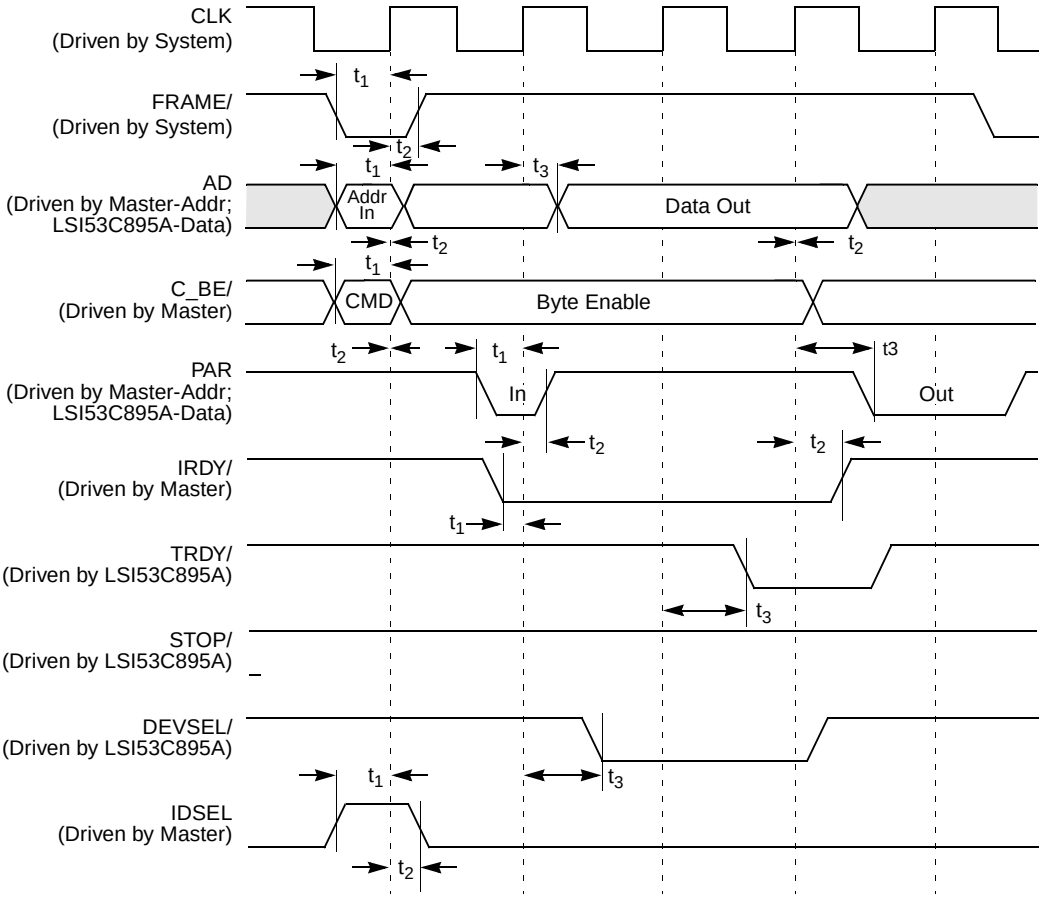


Table 6.19 PCI Configuration Register Write

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	–	11	ns

Figure 6.12 PCI Configuration Register Write

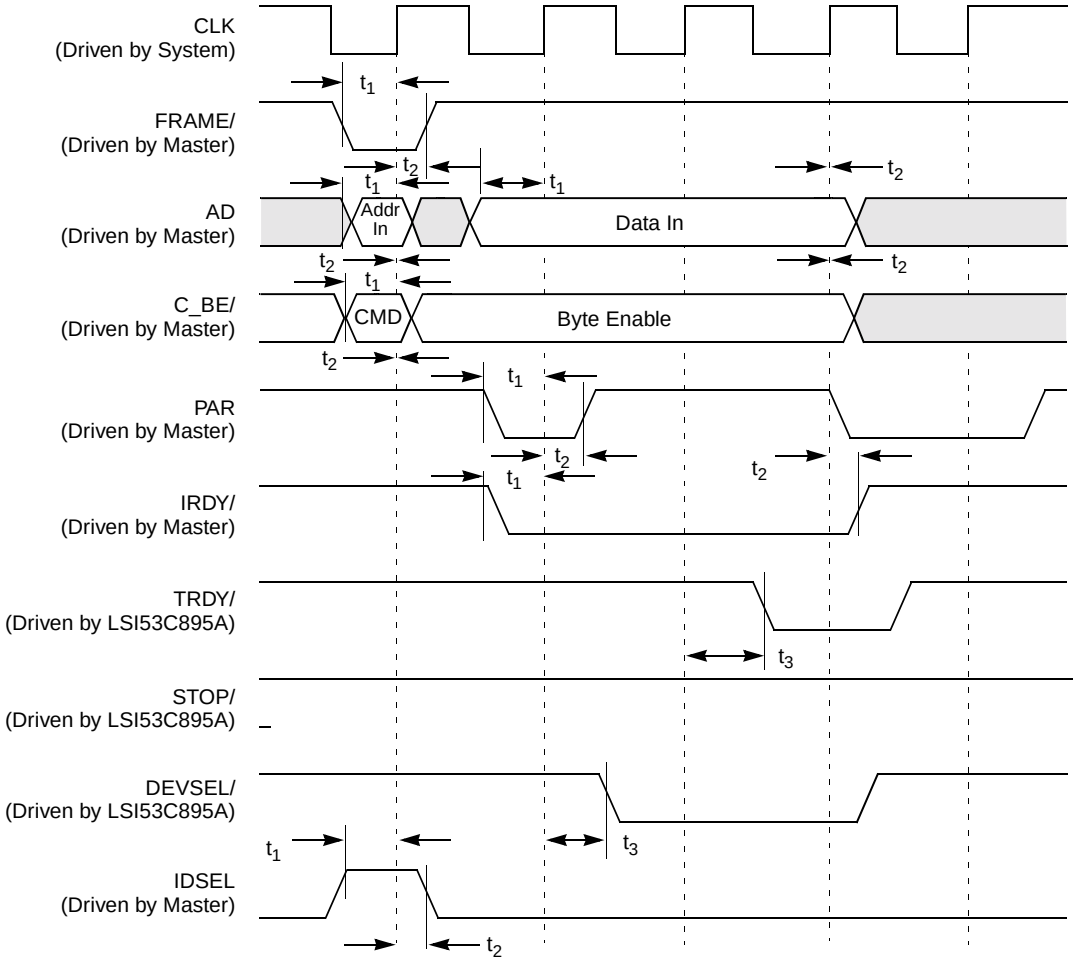


Table 6.20 32-Bit Operating Register/SCRIPTS RAM Read

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	–	11	ns

Figure 6.13 32-Bit Operating Register/SCRIPTS RAM Read

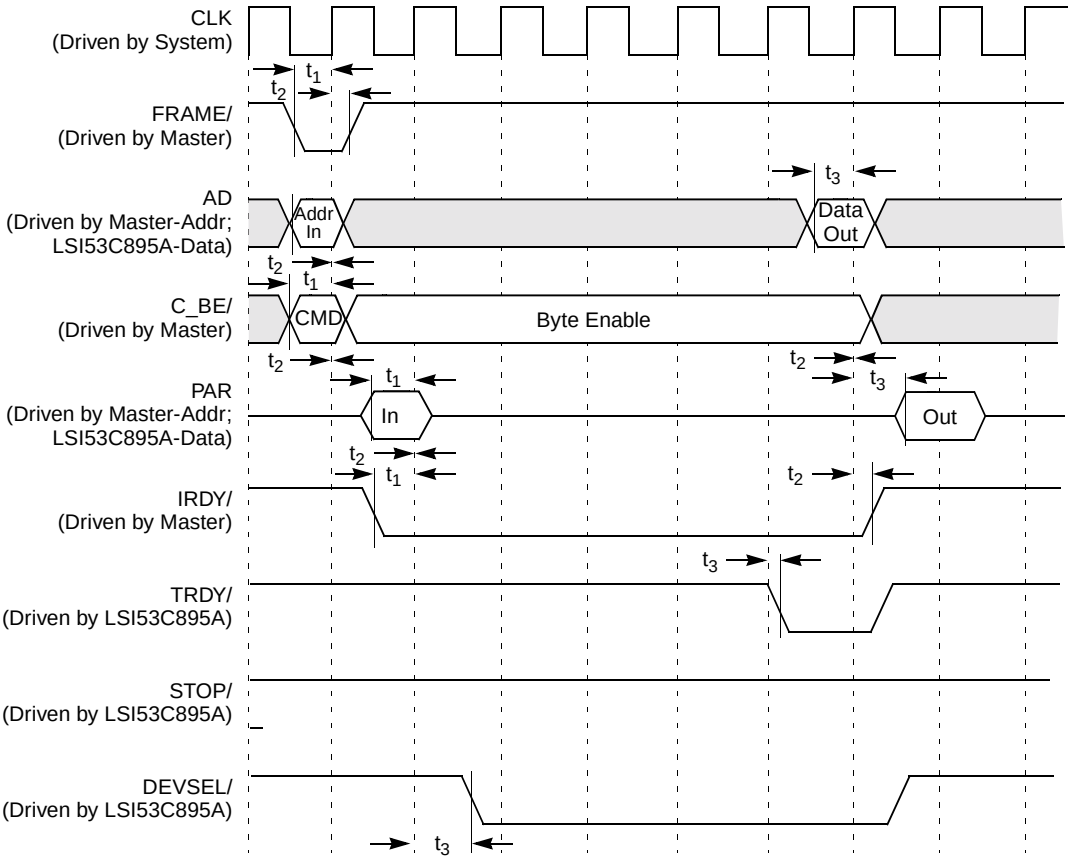


Table 6.21 64-Bit Address Operating Register/SCRIPTS RAM Read

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	–	11	ns

Figure 6.14 64-Bit Address Operating Register/SCRIPTS RAM Read

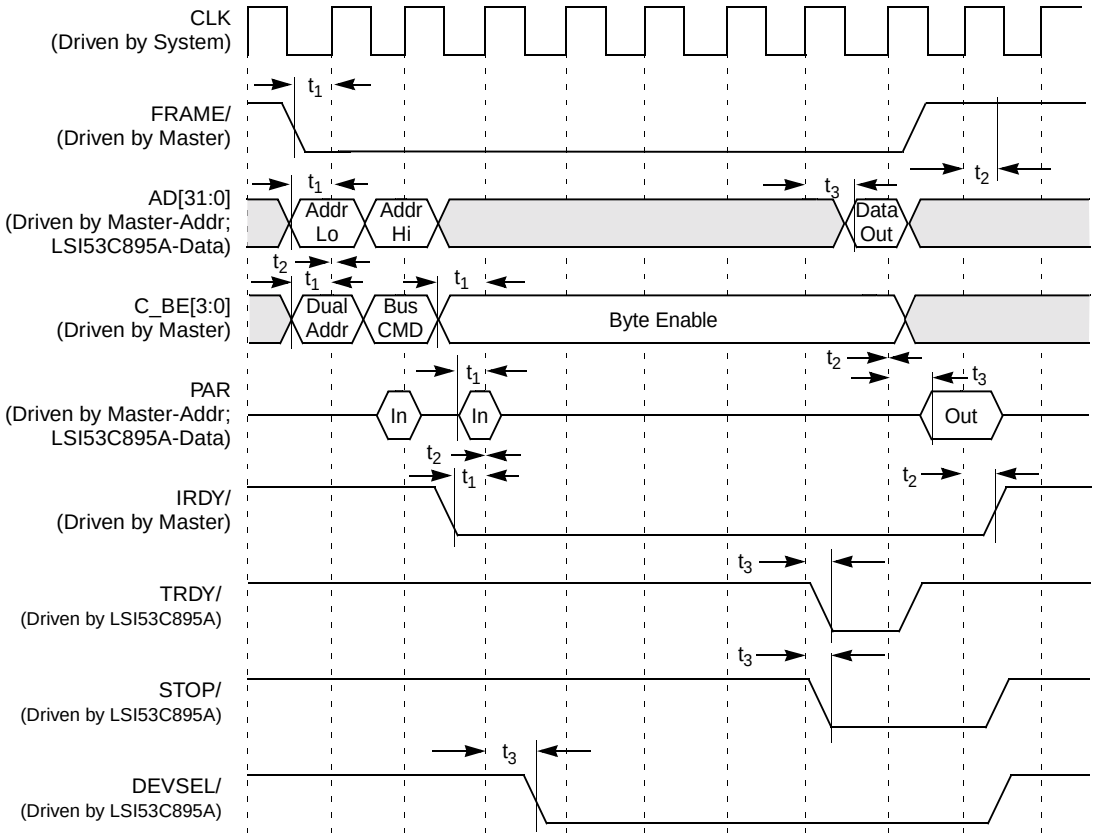


Table 6.22 32-Bit Operating Register/SCRIPTS RAM Write

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	–	11	ns

Figure 6.15 32-Bit Operating Register/SCRIPTS RAM Write

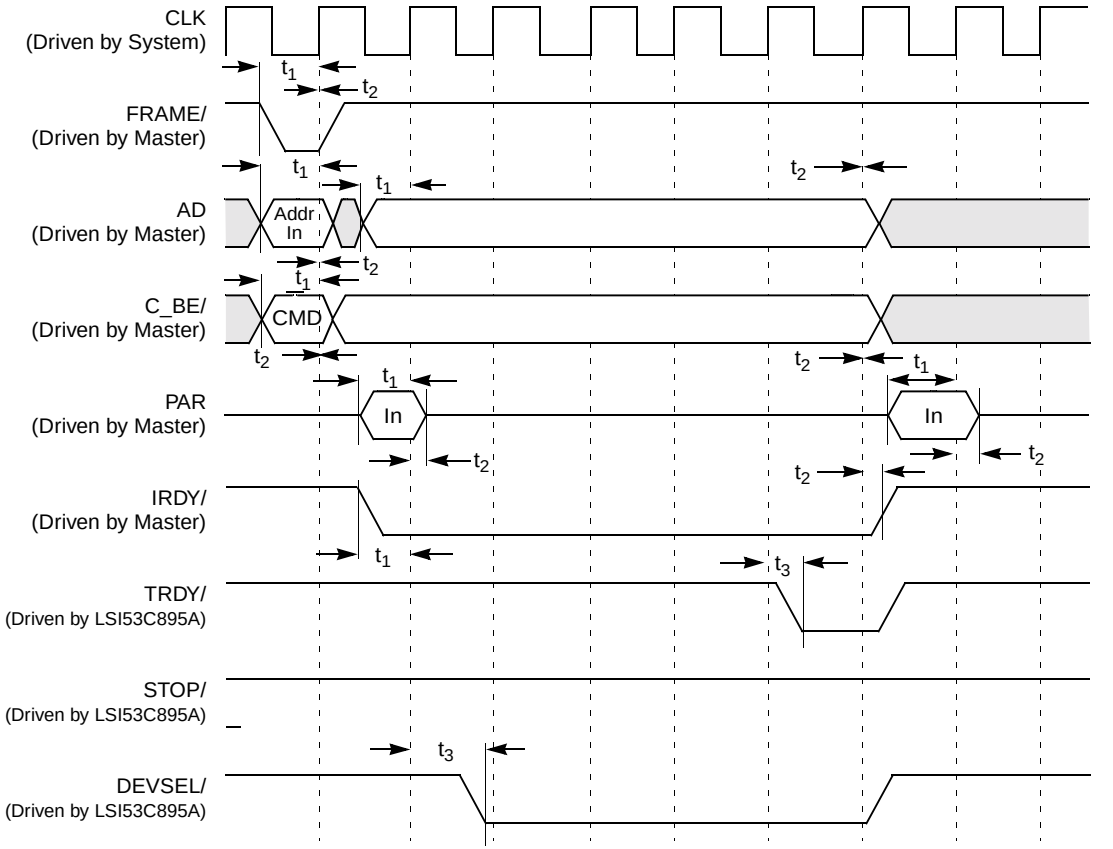
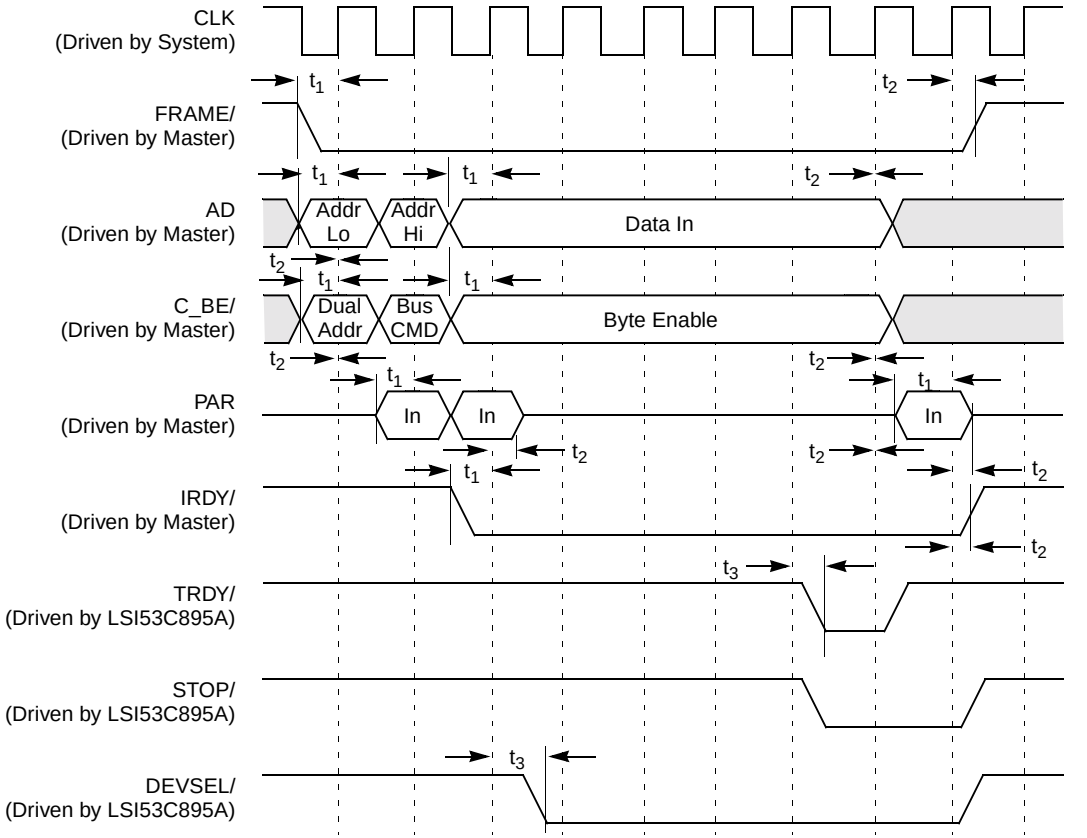


Table 6.23 64-Bit Address Operating Register/SCRIPTS RAM Write

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	–	11	ns

Figure 6.16 64-Bit Address Operating Register/SCRIPTS RAM Write



6.4.2 Initiator Timing

Tables 6.24 through 6.31 and Figures 6.17 and 6.24 describe Initiator timing.

Table 6.24 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
t_7	CLK HIGH to GPIO0_FETCH/ LOW	–	20	ns
t_8	CLK HIGH to GPIO0_FETCH/ HIGH	–	20	ns
t_9	CLK HIGh to GPIO1_MASTER/ LOW	–	20	ns
t_{10}	CLK HIGH to GPIO1_MASTER/ HIGH	–	20	ns

Figure 6.17 Nonburst Opcode Fetch, 32-Bit Address and Data

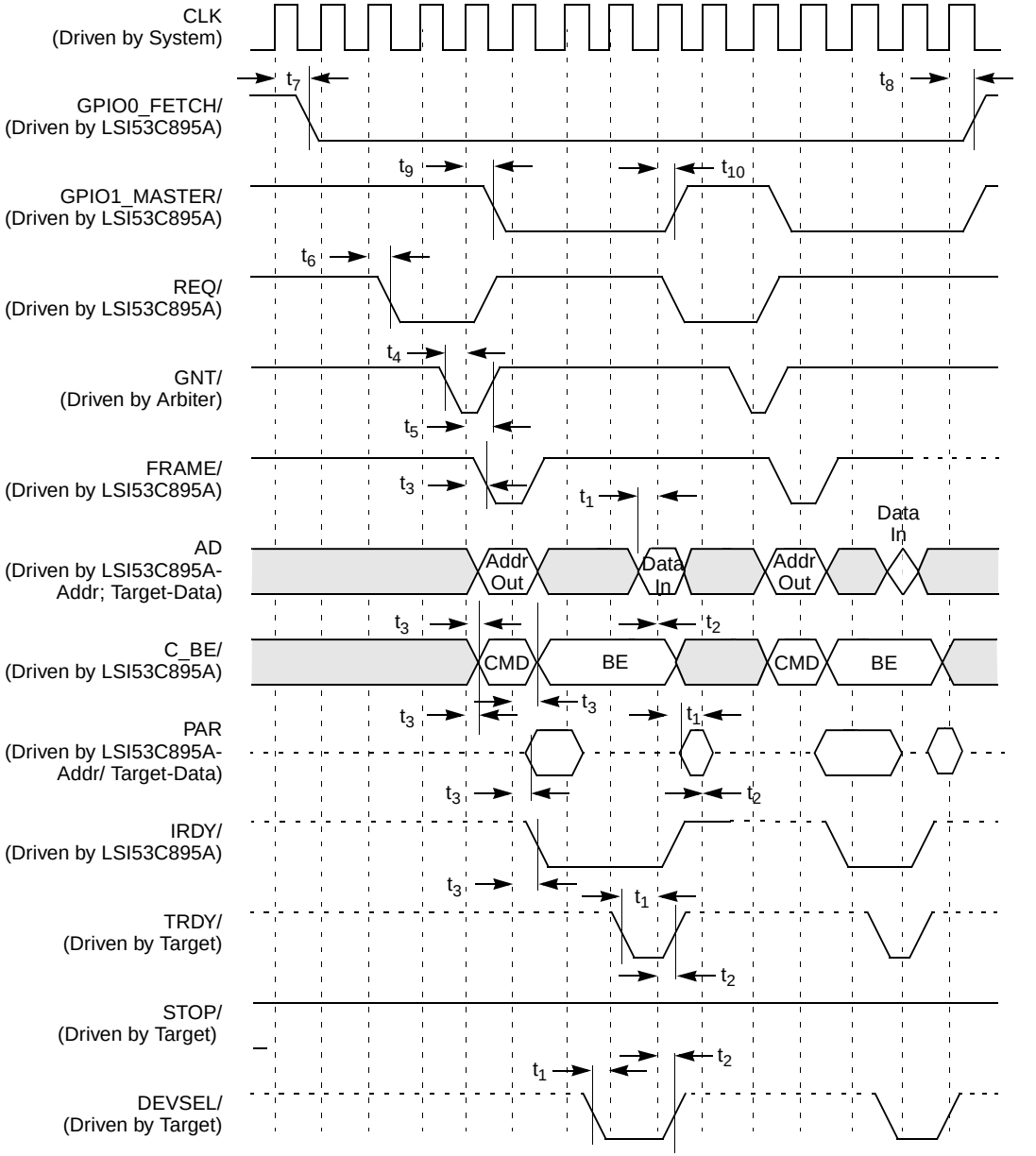


Table 6.25 Burst Opcode Fetch, 32-Bit Address and Data

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
t_7	CLK HIGH to GPIO0_FETCH/ LOW	–	20	ns
t_8	CLK HIGH to GPIO0_FETCH/ HIGH	–	20	ns
t_9	CLK HIGH to GPIO1_MASTER/ LOW	–	20	ns
t_{10}	CLK HIGH to GPIO1_MASTER/ HIGH	–	20	ns

Figure 6.18 Burst Opcode Fetch, 32-Bit Address and Data

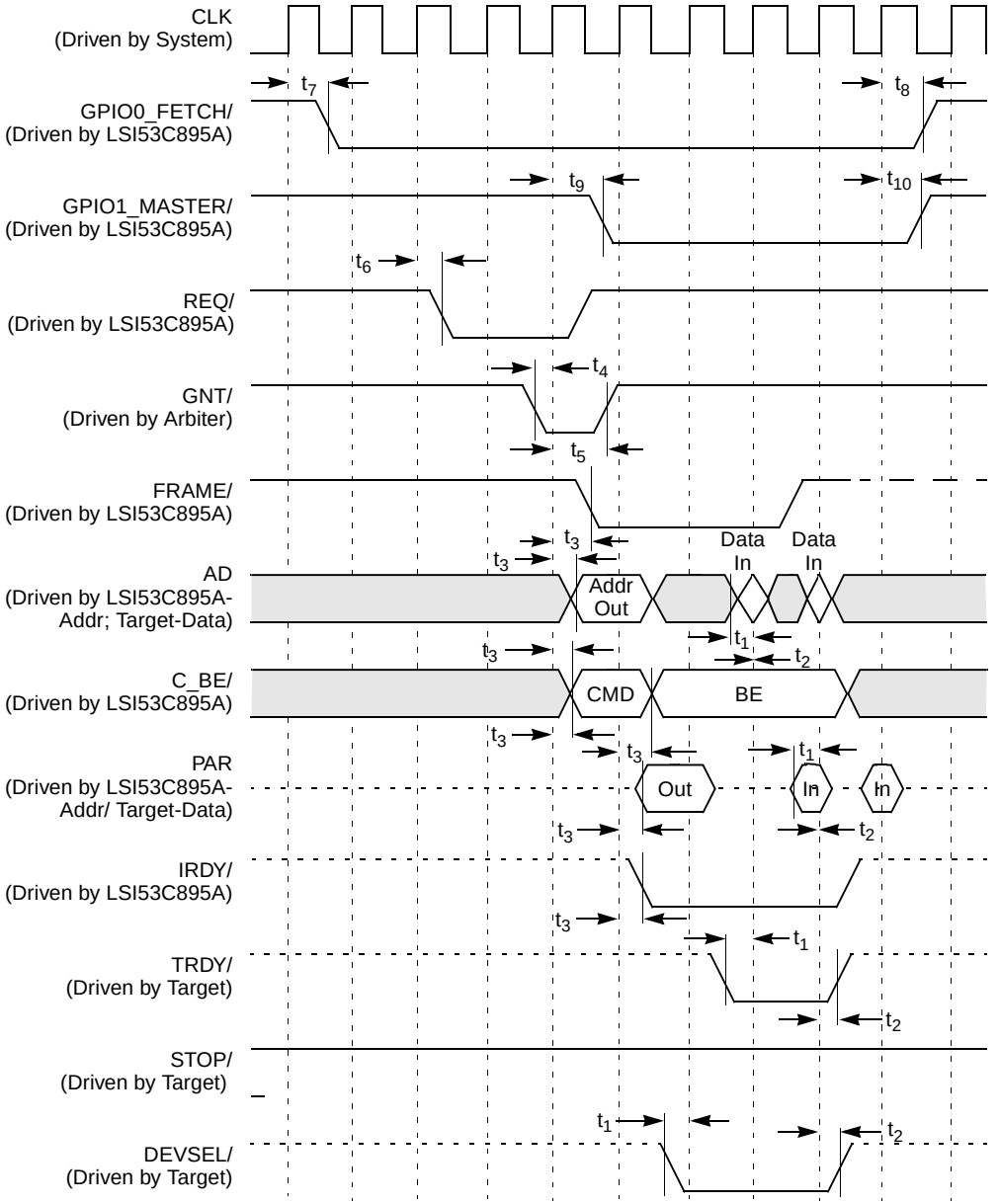


Table 6.26 Back to Back Read, 32-Bit Address and Data

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
t_9	CLK HIGH to GPIO1_MASTER/ LOW	–	20	ns
t_{10}	CLK HIGH to GPIO1_MASTER/ HIGH	–	20	ns

Figure 6.19 Back to Back Read, 32-Bit Address and Data

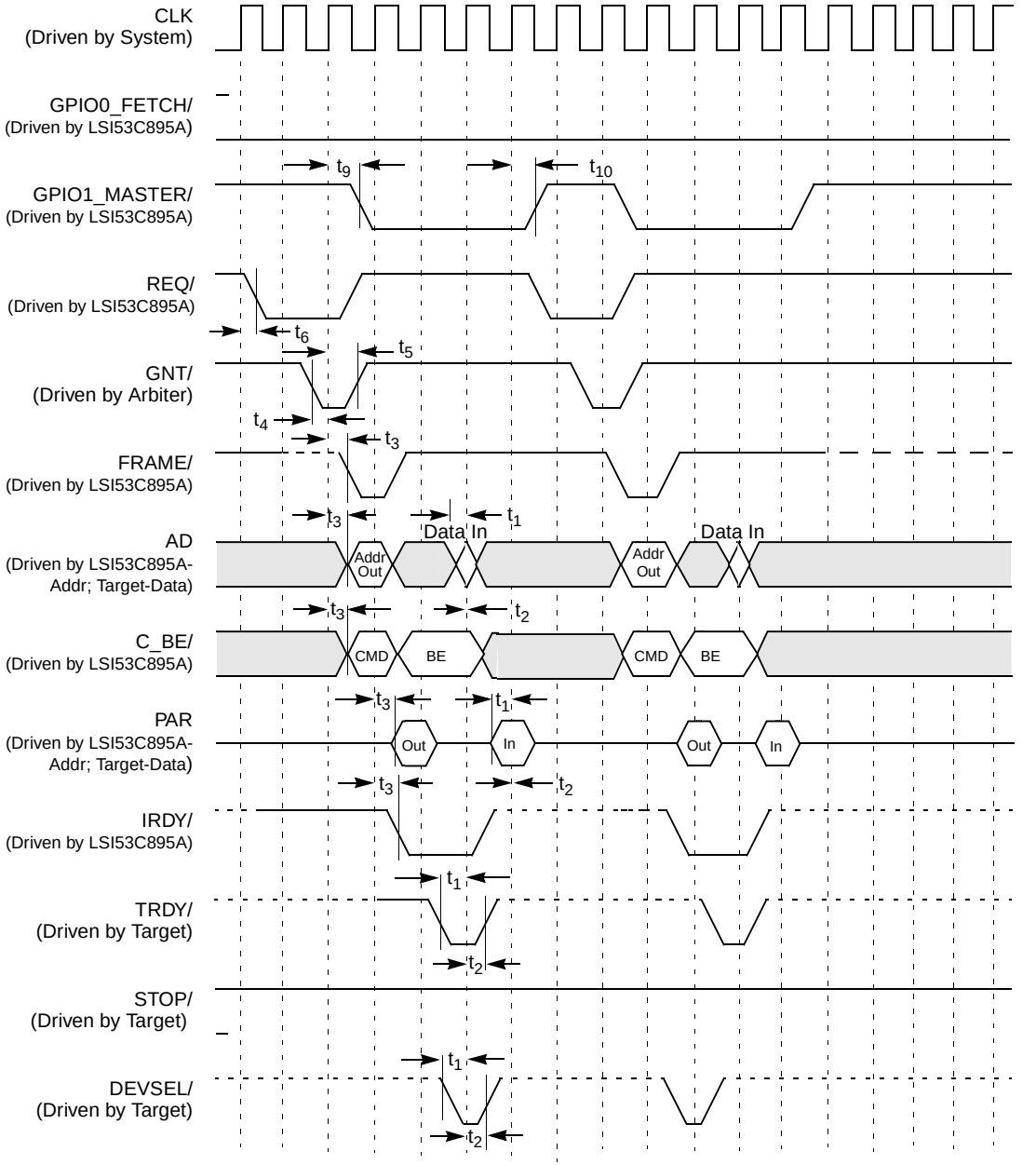


Table 6.27 Back to Back Write, 32-Bit Address and Data

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
t_9	CLK HIGH to GPIO1_MASTER/ LOW	–	20	ns
t_{10}	CLK HIGH to GPIO1_MASTER/ HIGH	–	20	ns

Figure 6.20 Back to Back Write, 32-Bit Address and Data

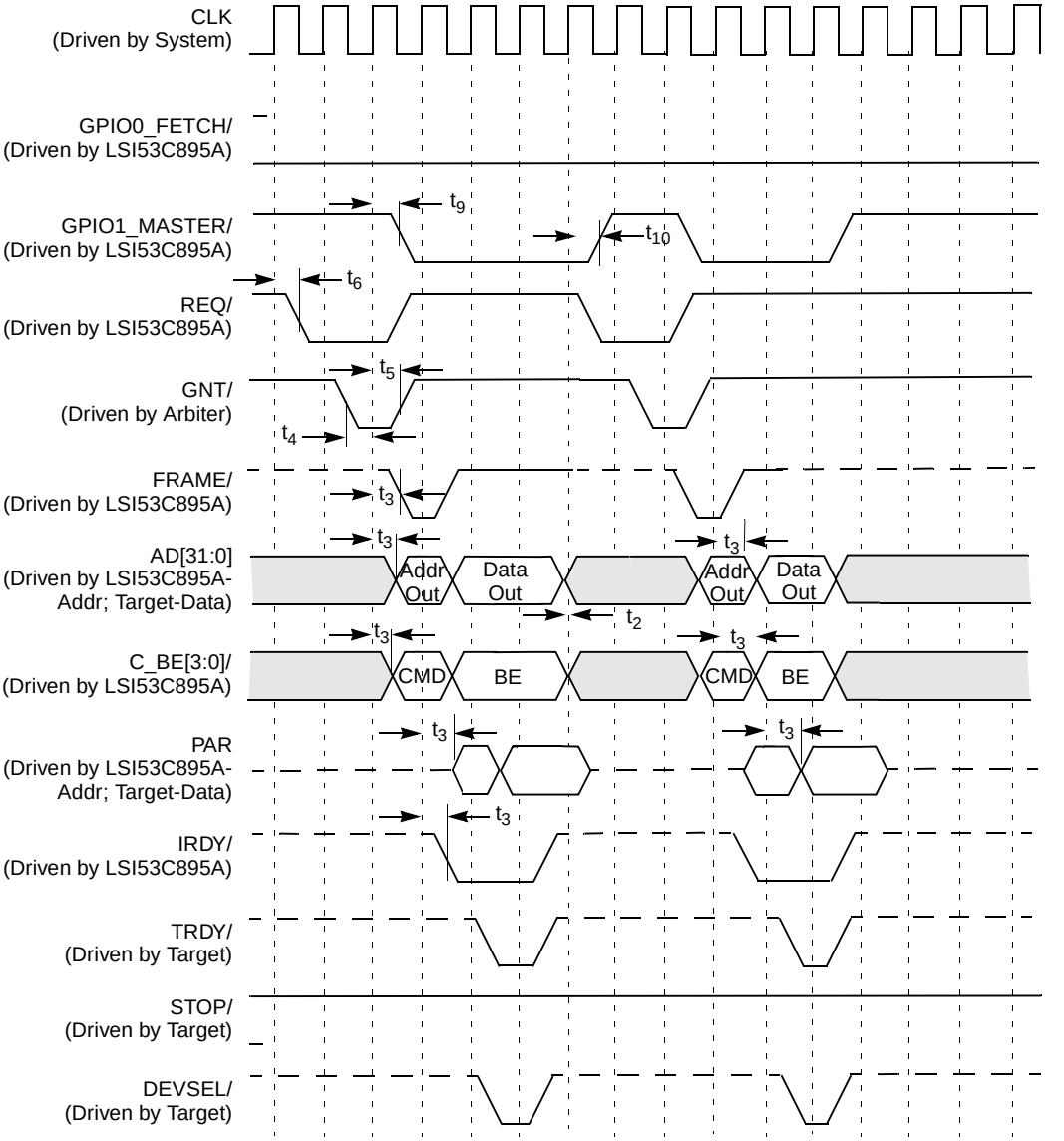


Table 6.28 Burst Read, 32-Bit Address and Data

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

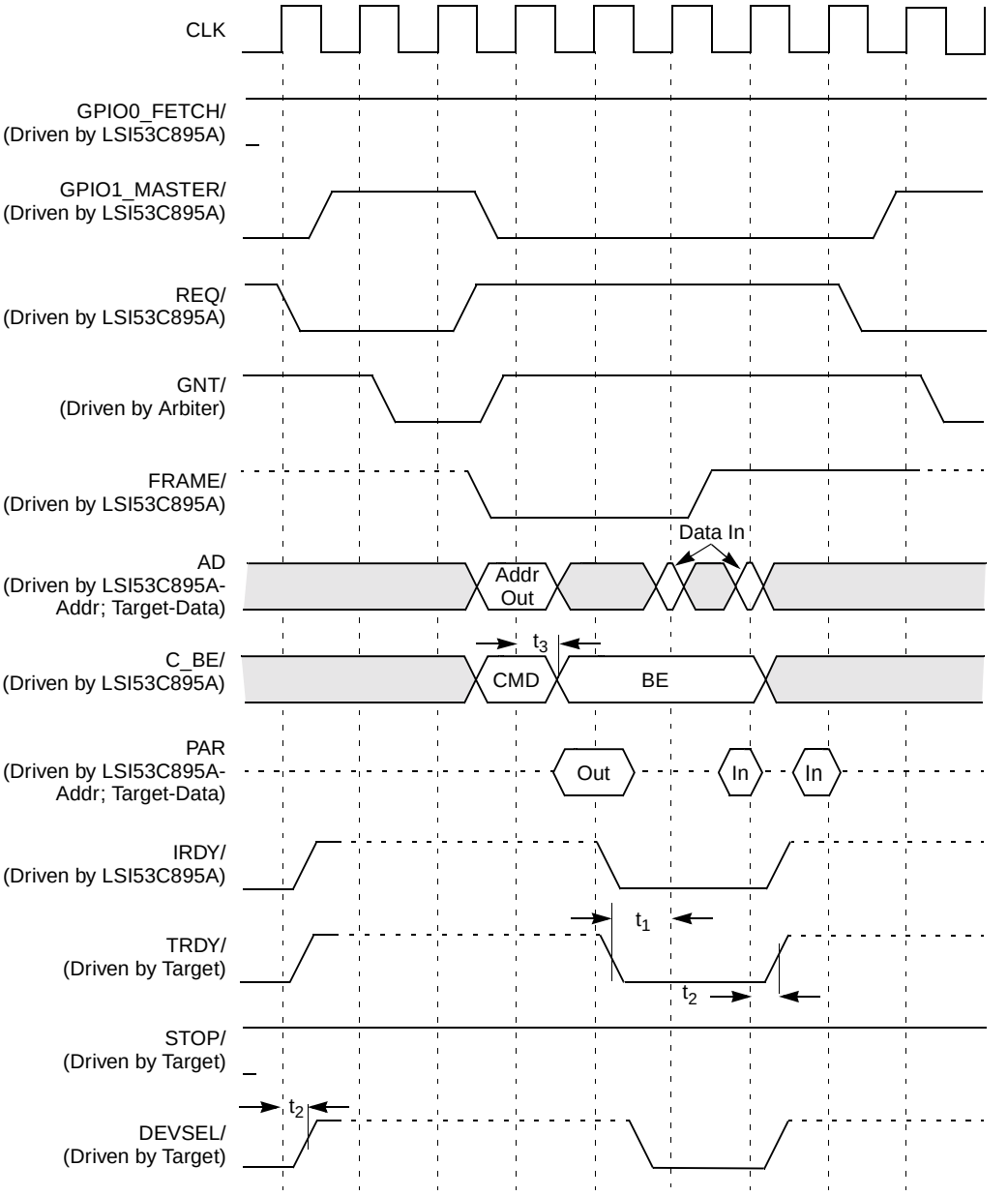


Table 6.29 Burst Read, 64-Bit Address and Data

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
t_{10}	CLK HIGH to GPIO1_MASTER/ HIGH	–	20	ns

Figure 6.22 Burst Read, 64-Bit Address and Data

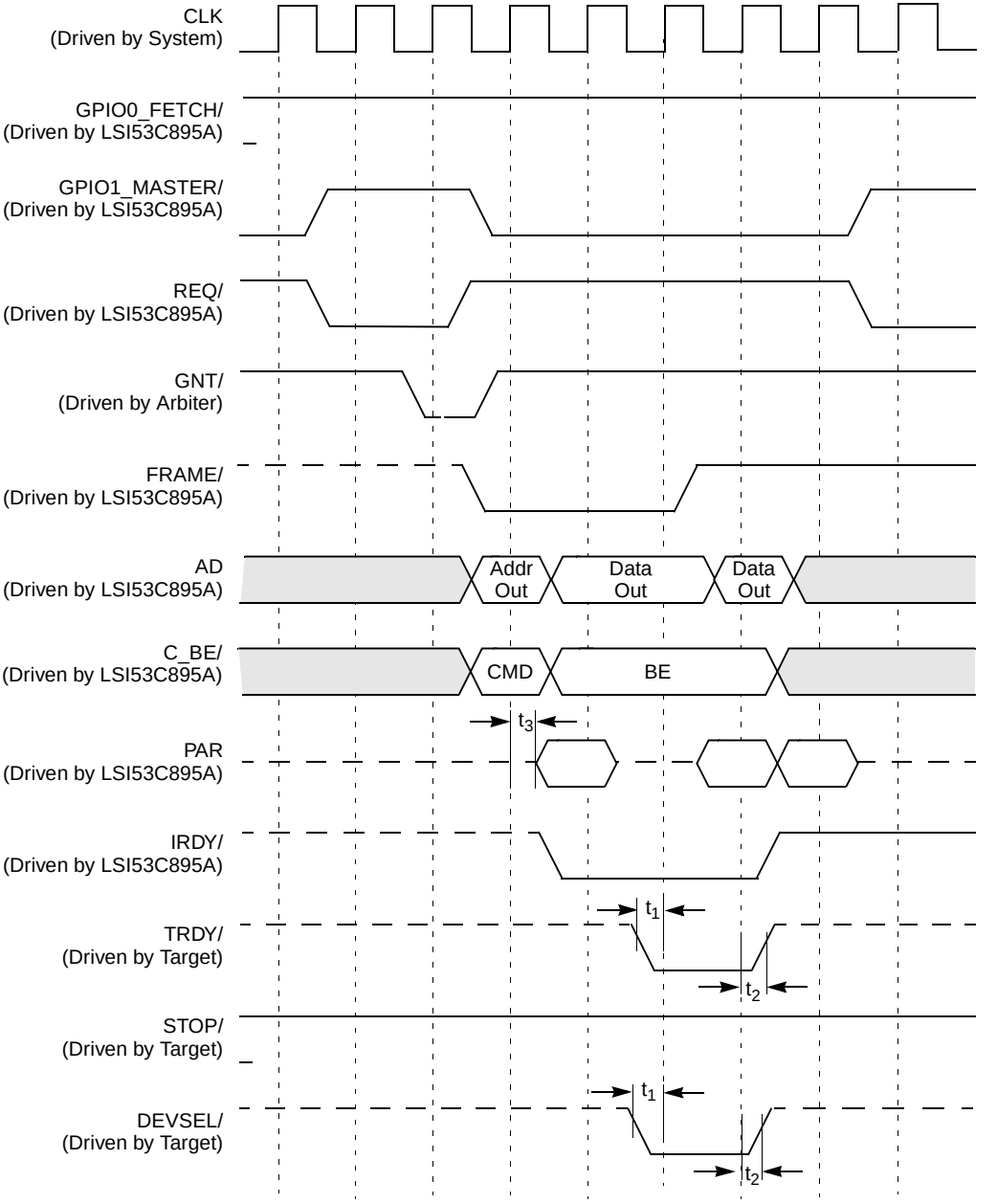


Table 6.30 Burst Write, 32-Bit Address and Data

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
t_{10}	CLK HIGH to GPIO1_MASTER/ HIGH	–	20	ns

Figure 6.23 Burst Write, 32-Bit Address and Data

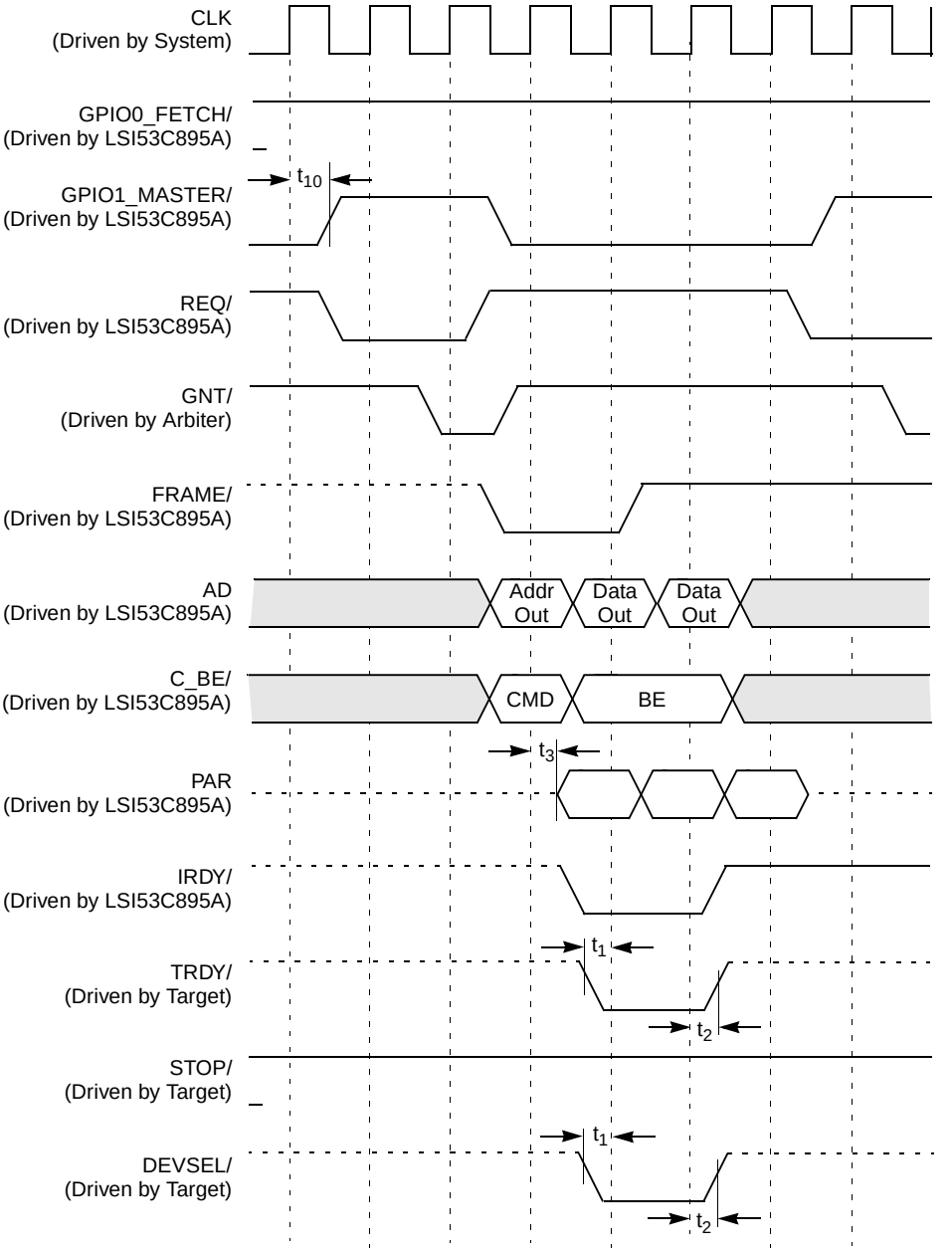
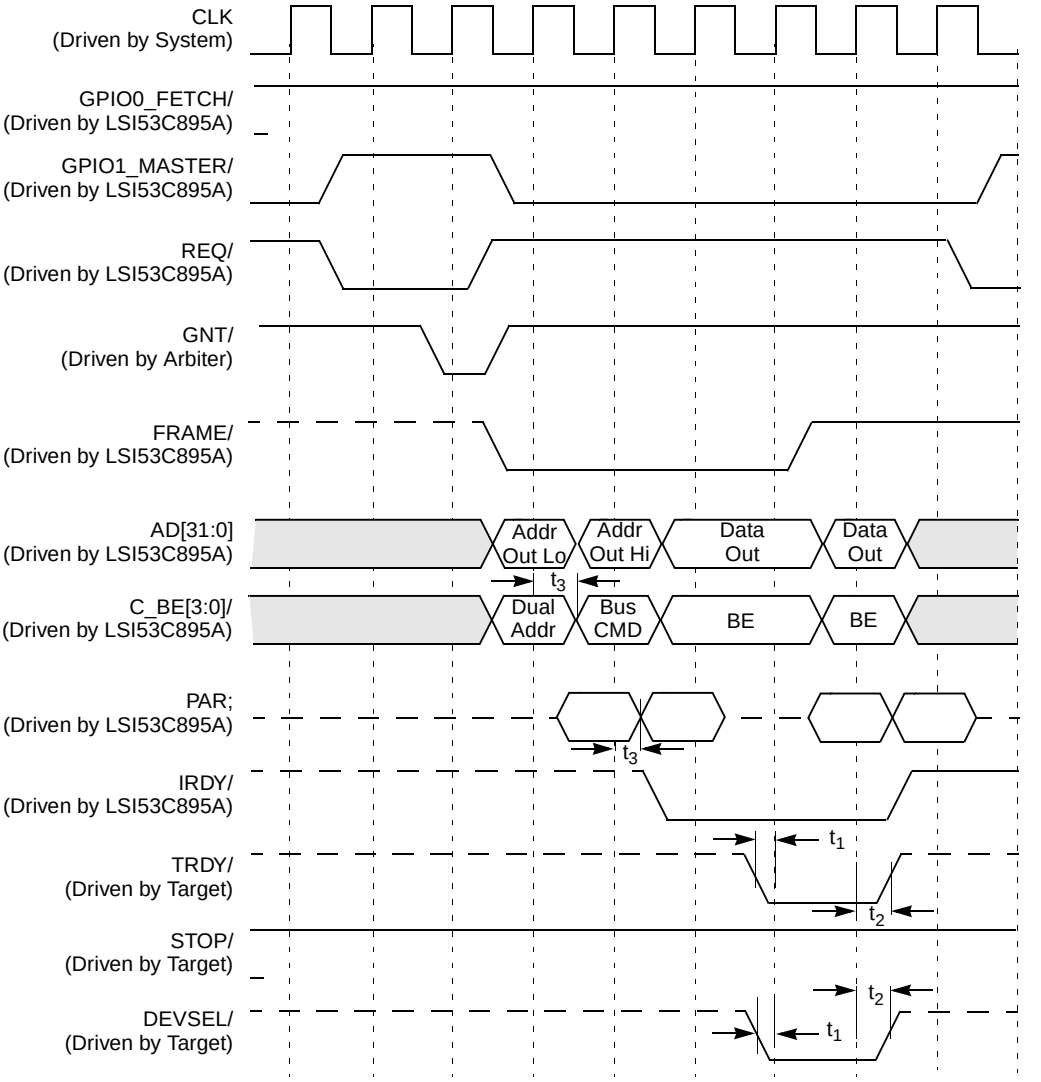


Table 6.31 Burst Write, 64-Bit Address and 32-Bit Data

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_{10}	CLK HIGH to GPIO1_MASTER/ HIGH	–	20	ns

Figure 6.24 Burst Write, 64-Bit Address and 32-Bit Data



This page intentionally left blank.

6.4.3 External Memory Timing

Tables 6.32 through 6.39 and Figures 6.25 through 6.34 describe External Memory timing.

Table 6.32 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	–	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

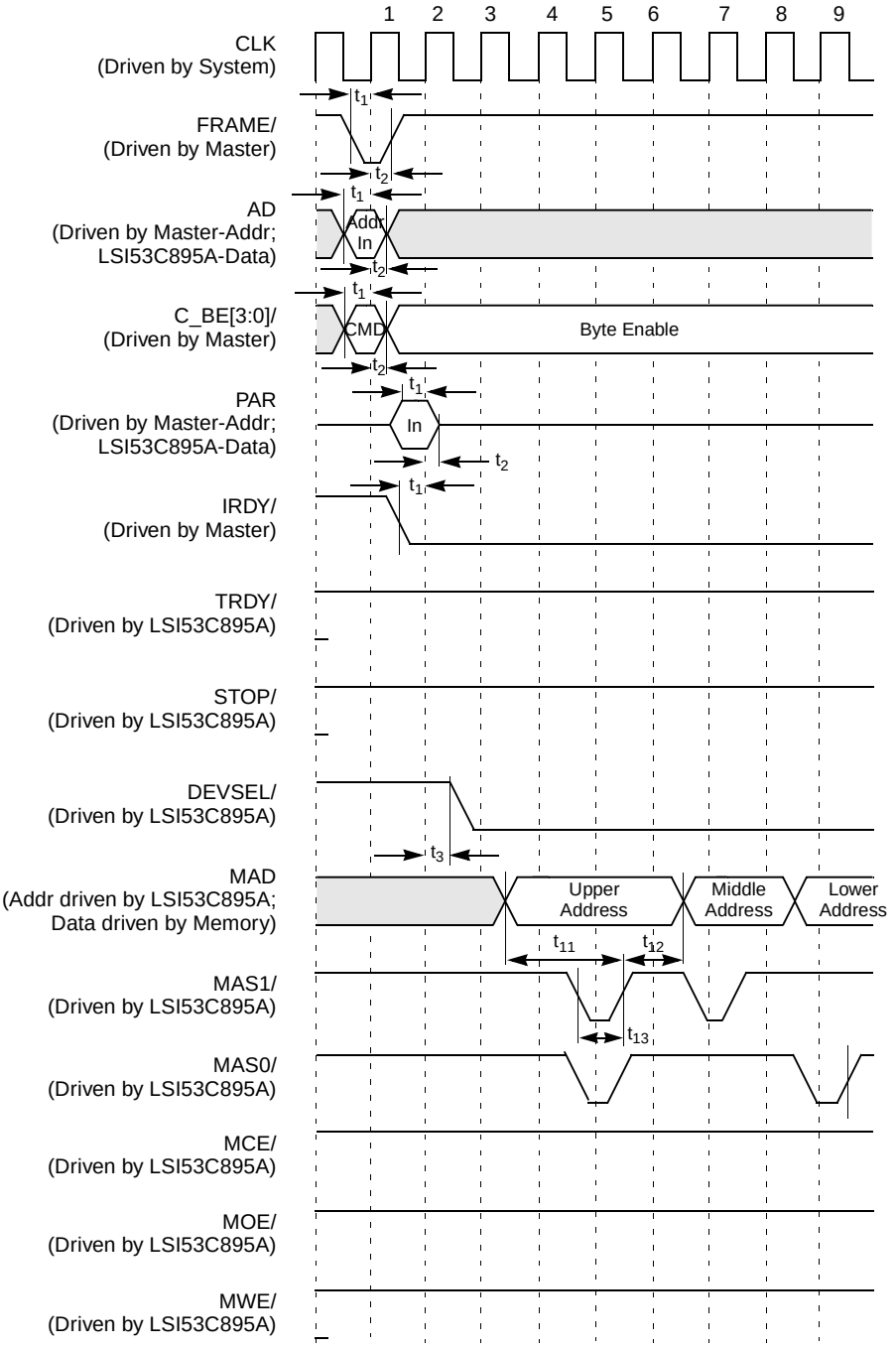
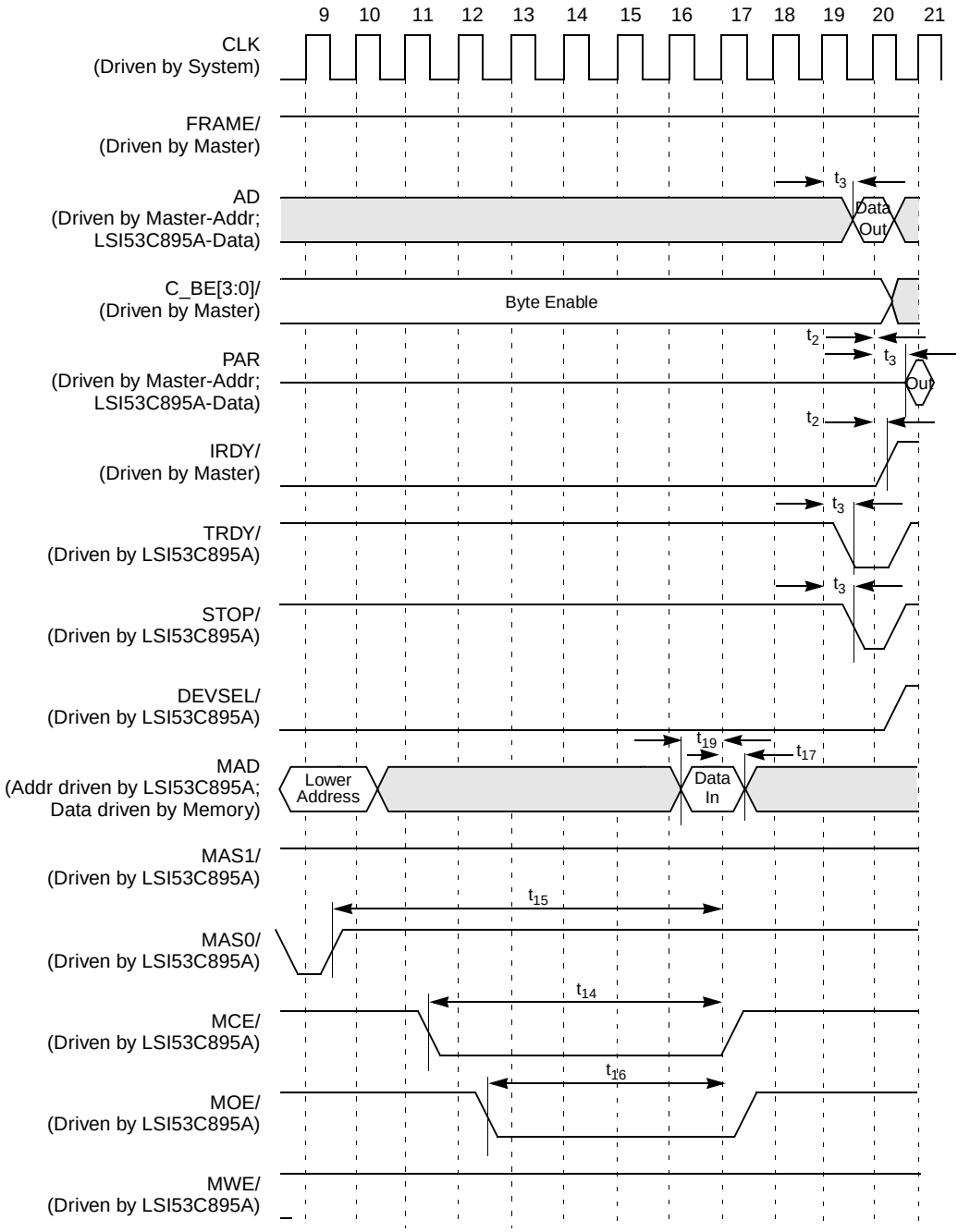


Figure 6.25 External Memory Read (Cont.)



This page intentionally left blank.

Table 6.33 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	–	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

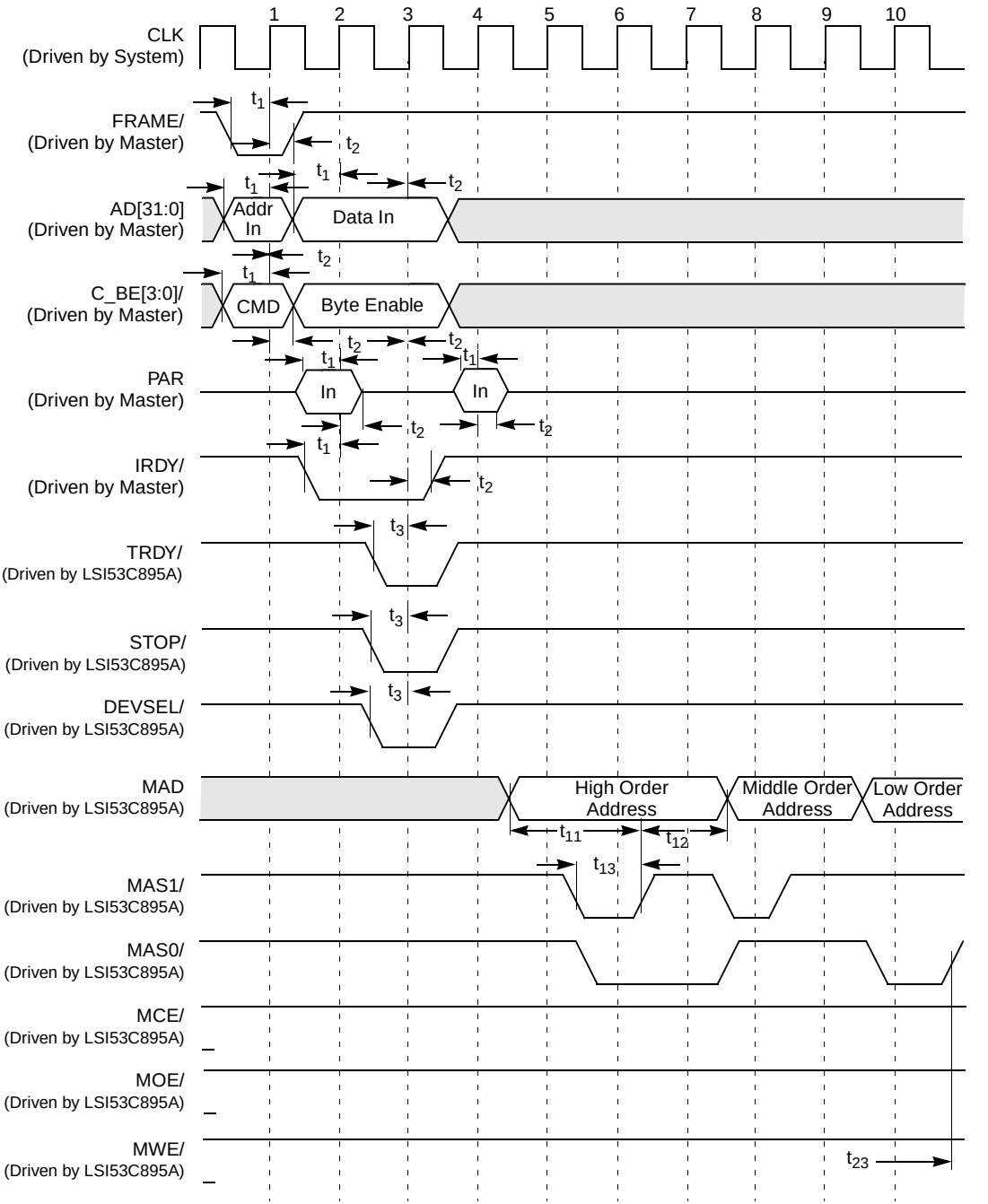


Figure 6.26 External Memory Write (Cont.)

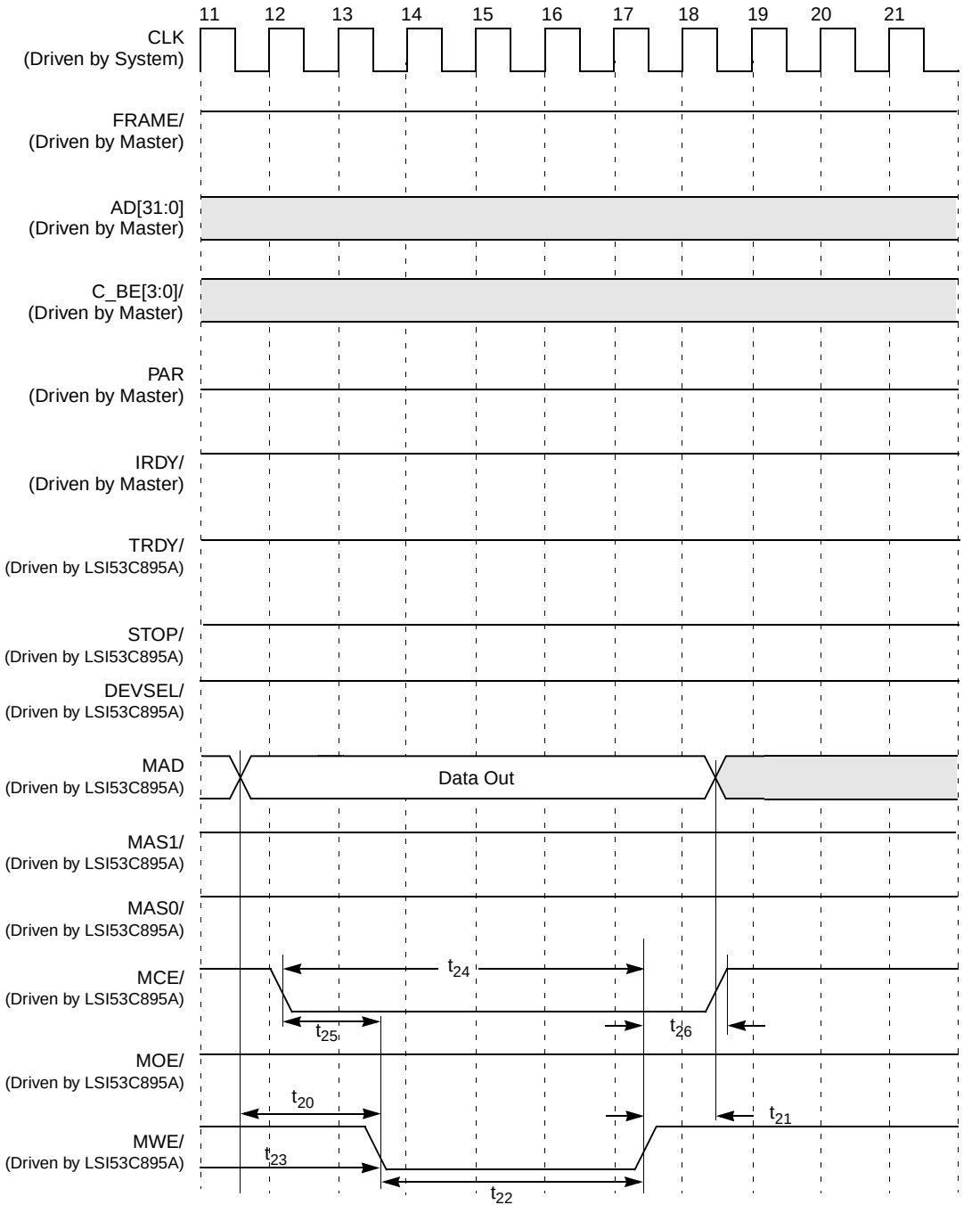
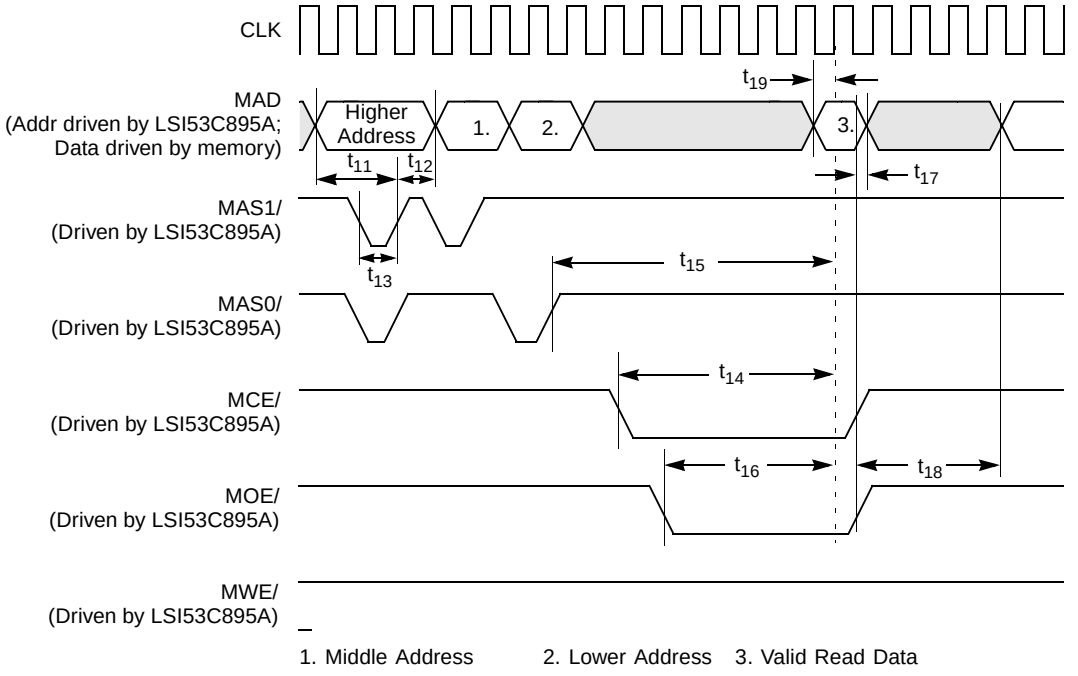


Table 6.34 Normal/Fast Memory (≥ 128 Kbytes) Single Byte Access Read Cycle

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



1. Middle Address 2. Lower Address 3. Valid Read Data

Table 6.35 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

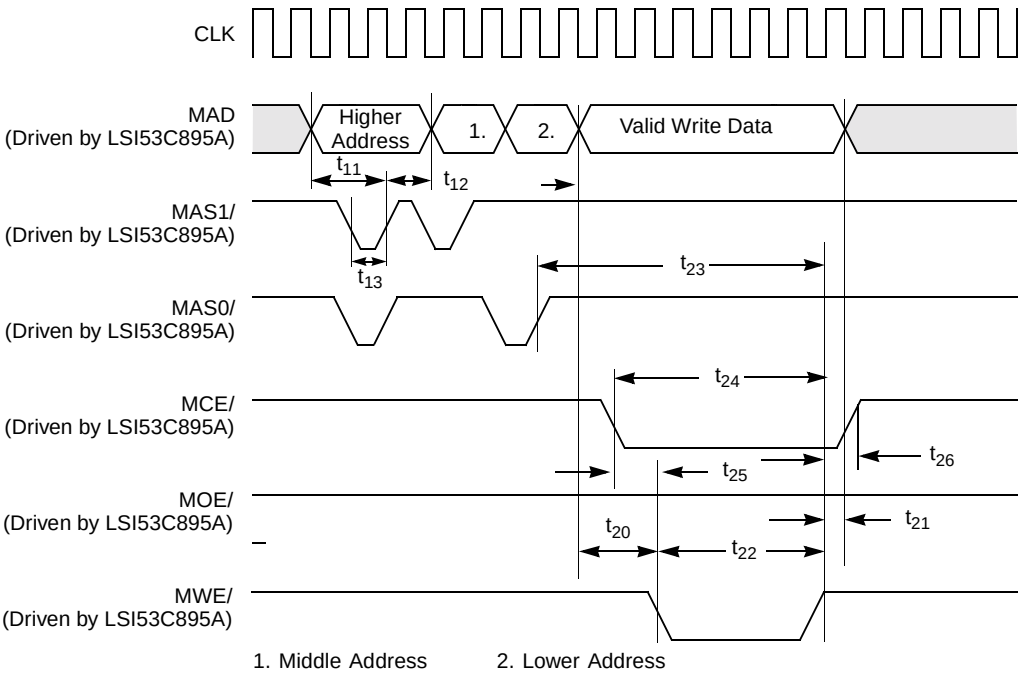


Figure 6.29 Normal/Fast Memory (≥ 128 Kbytes) Multiple Byte Access Read Cycle

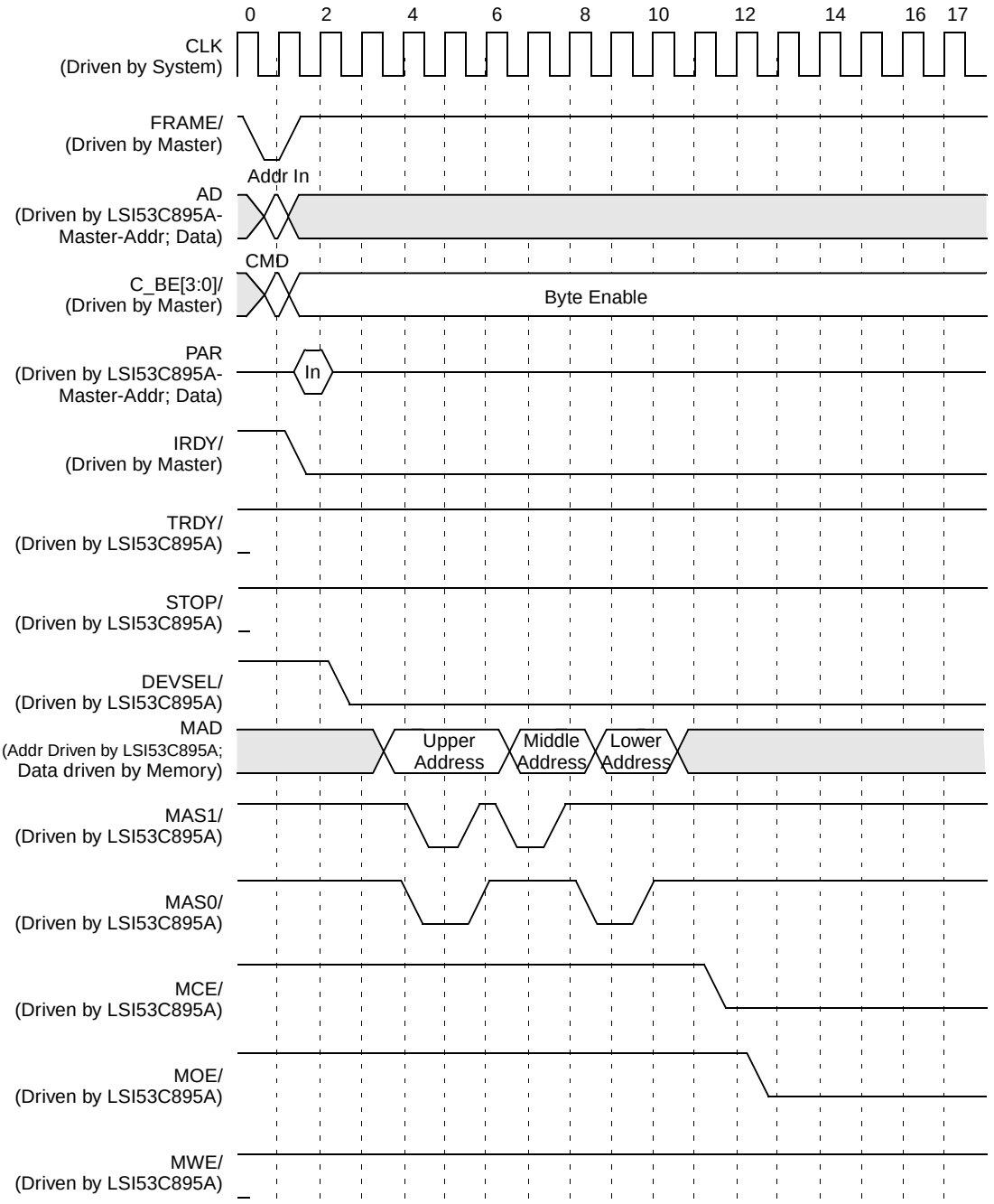


Figure 6.29 Normal/Fast Memory (≥ 128 Kbytes) Multiple Byte Access Read Cycle (Cont.)

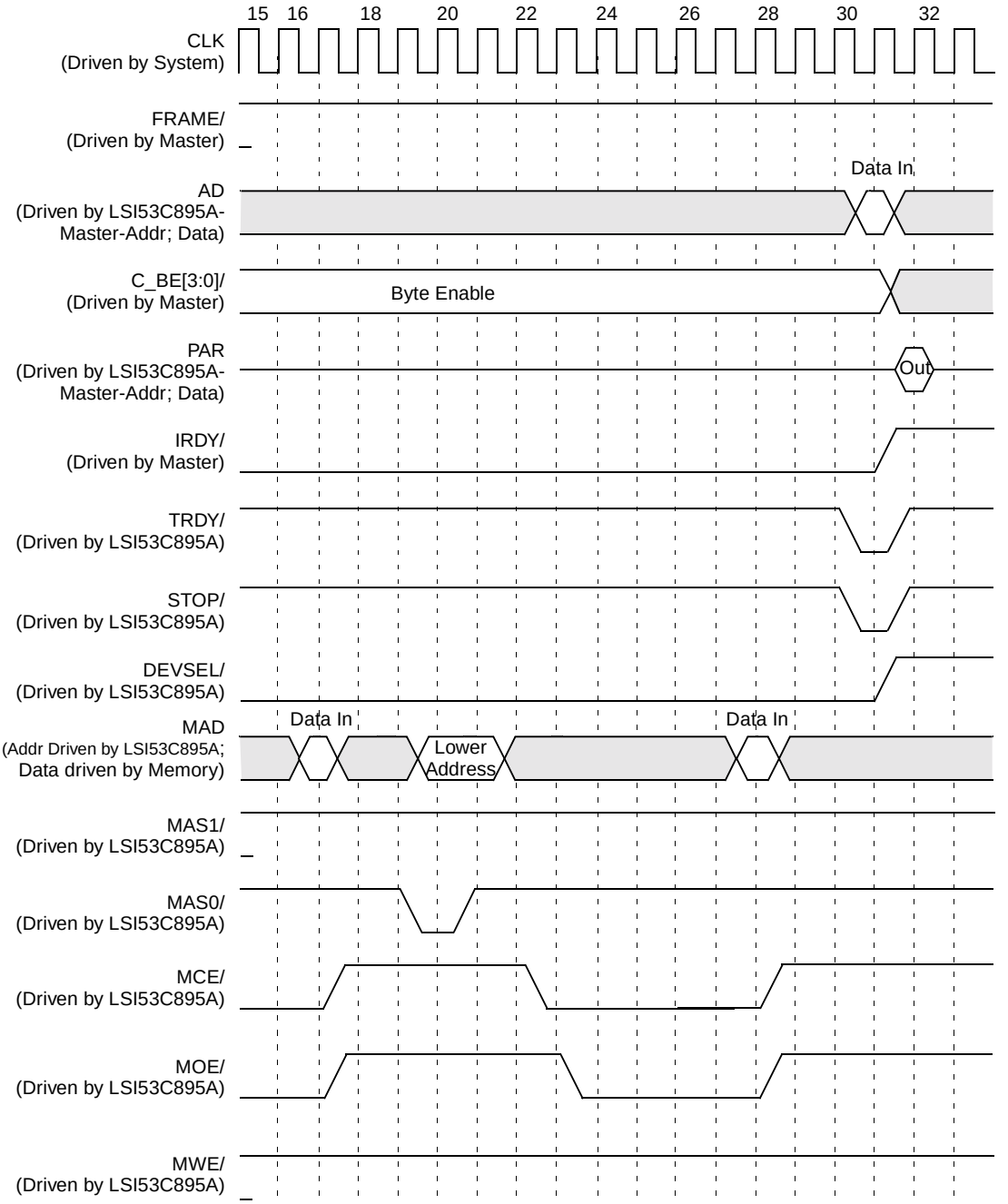


Figure 6.30 Normal/Fast Memory (≥ 128 Kbytes) Multiple Byte Access Write Cycle

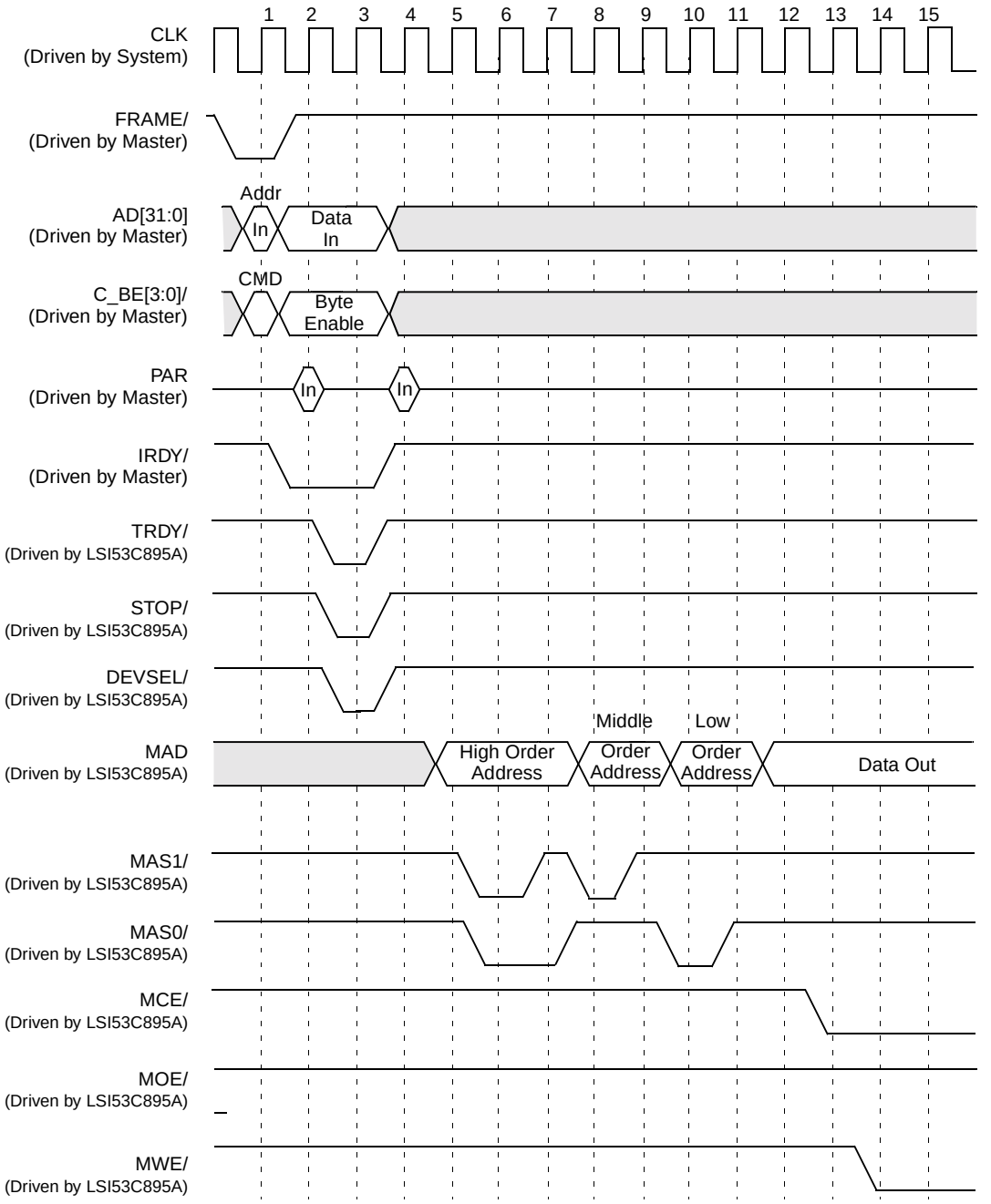


Figure 6.30 Normal/Fast Memory (≥ 128 Kbytes) Multiple Byte Access Write Cycle (Cont.)

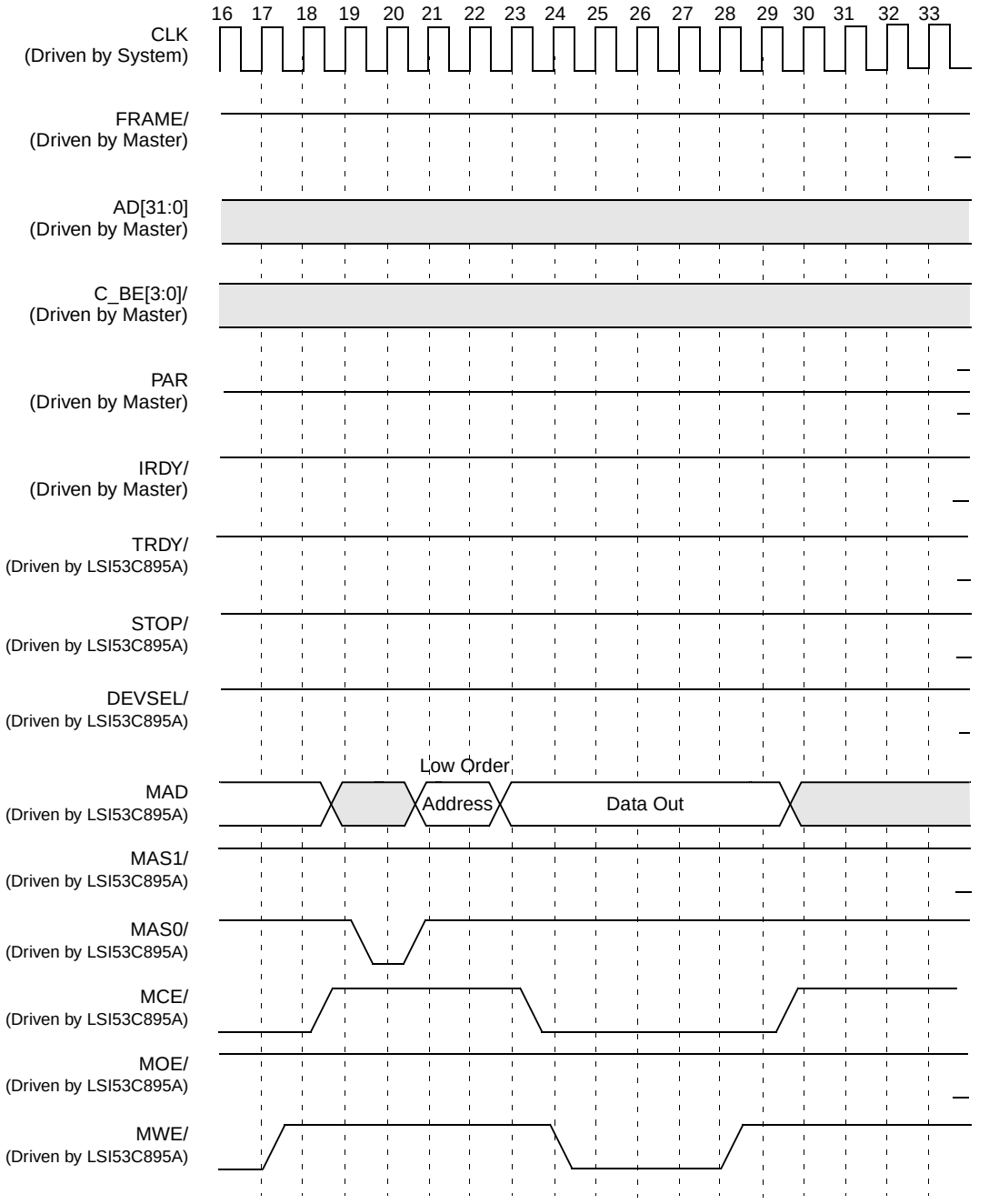


Table 6.36 Slow Memory (≤ 128 Kbytes) Read Cycle

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

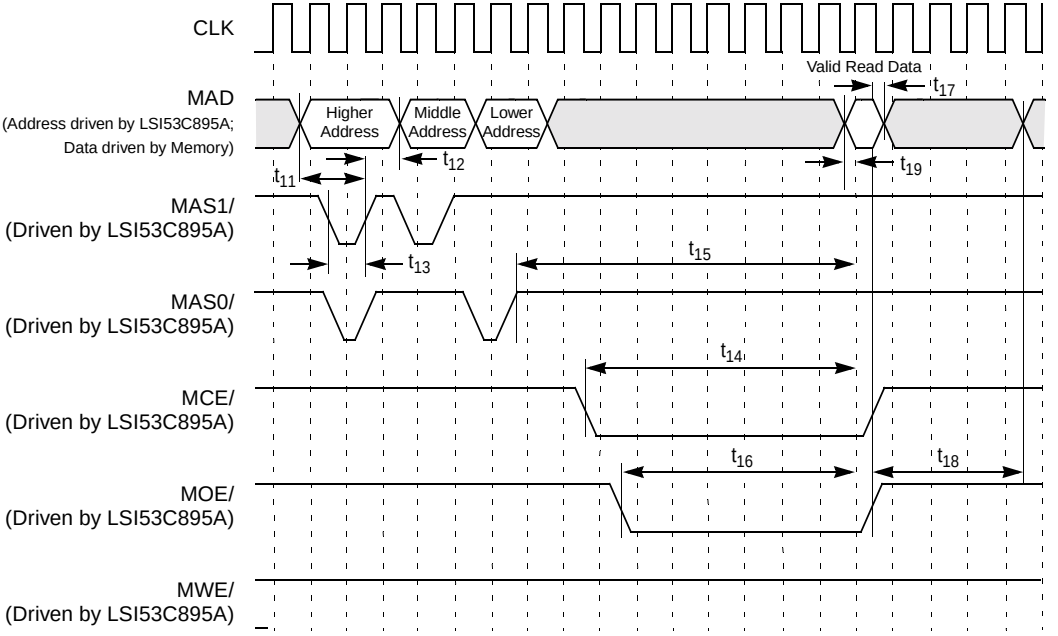


Table 6.37 Slow Memory (≤ 128 Kbytes) Write Cycle

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

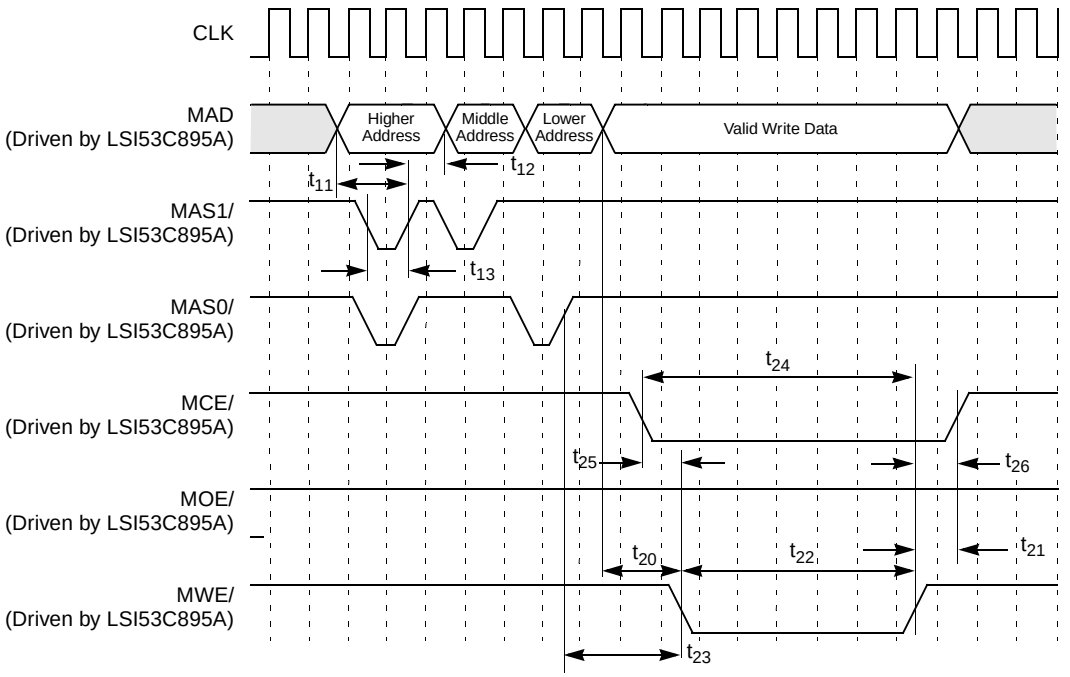


Table 6.38 ≤ 64 Kbytes ROM Read Cycle

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

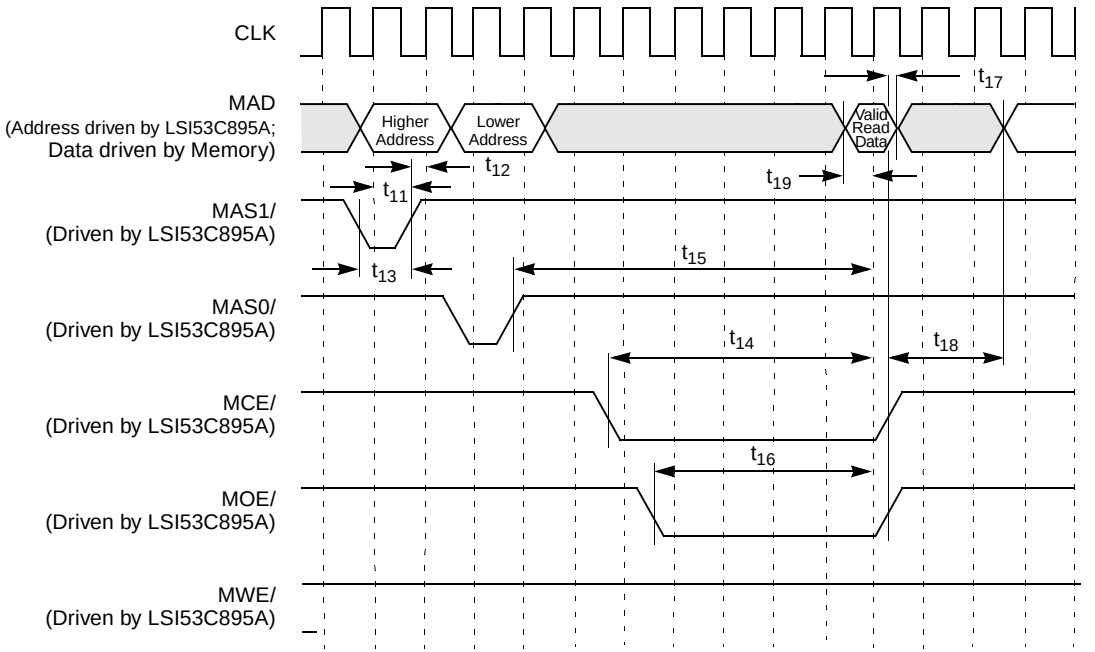
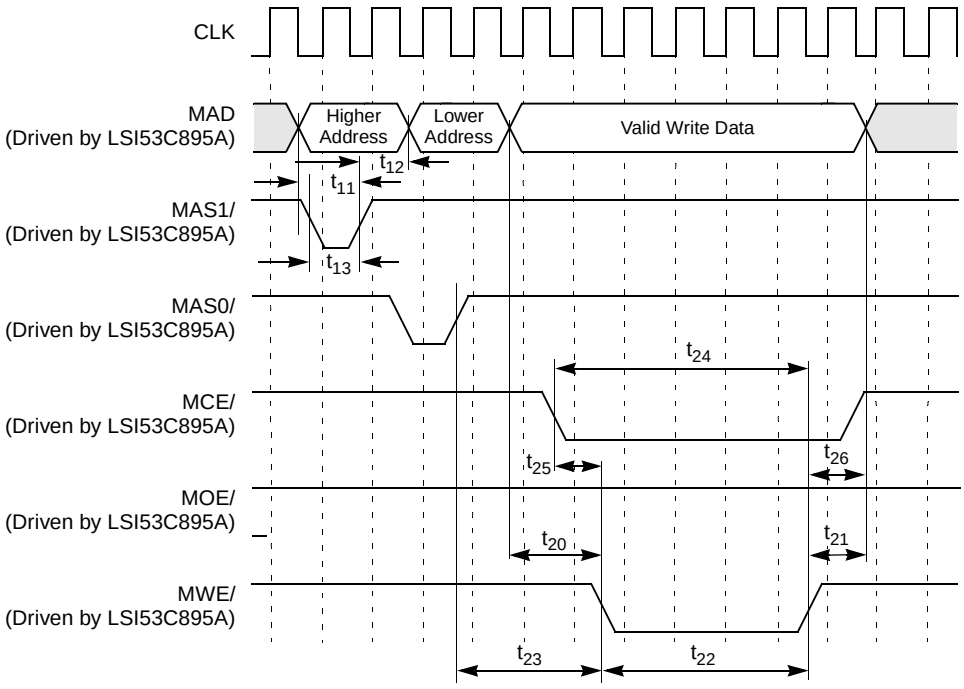


Table 6.39 ≤ 64 Kbyte ROM Write Cycle

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 Kbyte ROM Write Cycle



6.5 SCSI Timing Diagrams

Tables 6.40 through 6.50 and Figures 6.35 through 6.39 and describe the LSI53C895A SCSI timing.

Table 6.40 Initiator Asynchronous Send

Symbol	Parameter	Min	Max	Unit
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

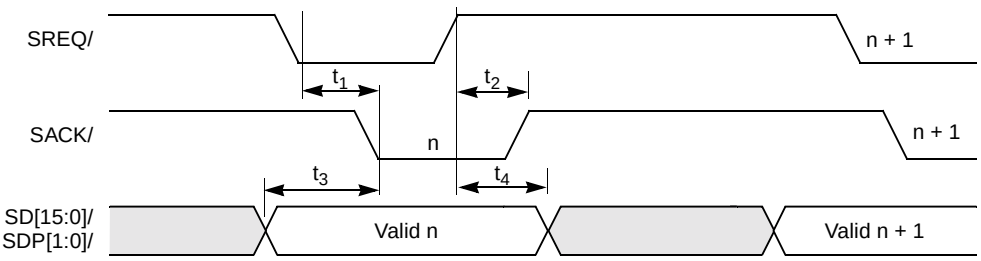


Table 6.41 Initiator Asynchronous Receive

Symbol	Parameter	Min	Max	Unit
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

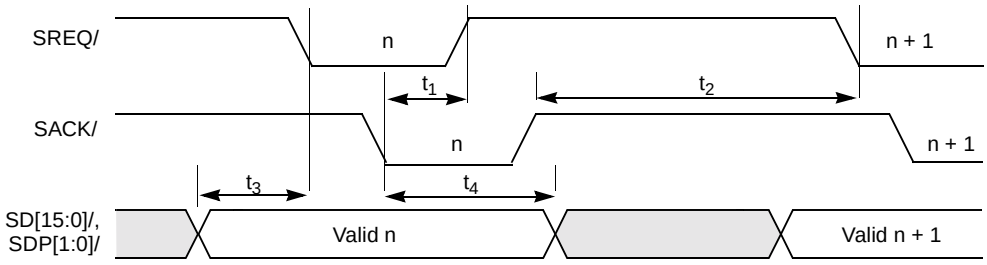


Table 6.42 Target Asynchronous Send

Symbol	Parameter	Min	Max	Unit
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

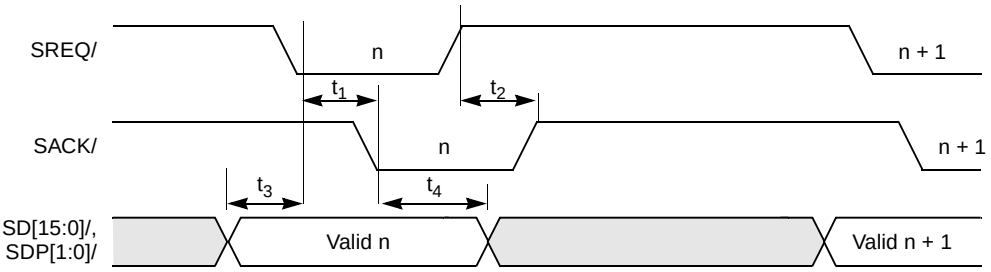


Table 6.43 Target Asynchronous Receive

Symbol	Parameter	Min	Max	Unit
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

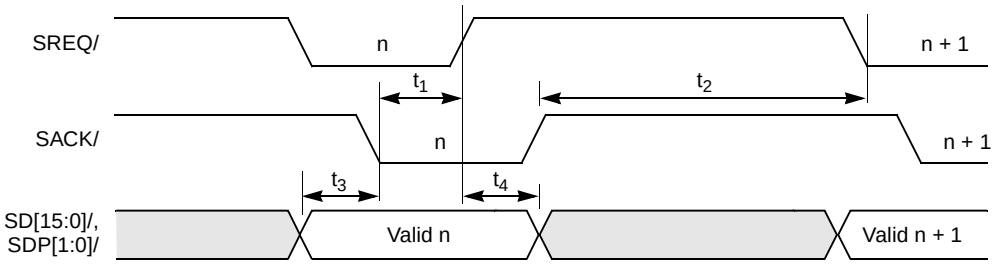


Table 6.44 SCSI-1 Transfers (SE 5.0 Mbytes)

Symbol	Parameter	Min	Max	Unit
t_1	Send SREQ/ or SACK/ assertion pulse width	80	–	ns
t_2	Send SREQ/ or SACK/ deassertion pulse width	80	–	ns
t_1	Receive SREQ/ or SACK/ assertion pulse width	70	–	ns
t_2	Receive SREQ/ or SACK/ deassertion pulse width	70	–	ns
t_3	Send data setup to SREQ/ or SACK/ asserted	24	–	ns
t_4	Send data hold from SREQ/ or SACK/ asserted	54	–	ns
t_5	Receive data setup to SREQ/ or SACK/ asserted	14	–	ns
t_6	Receive data hold from SREQ/ or SACK/ asserted	24	–	ns

Table 6.45 SCSI-1 Transfers (Differential 4.17 Mbytes)

Symbol	Parameter	Min	Max	Unit
t_1	Send SREQ/ or SACK/ assertion pulse width	96	–	ns
t_2	Send SREQ/ or SACK/ deassertion pulse width	96	–	ns
t_1	Receive SREQ/ or SACK/ assertion pulse width	84	–	ns
t_2	Receive SREQ/ or SACK/deassertion pulse width	84	–	ns
t_3	Send data setup to SREQ/ or SACK/ asserted	65	–	ns
t_4	Send data hold from SREQ/ or SACK/ asserted	110	–	ns
t_5	Receive data setup to SREQ/ or SACK/ asserted	0	–	ns
t_6	Receive data hold from SREQ/ or SACK/ asserted	45	–	ns

Table 6.46 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_1	Send SREQ/ or SACK/ assertion pulse width	30	–	ns
t_2	Send SREQ/ or SACK/ deassertion pulse width	30	–	ns
t_1	Receive SREQ/ or SACK/ assertion pulse width	22	–	ns
t_2	Receive SREQ/ or SACK/ deassertion pulse width	22	–	ns
t_3	Send data setup to SREQ/ or SACK/ asserted	24	–	ns
t_4	Send data hold from SREQ/ or SACK/ asserted	34	–	ns
t_5	Receive data setup to SREQ/ or SACK/ asserted	14	–	ns
t_6	Receive data hold from SREQ/ or SACK/ asserted	24	–	ns

Table 6.47 SCSI-2 Fast Transfers 10.0 Mbytes (8-Bit Transfers) or 20.0 Mbytes (16-Bit Transfers) 50 MHz Clock[1, 2]

Symbol	Parameter	Min	Max	Unit
t_1	Send SREQ/ or SACK/ assertion pulse width	30	–	ns
t_2	Send SREQ/ or SACK/ deassertion pulse width	30	–	ns
t_1	Receive SREQ/ or SACK/ assertion pulse width	22	–	ns
t_2	Receive SREQ/ or SACK/ deassertion pulse width	22	–	ns
t_3	Send data setup to SREQ/ or SACK/ asserted	24	–	ns
t_4	Send data hold from SREQ/ or SACK/ asserted	40[3]	–	ns
t_5	Receive data setup to SREQ/ or SACK/ asserted	14	–	ns
t_6	Receive data hold from SREQ/ or SACK/ asserted	24	–	ns

1. Transfer period bits (bits [7:5] in the SCSI Transfer (SXFER) register) are set to zero and the Extra Clock Cycle of Data Setup bit (bit 7 in SCSI Control One (SCNTL1)) is set.
2. For fast SCSI, set the TolerANT Enable bit (bit 7 in SCSI Test Three (STEST3)).
3. Analysis of system configuration is recommended due to reduced driver skew margin in differential systems.

Table 6.48 Ultra SCSI SE Transfers 20.0 Mbytes (8-Bit Transfers) or 40.0 Mbytes (16-Bit Transfers) Quadrupled 40 MHz Clock[1, 2]

Symbol	Parameter	Min	Max	Unit
t_1	Send SREQ/ or SACK/ assertion pulse width	15	–	ns
t_2	Send SREQ/ or SACK/ deassertion pulse width	15	–	ns
t_1	Receive SREQ/ or SACK/ assertion pulse width	11	–	ns
t_2	Receive SREQ/ or SACK/ deassertion pulse width	11	–	ns
t_3	Send data setup to SREQ/ or SACK/ asserted	12	–	ns
t_4	Send data hold from SREQ/ or SACK/ asserted	17	–	ns
t_5	Receive data setup to SREQ/ or SACK/ asserted	6	–	ns
t_6	Receive data hold from SREQ/ or SACK/ asserted	11	–	ns

1. Transfer period bits (bits [7:5] in the SCSI Transfer (SXFER) register) are set to zero and the Extra Clock Cycle of Data Setup bit (bit 7 in SCSI Control One (SCNTL1)) is set.
2. During Ultra2 SCSI transfers, the value of the Extend REQ/ACK Filtering bit (SCSI Test Two (STEST2), bit 1) has no effect.

Table 6.49 Ultra SCSI High Voltage Differential Transfers 20.0 Mbytes (8-Bit Transfers) or 40.0 Mbytes (16-Bit Transfers) 80 MHz Clock[1, 2]

Symbol	Parameter	Min	Max	Unit
t_1	Send SREQ/ or SACK/ assertion pulse width	15	–	ns
t_2	Send SREQ/ or SACK/ deassertion pulse width	15	–	ns
t_1	Receive SREQ/ or SACK/ assertion pulse width	11	–	ns
t_2	Receive SREQ/ or SACK/ deassertion pulse width	11	–	ns
t_3	Send data setup to SREQ/ or SACK/ asserted	16	–	ns
t_4	Send data hold from SREQ/ or SACK/ asserted	21	–	ns
t_5	Receive data setup to SREQ/ or SACK/ asserted	6	–	ns
t_6	Receive data hold from SREQ/ or SACK/ asserted	11	–	ns

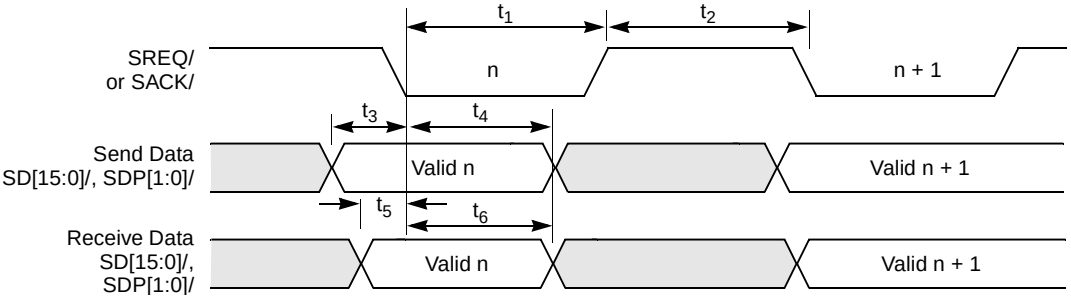
1. Transfer period bits (bits [7:5] in the SCSI Transfer (SXFER) register) are set to zero and the Extra Clock Cycle of Data Setup bit (bit 7 in SCSI Control One (SCNTL1)) is set.
2. During Ultra SCSI transfers, the value of the Extend REQ/ACK Filtering bit (SCSI Test Two (STEST2), bit 1) has no effect.

Table 6.50 Ultra2 SCSI Transfers 40.0 Mbytes (8-Bit Transfers) or 80.0 Mbytes (16-Bit Transfers) Quadrupled 40 MHz Clock[1, 2]

Symbol	Parameter	Min	Max	Unit
t_1	Send SREQ/ or SACK/ assertion pulse width	8	–	ns
t_2	Send SREQ/ or SACK/ deassertion pulse width	8	–	ns
t_1	Receive SREQ/ or SACK/ assertion pulse width	6.5	–	ns
t_2	Receive SREQ/ or SACK/ deassertion pulse width	6.5	–	ns
t_3	Send data setup to SREQ/ or SACK/ asserted	9.5	–	ns
t_4	Send data hold from SREQ/ or SACK/ asserted	9.5	–	ns
t_5	Receive data setup to SREQ/ or SACK/ asserted	4.5	–	ns
t_6	Receive data hold from SREQ/ or SACK/ asserted	4.5	–	ns

1. Transfer period bits (bits [7:5] in the SCSI Transfer (SXFER) register) are set to zero and the Extra Clock Cycle of Data Setup bit (bit 7 in SCSI Control One (SCNTL1)) is set.
2. During Ultra2 SCSI transfers, the value of the Extend REQ/ACK Filtering bit (SCSI Test Two (STEST2), bit 1) has no effect.

Figure 6.39 Initiator and Target Synchronous Transfer



6.6 Package Diagrams

This section provides pinout information for both chips. Figure 6.40 is pinout information for the LSI53C895A 272-pin BGA and Figure 6.41 is pinout information for the LSI53C895A 208-pin PQFP. Table 6.51 provides the 272 BGA pin list by location and Table 6.52 provides the same pin list alphabetically. Table 6.53 provides pinout information for the 208 PQFP. Section 6.6.1, “LSI53C895A vs. LSI53C895 Pin/Ball Differences,” provides the pin and signal differences of the two parts. The end of this section provides the mechanical drawings for both packages.

Figure 6.40 LSI53C895A 272-Pin BGA Top View

A1 VSS	A2 N/C	A3 SD2+	A4 SD3+	A5 SD4+	A6 N/C	A7 SD5-	A8 SD6-	A9 SD7-	A10 RBIAS	A11 VDD_ BIAS	A12 SATN+	A13 SBSY+	A14 SACK+	A15 SRST+	A16 SMSG+	A17 SSEL+	A18 SSEL-	A19 N/C	A20 N/C
B1 N/C	B2 SD1+	B3 SD1-	B4 SD2-	B5 SD3-	B6 SD4-	B7 SD5+	B8 SD6+	B9 SD7+	B10 SDP0-	B11 N/C	B12 SATN-	B13 SBSY-	B14 SACK-	B15 SRST-	B16 SMSG-	B17 N/C	B18 N/C	B19 SREQ2+	B20 SREQ2-
C1 SD0-	C2 N/C	C3 N/C	C4 N/C	C5 N/C	C6 N/C	C7 N/C	C8 N/C	C9 N/C	C10 SDP0+	C11 N/C	C12 N/C	C13 N/C	C14 N/C	C15 N/C	C16 N/C	C17 SCD-	C18 N/C	C19 SREQ+	C20 SIO-
D1 SD0+	D2 SACK2-	D3 SACK2+	D4 VSS	D5 N/C	D6 VDD	D7 N/C	D8 VSS	D9 N/C	D10 N/C	D11 VDD	D12 N/C	D13 VSS	D14 N/C	D15 VDD	D16 SCD+	D17 VSS	D18 SREQ-	D19 SD8+	D20 SD8-
E1 SDP1-	E2 N/C	E3 N/C	E4 N/C													E17 SIO+	E18 N/C	E19 SD9+	E20 SD9-
F1 SD15+	F2 SD15-	F3 SDP1+	F4 VDD													F17 VDD	F18 N/C	F19 SD10+	F20 SD10-
G1 SD14+	G2 SD14-	G3 N/C	G4 N/C													G17 N/C	G18 N/C	G19 SD11+	G20 SD11-
H1 SD13+	H2 SD13-	H3 N/C	H4 VSS													H17 VSS	H18 N/C	H19 VDDA	H20 DIFFSENS
J1 N/C	J2 SD12+	J3 SD12-	J4 N/C					J9 VSS	J10 VSS	J11 VSS	J12 VSS					J17 N/C	J18 VSSA	J19 TEST_ HSC/	J20 SCLK
K1 TCK	K2 TEST_ RST/	K3 N/C	K4 VDD					K9 VSS	K10 VSS	K11 VSS	K12 VSS					K17 N/C	K18 N/C	K19 MAC/_ TESTOUT	K20 MAD0
L1 TMS	L2 TDO	L3 TDI	L4 N/C					L9 VSS	L10 VSS	L11 VSS	L12 VSS					L17 VDD	L18 N/C	L19 MAD2	L20 MAD1
M1 MAS1/	M2 MAS0/	M3 VSS_ CORE	M4 N/C					M9 VSS	M10 VSS	M11 VSS	M12 VSS					M17 N/C	M18 MAD5	M19 MAD4	M20 MAD3
N1 VSS_ CORE2	N2 MWE/	N3 MOE/	N4 VSS													N17 VSS	N18 VSS_ CORE	N19 MAD7	N20 MAD6
P1 VDD_ CORE	P2 VDD_ CORE	P3 N/C	P4 N/C													P17 VDD_ CORE	P18 GPIO3	P19 GPIO4	P20 VSS_ CORE
R1 MCE/	R2 RST/	R3 N/C	R4 VDD													R17 VDD	R18 GPIO_ MASTER/	R19 VDD_ CORE	R20 GPIO2
T1 CLK	T2 GNT/	T3 N/C	T4 N/C													T17 N/C	T18 N/C	T19 GPIO0_ FETCH/	T20 N/C
U1 REQ/	U2 AD31	U3 N/C	U4 VSS	U5 N/C	U6 VDD	U7 N/C	U8 VSS	U9 C_BE2/	U10 VDD	U11 SERR/	U12 N/C	U13 VSS	U14 N/C	U15 VDD	U16 N/C	U17 VSS	U18 AD1	U19 N/C	U20 IRQ/
V1 AD30	V2 AD29	V3 AD27	V4 N/C	V5 AD25	V6 IDSEL	V7 AD21	V8 AD18	V9 FRAME/	V10 N/C	V11 PERR/	V12 AD15	V13 AD12	V14 N/C	V15 N/C	V16 N/C	V17 AD4	V18 AD2	V19 GPIO7	V20 AD0
W1 AD28	W2 GPIO8	W3 N/C	W4 N/C	W5 AD24	W6 AD23	W7 AD20	W8 AD17	W9 IRDY/	W10 TRDY/	W11 STOP/	W12 C_BE1/	W13 AD13	W14 AD10	W15 AD8	W16 N/C	W17 AD6	W18 N/C	W19 GPIO5	W20 GPIO6
Y1 TRST	Y2 ALT_IRQ/	Y3 AD26	Y4 N/C	Y5 C_BE3/	Y6 AD22	Y7 AD19	Y8 AD16	Y9 N/C	Y10 DEVSEL/	Y11 N/C	Y12 PAR	Y13 AD14	Y14 AD11	Y15 AD9	Y16 C_BE0/	Y17 AD7	Y18 AD5	Y19 AD3	Y20 N/C

Table 6.51 272 BGA Pin List by Location

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
VSS	A1	NC	C16	VDDA	H19	VSS	N17	IDSEL	V6
NC	A2	SCD–	C17	DIFFSENS	H20	VSSCORE	N18	AD21	V7
SD2+	A3	NC	C18	NC	J1	MAD7	N19	AD18	V8
SD3+	A4	SREQ+	C19	SD12+	J2	MAD6	N20	FRAME/	V9
SD4+	A5	SIO–	C20	SD12–	J3	VDDCORE	P1	NC	V10
NC	A6	SD0+	D1	NC	J4	VDDCORE	P2	PERR/	V11
SD5–	A7	SACK2–	D2	VSS	J9	NC	P3	AD15	V12
SD6–	A8	SACK2+	D3	VSS	J10	NC	P4	AD12	V13
SD7–	A9	VSS	D4	VSS	J11	VDDCORE	P17	NC	V14
RBIAS	A10	NC	D5	VSS	J12	GPIO3	P18	NC	V15
VDD_RBIAS	A11	VDD	D6	NC	J17	GPIO4	P19	NC	V16
SATN+	A12	NC	D7	VSSA	J18	VSSCORE	P20	AD4	V17
SBSY+	A13	VSS	D8	TEST_HSC/	J19	MCE/	R1	AD2	V18
SACK+	A14	NC	D9	SCLK	J20	RST/	R2	GPIO7	V19
SRST+	A15	NC	D10	TCK	K1	NC	R3	AD0	V20
SMSG+	A16	VDD	D11	TEST_RST/	K2	VDD	R4	AD28	W1
SSEL+	A17	NC	D12	NC	K3	VDD	R17	GPIO8	W2
SSEL–	A18	VSS	D13	VDD	K4	GPIO1		NC	W3
NC	A19	NC	D14	VSS	K9	MASTER/	R18	NC	W4
NC	A20	VDD	D15	VSS	K10	VDDCORE	R19	AD24	W5
NC	B1	SCD+	D16	VSS	K11	GPIO2	R20	AD23	W6
SD1+	B2	VSS	D17	VSS	K12	CLK	T1	AD20	W7
SD1–	B3	SREQ–	D18	NC	K17	GNT/	T2	AD17	W8
SD2–	B4	SD8+	D19	NC	K18	NC	T3	IRDY/	W9
SD3–	B5	SD8–	D20	MAC/		NC	T4	TRDY/	W10
SD4–	B6	SDP1–	E1	TESTOUT	K19	NC	T17	STOP/	W11
SD5+	B7	NC	E2	MAD0	K20	NC	T18	C_BE1/	W12
SD6+	B8	NC	E3	TMS	L1	GPIO0		AD13	W13
SD7+	B9	NC	E4	TDO	L2	FETCH/	T19	AD10	W14
SDP0–	B10	SIO+	E17	TDI	L3	NC	T20	AD8	W15
NC	B11	NC	E18	NC	L4	REQ/	U1	NC	W16
SATN–	B12	SD9+	E19	VSS	L9	AD31	U2	AD6	W17
SBSY–	B13	SD9–	E20	VSS	L10	NC	U3	NC	W18
SACK–	B14	SD15+	F1	VSS	L11	VSS	U4	GPIO5	W19
SRST–	B15	SD15–	F2	VSS	L12	NC	U5	GPIO6	W20
SMSG–	B16	SDP1+	F3	VDD	L17	VDD	U6	TRST	Y1
NC	B17	VDD	F4	NC	L18	NC	U7	ALT_IRQ/	Y2
NC	B18	VDD	F17	MAD2	L19	VSS	U8	AD26	Y3
SREQ2+	B19	NC	F18	MAD1	L20	C_BE2/	U9	NC	Y4
SREQ2–	B20	SD10+	F19	MAS1/	M1	VDD	U10	C_BE3/	Y5
SD0–	C1	SD10–	F20	MAS0/	M2	SERR/	U11	AD22	Y6
NC	C2	SD14+	G1	VSSCORE	M3	NC	U12	AD19	Y7
NC	C3	SD14–	G2	NC	M4	VSS	U13	AD16	Y8
NC	C4	NC	G3	VSS	M9	NC	U14	NC	Y9
NC	C5	NC	G4	VSS	M10	VDD	U15	DEVSEL/	Y10
NC	C6	NC	G17	VSS	M11	NC	U16	NC	Y11
NC	C7	NC	G18	VSS	M12	VSS	U17	PAR	Y12
NC	C8	SD11+	G19	NC	M17	AD1	U18	AD14	Y13
NC	C9	SD11–	G20	MAD5	M18	NC	U19	AD11	Y14
SDP0+	C10	SD13+	H1	MAD4	M19	IRQ/	U20	AD9	Y15
NC	C11	SD13–	H2	MAD3	M20	AD30	V1	C_BE0/	Y16
NC	C12	NC	H3	VSSCORE2	N1	AD29	V2	AD7	Y17
NC	C13	VSS	H4	MWE/	N2	AD27	V3	AD5	Y18
NC	C14	VSS	H17	MOE/	N3	NC	V4	AD3	Y19
NC	C15	NC	H18	VSS	N4	AD25	V5	NC	Y20

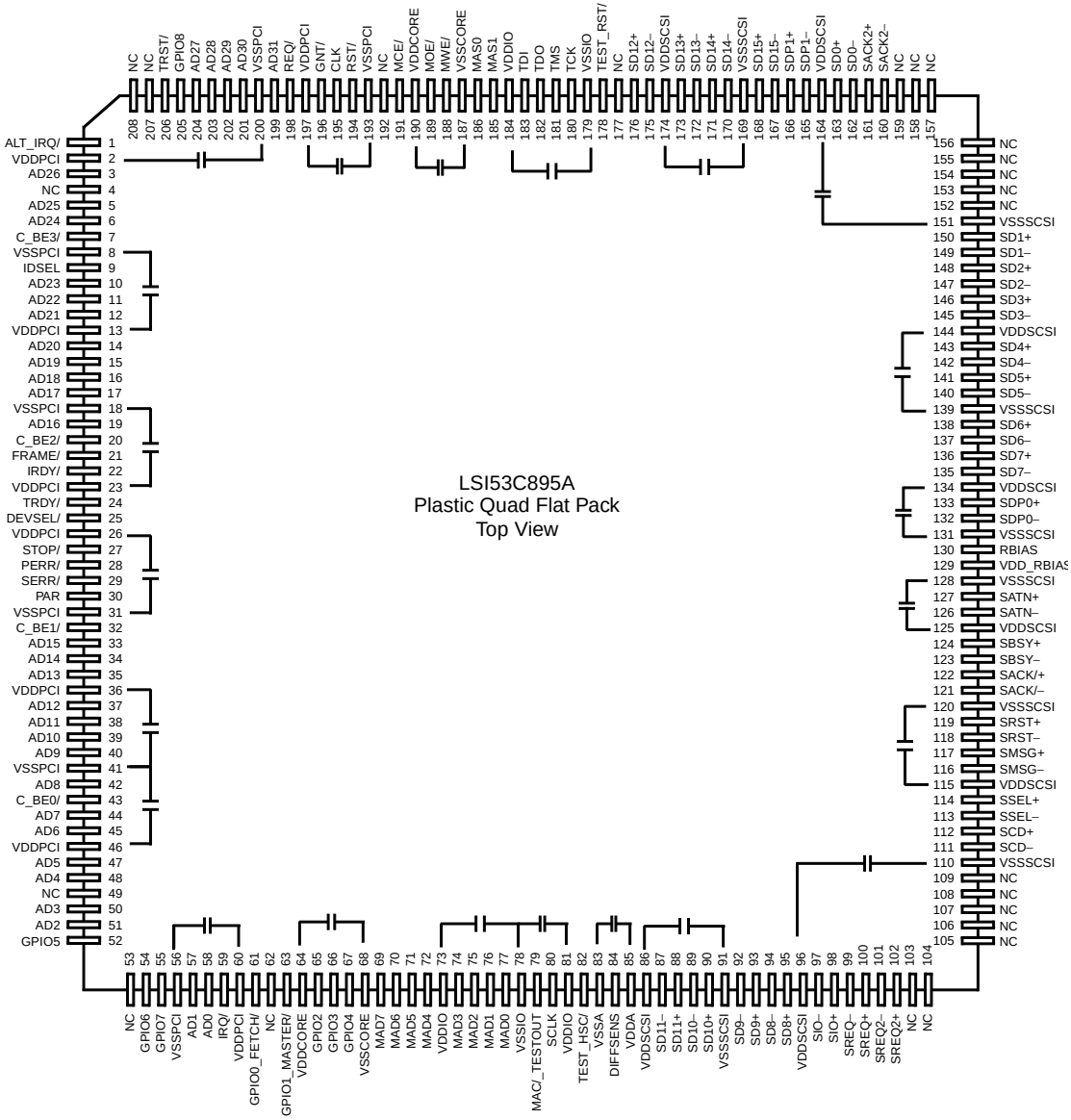
1. *NC* pins are not connected.

Table 6.52 BGA Pin List Alphabetically

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
AD0	V20	MAC/		No Connect	J4	SD1+	B2	VDD	D11
AD1	U18	TESTOUT	K19	No Connect	J17	SD2–	B4	VDD	D15
AD2	V18	MAD0	K20	No Connect	K3	SD2+	A3	VDD	F4
AD3	Y19	MAD1	L20	No Connect	K17	SD3–	B5	VDD	F17
AD4	V17	MAD2	L19	No Connect	K18	SD3+	A4	VDD	K4
AD5	Y18	MAD3	M20	No Connect	L4	SD4–	B6	VDD	L17
AD6	W17	MAD4	M19	No Connect	L18	SD4+	A5	VDD	R4
AD7	Y17	MAD5	M18	No Connect	M4	SD5–	A7	VDD	R17
AD8	W15	MAD6	N20	No Connect	M17	SD5+	B7	VDD	U6
AD9	Y15	MAD7	N19	No Connect	P3	SD6–	A8	VDD	U10
AD10	W14	MAS0/	M2	No Connect	P4	SD6+	B8	VDD	U15
AD11	Y14	MAS1/	M1	No Connect	R3	SD7–	A9	VDD	
AD12	V13	MCE/	R1	No Connect	T3	SD7+	B9	RBIAS	A11
AD13	W13	MOE/	N3	No Connect	T4	SD8–	D20	VDDA	H19
AD14	Y13	MWE/	N2	No Connect	T17	SD8+	D19	VDDCORE	P1
AD15	V12	No Connect	A2	No Connect	T18	SD9–	E20	VDDCORE	P2
AD16	Y8	No Connect	A6	No Connect	T20	SD9+	E19	VDDCORE	P17
AD17	W8	No Connect	A19	No Connect	U3	SD10–	F20	VDDCORE	R19
AD18	V8	No Connect	A20	No Connect	U5	SD10+	F19	VSS	A1
AD19	Y7	No Connect	B1	No Connect	U7	SD11–	G20	VSS	D4
AD20	W7	No Connect	B11	No Connect	U12	SD11+	G19	VSS	D8
AD21	V7	No Connect	B17	No Connect	U14	SD12–	J3	VSS	D13
AD22	Y6	No Connect	B18	No Connect	U16	SD12+	J2	VSS	D17
AD23	W6	No Connect	C2	No Connect	U19	SD13–	H2	VSS	H4
AD24	W5	No Connect	C3	No Connect	V4	SD13+	H1	VSS	H17
AD25	V5	No Connect	C4	No Connect	V10	SD14–	G2	VSS	J9
AD26	Y3	No Connect	C5	No Connect	V14	SD14+	G1	VSS	J10
AD27	V3	No Connect	C6	No Connect	V15	SD15–	F2	VSS	J11
AD28	W1	No Connect	C7	No Connect	V16	SD15+	F1	VSS	J12
AD29	V2	No Connect	C8	No Connect	W3	SDP0–	B10	VSS	K9
AD30	V1	No Connect	C9	No Connect	W4	SDP0+	C10	VSS	K10
AD31	U2	No Connect	C11	No Connect	W16	SDP1–	E1	VSS	K11
ALT_IRQ/	Y2	No Connect	C12	No Connect	W18	SDP1+	F3	VSS	K12
C_BE0/	Y16	No Connect	C13	No Connect	Y4	SERR/	U11	VSS	L9
C_BE1/	W12	No Connect	C14	No Connect	Y9	SIO–	C20	VSS	L10
C_BE2/	U9	No Connect	C15	No Connect	Y11	SIO+	E17	VSS	L11
C_BE3/	Y5	No Connect	C16	No Connect	Y20	SMSG–	B16	VSS	L12
CLK	T1	No Connect	C18	PAR	Y12	SMSG+	A16	VSS	M9
DEVSEL/	Y10	No Connect	D5	PERR/	V11	SREQ–	D18	VSS	M10
DIFFSENS	H20	No Connect	D7	RBIAS	A10	SREQ+	C19	VSS	M11
FRAME/	V9	No Connect	D9	REQ/	U1	SREQ2–	B20	VSS	M12
GNT/	T2	No Connect	D10	RST/	R2	SREQ2+	B19	VSS	N4
GPIO0		No Connect	D12	SACK–	B14	SRST–	B15	VSS	N17
FETCH/	T19	No Connect	D14	SACK+	A14	SRST+	A15	VSS	U4
GPIO1		No Connect	E2	SACK2–	D2	SSEL–	A18	VSS	U8
MASTER/	R18	No Connect	E3	SACK2+	D3	SSEL+	A17	VSS	U13
GPIO2	R20	No Connect	E4	SATN–	B12	STOP/	W11	VSS	U17
GPIO3	P18	No Connect	E18	SATN+	A12	TCK	K1	VSSA	J18
GPIO4	P19	No Connect	F18	SBSY–	B13	TDI	L3	VSSCORE	M3
GPIO5	W19	No Connect	G3	SBSY+	A13	TDO	L2	VSSCORE	N18
GPIO6	W20	No Connect	G4	SCD–	C17	TEST_HSC/	J19	VSSCORE	P20
GPIO7	V19	No Connect	G17	SCD+	D16	TEST_RST/	K2	VSSCORE2	N1
GPIO8	W2	No Connect	G18	SCLK	J20	TMS	L1		
IDSEL	V6	No Connect	H3	SD0–	C1	TRDY/	W10		
IRDY/	W9	No Connect	H18	SD0+	D1	TRST	Y1		
IRQ/	U20	No Connect	J1	SD1–	B3	VDD	D6		

1. *NC* pins are not connected.

Figure 6.41 LSI53C895A 208-Pin Plastic Quad Flat Pack



1. *NC* pins are not connected.

Table 6.53 Signal Names vs. Pin Number: 208-Pin Plastic Quad Flat Pack

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
AD0	58	GPIO0_		No Connect	155	SD9–	92	VDDIO	73
AD1	57	FETCH/	61	No Connect	156	SD9+	93	VDDIO	81
AD2	51	GPIO1_		No Connect	157	SD10–	89	VDDIO	184
AD3	50	MASTER/	63	No Connect	158	SD10+	90	VDDPCI	2
AD4	48	GPIO2	65	No Connect	159	SD11–	87	VDDPCI	13
AD5	47	GPIO3	66	No Connect	177	SD11+	88	VDDPCI	23
AD6	45	GPIO4	67	No Connect	192	SD12–	175	VDDPCI	26
AD7	44	GPIO5	52	No Connect	207	SD12+	176	VDDPCI	36
AD8	42	GPIO6	54	No Connect	208	SD13–	172	VDDPCI	46
AD9	40	GPIO7	55	PAR	30	SD13+	173	VDDPCI	60
AD10	39	GPIO8	205	PERR/	28	SD14–	170	VDDPCI	197
AD11	38	IDSEL	9	RBIAS	130	SD14+	171	VDD_RBIAS	129
AD12	37	IRDY/	22	REQ/	198	SD15–	167	VDDSCSI	86
AD13	35	IRQ/	59	RST/	194	SD15+	168	VDDSCSI	96
AD14	34	MAC/_		SACK/–	121	SDP0–	132	VDDSCSI	115
AD15	33	TESTOUT	79	SACK/+	122	SDP0+	133	VDDSCSI	125
AD16	19	MAD0	77	SACK2–	160	SDP1–	165	VDDSCSI	134
AD17	17	MAD1	76	SACK2+	161	SDP1+	166	VDDSCSI	144
AD18	16	MAD2	75	SATN–	126	SERR/	29	VDDSCSI	164
AD19	15	MAD3	74	SATN+	127	SIO–	97	VDDSCSI	174
AD20	14	MAD4	72	SBSY–	123	SIO+	98	VSSA	83
AD21	12	MAD5	71	SBSY+	124	SMSG–	116	VSSCORE	68
AD22	11	MAD6	70	SCD–	111	SMSG+	117	VSSCORE	187
AD23	10	MAD7	69	SCD+	112	SREQ–	99	VSSIO	78
AD24	6	MAS0/	186	SCLK	80	SREQ+	100	VSSIO	179
AD25	5	MAS1/	185	SD0–	162	SREQ2–	101	VSSPCI	8
AD26	3	MCE/	191	SD0+	163	SREQ2+	102	VSSPCI	18
AD27	204	MOE/	189	SD1–	149	SRST–	118	VSSPCI	31
AD28	203	MWE/	188	SD1+	150	SRST+	119	VSSPCI	41
AD29	202	No Connect	4	SD2–	147	SSEL–	113	VSSPCI	56
AD30	201	No Connect	49	SD2+	148	SSEL+	114	VSSPCI	193
AD31	199	No Connect	53	SD3–	145	STOP/	27	VSSPCI	200
ALT_IRQ/	1	No Connect	62	SD3+	146	TCK	180	VSSSCSI	91
C_BE0/	43	No Connect	103	SD4–	142	TDI	183	VSSSCSI	110
C_BE1/	32	No Connect	104	SD4+	143	TDO	182	VSSSCSI	120
C_BE2/	20	No Connect	105	SD5–	140	TEST_HSC/	82	VSSSCSI	128
C_BE3/	7	No Connect	106	SD5+	141	TEST_RST/	178	VSSSCSI	131
CLK	195	No Connect	107	SD6–	137	TMS	181	VSSSCSI	139
DEVSEL/	25	No Connect	108	SD6+	138	TRDY/	24	VSSSCSI	151
DIFFSENS	84	No Connect	109	SD7–	135	TRST/	206	VSSSCSI	169
FRAME/	21	No Connect	152	SD7+	136	VDDA	85		
GNT/	196	No Connect	153	SD8–	94	VDDCORE	64		
		No Connect	154	SD8+	95	VDDCORE	190		

1. *NC* pins are not connected.

This page intentionally left blank.

6.6.1 LSI53C895A vs. LSI53C895 Pin/Ball Differences

The LSI53C895A can be used as a drop-in replacement for the LSI53C895. The LSI53C895A is packaged in a 208 PQFP and alternatively a 272 BGA.

Note: The BGA package for the LSI53C895 is a 292 BGA with the only difference being the center grid is 6 x 6 balls vs. the LSI53C895A's 272 BGA which has a center grid of 4 x 4 balls. The LSI53C895A 272 BGA can be dropped onto a board that has been laid out for the LSI53C895 292 BGA with no changes to the board.

Some of the original LSI53C895 no-connections have been used for new signals. In addition, several of the previous LSI53C895 signals have been renamed, and some of them are no longer used and are now no-connections.

Table 6.54 indicates the differences between the LSI53C895A and the LSI53C895 signal names and locations.

Table 6.54 LSI53C895A vs. LSI53C895 Pin/Ball Differences

Pin/Ball Name	Type	208 PQFP Pin #	272 BGA Ball #	Description
ALT_IRQ/	I/O	1	Y2	Previously NC. Alternate Interrupt Request, when asserted low, indicates that an interrupting condition has occurred in the SCSI Function and that service is required from the host CPU. The output drive of this pin is open drain.
SREQ2– SREQ2+	I/O I/O	101 102	B20 B19	Previously NC. SREQ2– and SREQ2+ are data handshake lines from target device. They are duplicates of SREQ– and SREQ+ enabled by pulling MAD5 HIGH at reset.
SACK2– SACK2+	I/O I/O	160 161	D2 D3	Previously NC. SACK2– and SACK2+ are data handshake lines from the initiator device. They are duplicates of SACK– and SACK+ enabled by pulling MAD5 HIGH at reset.
TEST_RST/	I	178	K2	Previously TESTIN. TEST_RST/ is used for LSI Logic test purposes only to reset the LSI53C895A. It should not be driven LOW.
TCK	I	180	K1	Previously TEST. TCK provides the clock for the JTAG test logic.
TMS	I	181	L1	Previously TEST. TMS is decoded by the TAP controller and is used to control the JTAG test operations.
TDO	O	182	L2	Previously TEST. TDO is a serial output used for test instructions and data from the JTAG test logic.
TDI	I	183	L3	Previously TEST. TDI is a serial input used for test instructions and data to the JTAG test logic.
TRST/	I	206	Y1	Previously NC. TRST/ provides a reset for the JTAG test logic.
GPIO5 GPIO6 GPIO7 GPIO8	I/O I/O I/O I/O	52 54 55 205	W19 W20 V19 W2	Previously NC. GPIO[8:5] are SCSI General Purpose I/O pins. These pins power-up as inputs.

Table 6.54 LSI53C895A vs. LSI53C895 Pin/Ball Differences (Cont.)

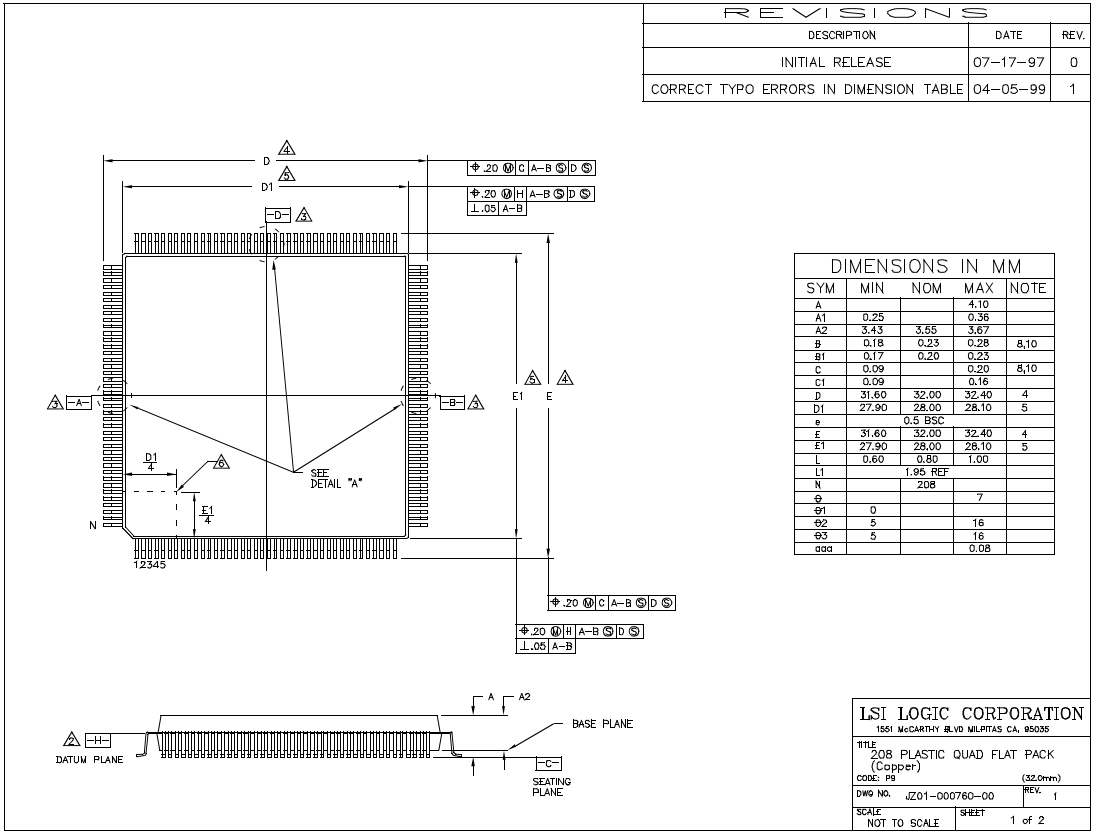
Pin/Ball Name	Type	208 PQFP Pin #	272 BGA Ball #	Description
TEST_HSC/	I	82	J19	Previously TEST. TEST_HSC/ is used for LSI Logic test purposes only. This signal can cause a full chip reset.
V_{DD}_RBIAS RBIAS	I	129 130	A11 A10	Previously RBIAS– and RBIAS+. V_{DD}_RBIAS and RBIAS are functionally the same as RBIAS– and RBIAS+ in the LSI53C895. They are used to connect an external resistor to generate the bias current used by LVDlink pads.
NC	N/A	4, 49	Y4, W18	Previously V5BIAS(P). It was used for voltage biasing of the PCI signals. The biasing is now automatically handled internally.
NC	N/A	62	T20	Previously V5BIAS(M). It was used for voltage biasing of the external memory interface signals. The biasing is now automatically handled internally.
NC	N/A	177	J1	Previously TEST. It was never used in actual board design.
NC	N/A	192	P3	Previously BIG_LIT/. It was used for big or little endian selection, and is no longer used. The LSI53C895A is little endian only.

LSI Logic component dimensions conform to a current revision of the JEDEC Publication 95 standard package outline, using ANSI 14.5Y “Dimensioning and Tolerancing” interpretations. As JEDEC drawings are balloted and updated, changes may have occurred. To ensure the use of a current drawing, the JEDEC drawing revision level should be verified. Visit www.eia.org/jedec for review of Publication 95 drawings and revision levels.

For printed circuit board land patterns that will accept LSI Logic components, it is recommended that customers refer to the IPC standards (Institute for Interconnecting and Packaging Electronic Circuits). Specification number IPC-SM-782, *Surface Mount Design and Land Pattern Standard* is an established method of designing land patterns. Feature size and tolerances are industry standards based on IPC assumptions.

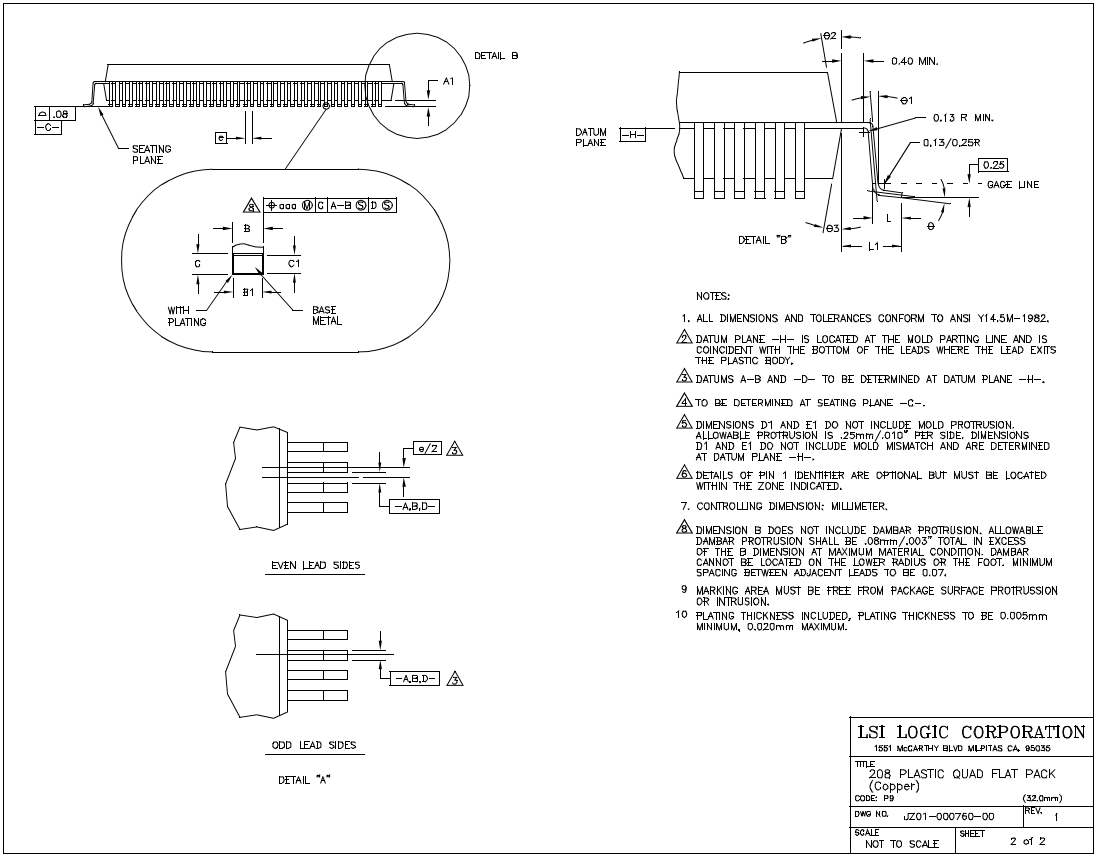
Figure 6.42 is the mechanical drawing for the 208 PQFP and Figure 6.43 is the mechanical drawing for the 272 PBGA for the LSI53C895A.

Figure 6.42 LSI53C895A 208 PQFP Mechanical Drawing (Sheet 1 of 2)



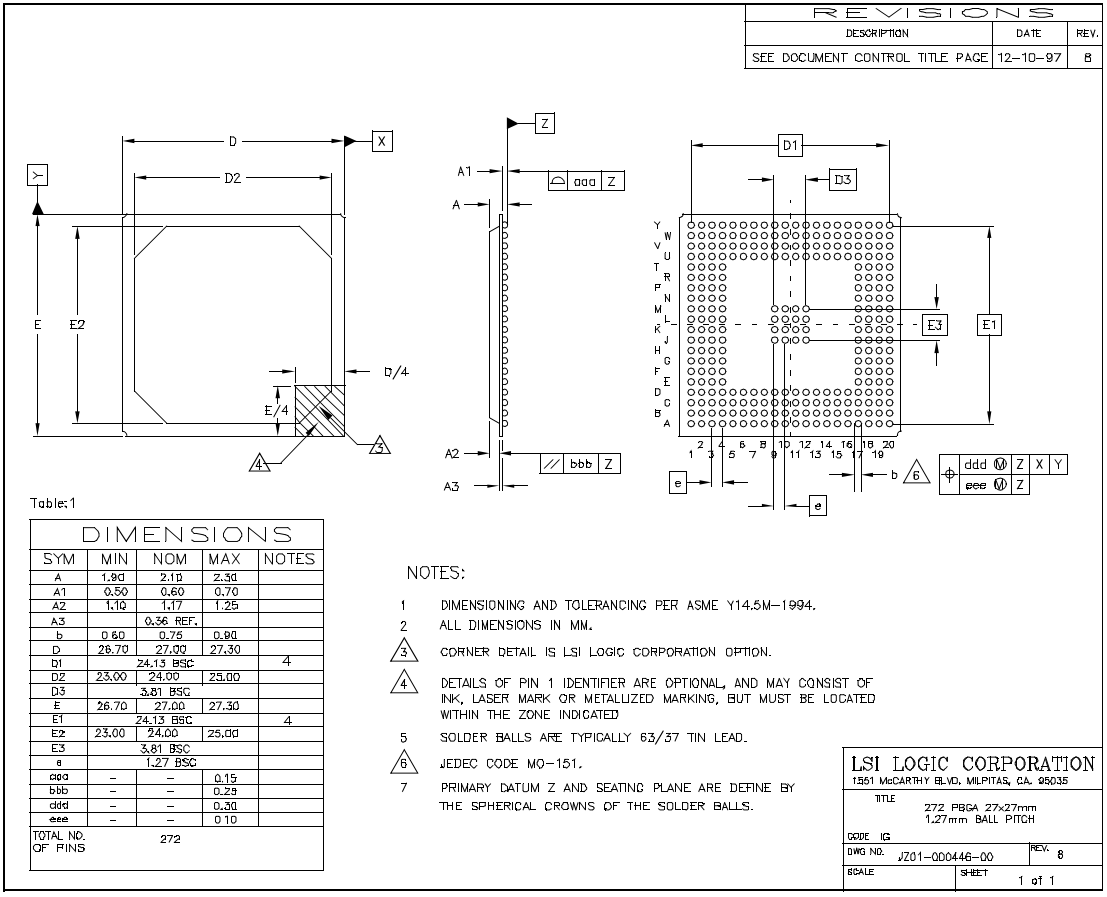
Important: **This drawing may not be the latest version. For board layout and manufacturing, obtain the most recent engineering drawings from your LSI Logic marketing representative by requesting the outline drawing for package code P9.**

Figure 6.42 208-Pin PQFP (P9) Mechanical Drawing (Sheet 2 of 2)



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 P9.**

Figure 6.43 LSI53C895A 272 PBGA 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 IG.**

Appendix A
Register Summary

Table A.1 lists the PCI register summary by register name for the LSI53C895A.

Table A.1 LSI53C895A PCI Register Map

Register Name	Address	Read/Write	Page
Base Address Register One (MEMORY)	0x14–0x17	Read/Write	4-9
Base Address Register Two (SCRIPTS RAM)	0x18–0x1B	Read/Write	4-10
Base Address Register Zero (I/O)	0x10–0x13	Read/Write	4-9
Bridge Support Extensions (PMCSR_BSE)	0x46	Read Only	4-18
Cache Line Size	0x0C	Read/Write	4-7
Capabilities Pointer	0x34	Read Only	4-13
Capability ID	0x40	Read Only	4-15
Class Code	0x09–0x0B	Read Only	4-7
Command	0x04–0x05	Read/Write	4-3
Data	0x47	Read Only	4-18
Device ID	0x02–0x03	Read Only	4-3
Expansion ROM Base Address	0x30–0x33	Read/Write	4-12
Header Type	0x0E	Read Only	4-8
Interrupt Line	0x3C	Read/Write	4-13
Interrupt Pin	0x3D	Read Only	4-14
Latency Timer	0x0D	Read/Write	4-8
Max_Lat	0x3F	Read Only	4-15

Table A.1 LSI53C895A PCI Register Map (Cont.)

Register Name	Address	Read/Write	Page
Min_Gnt	0x3E	Read Only	4-14
Next Item Pointer	0x41	Read Only	4-15
Power Management Capabilities (PMC)	0x42–0x43	Read Only	4-16
Power Management Control/Status (PMCSR)	0x44–0x45	Read/Write	4-17
Reserved	0x28–0x2B	–	4-10
Reserved	0x35–0x3B	–	4-13
Revision ID (Rev ID)	0x08	Read Only	4-7
Status	0x06–0x07	Read/Write	4-5
Subsystem ID	0x2E–0x2F	Read Only	4-11
Subsystem ID Access	0x48–0x4B	Write Only	4-18
Subsystem Vendor ID	0x2C–0x2D	Read Only	4-10
Vendor ID	0x00–0x01	Read Only	4-3

Table A.2 lists the SCSI register summary by register name for the LSI53C895A.

Table A.2 LSI53C895A SCSI Register Map

Register Name	Address	Read/Write	Page
Adder Sum Output (ADDER)	0x3C–0x3F	Read Only	4-74
Chip Control 0 (CCNTL0)	0x56	Read/Write	4-98
Chip Control 1 (CCNTL1)	0x57	Read/Write	4-100
Chip Test Five (CTEST5)	0x22	Read/Write	4-62
Chip Test Four (CTEST4)	0x21	Read/Write	4-60
Chip Test One (CTEST1)	0x19	Read Only	4-54
Chip Test Six (CTEST6)	0x23	Read/Write	4-63
Chip Test Three (CTEST3)	0x1B	Read/Write	4-57

Table A.2 LSI53C895A SCSI Register Map (Cont.)

Register Name	Address	Read/Write	Page
Chip Test Two (CTEST2)	0x1A	Read Only (bit 3 write)	4-55
Chip Test Zero (CTEST0)	0x18	Read/Write	4-54
Cumulative SCSI Byte Count (CSBC)	0xDC–0xDF	Read/Write	4-112
Data Structure Address (DSA)	0x10–0x13	Read/Write	4-48
DMA Byte Counter (DBC)	0x24–0x26	Read/Write	4-64
DMA Command (DCMD)	0x27	Read/Write	4-65
DMA Control (DCNTL)	0x3B	Read/Write	4-71
DMA FIFO (DFIFO)	0x20	Read/Write	4-58
DMA Interrupt Enable (DIEN)	0x39	Read/Write	4-70
DMA Mode (DMODE)	0x38	Read/Write	4-67
DMA Next Address (DNAD)	0x28–0x2B	Read/Write	4-65
DMA Next Address 64 (DNAD64)	0xB8–0xBB	Read/Write	4-107
DMA SCRIPTS Pointer (DSP)	0x2C–0x2F	Read/Write	4-65
DMA SCRIPTS Pointer Save (DSPS)	0x30–0x33	Read/Write	4-66
DMA Status (DSTAT)	0x0C	Read Only	4-40
DSA Relative Selector (DRS)	0xAC–0xAF	Read/Write	4-105
Dynamic Block Move Selector (DBMS)	0xB4–0xB7	Read/Write	4-107
Entry Storage Address (ESA)	0xD0–0xD3	Read/Write	4-111
General Purpose (GPREG0)	0x07	Read/Write	4-36
General Purpose One (GPREG1)	0x5B	Read/Write	4-102
General Purpose Pin Control One (GPCNTL1)	0x5A	Read/Write	4-102
General Purpose Pin Control Zero (GPCNTL0)	0x47	Read/Write	4-84
Instruction Address (IA)	0xD4–0xD7	Read/Write	4-111
Interrupt Status One (ISTAT1)	0x15	Read/Write	4-52
Interrupt Status Zero (ISTAT0)	0x14	Read/Write	4-48

Table A.2 LSI53C895A SCSI Register Map (Cont.)

Register Name	Address	Read/Write	Page
Mailbox One (MBOX1)	0x17	Read/Write	4-53
Mailbox Zero (MBOX0)	0x16	Read/Write	4-53
Memory Access Control (MACNTL)	0x46	Read/Write	4-83
Memory Move Read Selector (MMRS)	0xA0–0xA3	Read/Write	4-103
Memory Move Write Selector (MMWS)	0xA4–0xA7	Read/Write	4-104
Phase Mismatch Jump Address 1 (PMJAD1)	0xC0–0xC3	Read/Write	4-108
Phase Mismatch Jump Address 2 (PMJAD2)	0xC4–0xC7	Read/Write	4-109
Remaining Byte Count (RBC)	0xC8–0xCB	Read/Write	4-109
Reserved	0x53	–	4-97
Reserved	0xBC–0xBF	–	4-107
Reserved	0xDB	–	4-112
Reserved	0xE0–0xFF	–	4-113
Response ID One (RESPID1)	0x4B	Read/Write	4-88
Response ID Zero (RESPID0)	0x4A	Read/Write	4-88
Scratch Byte Register (SBR)	0x3A	Read/Write	4-71
Scratch Register A (SCRATCHA)	0x34–0x37	Read/Write	4-67
Scratch Register B (SCRATCHB)	0x5C–0x5F	Read/Write	4-103
Scratch Registers C–R (SCRATCHC–SCRATCHR)	0x60–x9F	Read/Write	4-103
SCRIPTS Fetch Selector (SFS)	0xA8–0xAB	Read/Write	4-105
SCSI Bus Control Lines (SBCL)	0x0B	Read Only	4-39
SCSI Bus Data Lines (SBDL)	0x58–0x59	Read Only	4-101
SCSI Byte Count (SBC)	0xD8–0xDA	Read only	4-112
SCSI Chip ID (SCID)	0x04	Read/Write	4-31
SCSI Control One (SCNTL1)	0x01	Read/Write	4-24
SCSI Control Three (SCNTL3)	0x03	Read/Write	4-29

Table A.2 LSI53C895A SCSI Register Map (Cont.)

Register Name	Address	Read/Write	Page
SCSI Control Two (SCNTL2)	0x02	Read/Write	4-27
SCSI Control Zero (SCNTL0)	0x00	Read/Write	4-21
SCSI Destination ID (SDID)	0x06	Read/Write	4-36
SCSI First Byte Received (SFBR)	0x08	Read/Write	4-37
SCSI Input Data Latch (SIDL)	0x50–0x51	Read Only	4-96
SCSI Interrupt Enable One (SIEN1)	0x41	Read/Write	4-76
SCSI Interrupt Enable Zero (SIEN0)	0x40	Read/Write	4-74
SCSI Interrupt Status One (SIST1)	0x43	Read Only	4-80
SCSI Interrupt Status Zero (SIST0)	0x42	Read Only	4-77
SCSI Longitudinal Parity (SLPAR)	0x44	Read/Write	4-81
SCSI Output Control Latch (SOCL)	0x09	Read/Write	4-38
SCSI Output Data Latch (SODL)	0x54–0x55	Read/Write	4-97
SCSI Selector ID (SSID)	0x0A	Read Only	4-39
SCSI Status One (SSTAT1)	0x0E	Read Only	4-44
SCSI Status Two (SSTAT2)	0x0F	Read Only	4-46
SCSI Status Zero (SSTAT0)	0x0D	Read Only	4-43
SCSI Test Four (STEST4)	0x52	Read Only	4-96
SCSI Test One (STEST1)	0x4D	Read/Write	4-90
SCSI Test Three (STEST3)	0x4F	Read/Write	4-93
SCSI Test Two (STEST2)	0x4E	Read/Write	4-91
SCSI Test Zero (STEST0)	0x4C	Read Only	4-89
SCSI Timer One (STIME1)	0x49	Read/Write	4-87

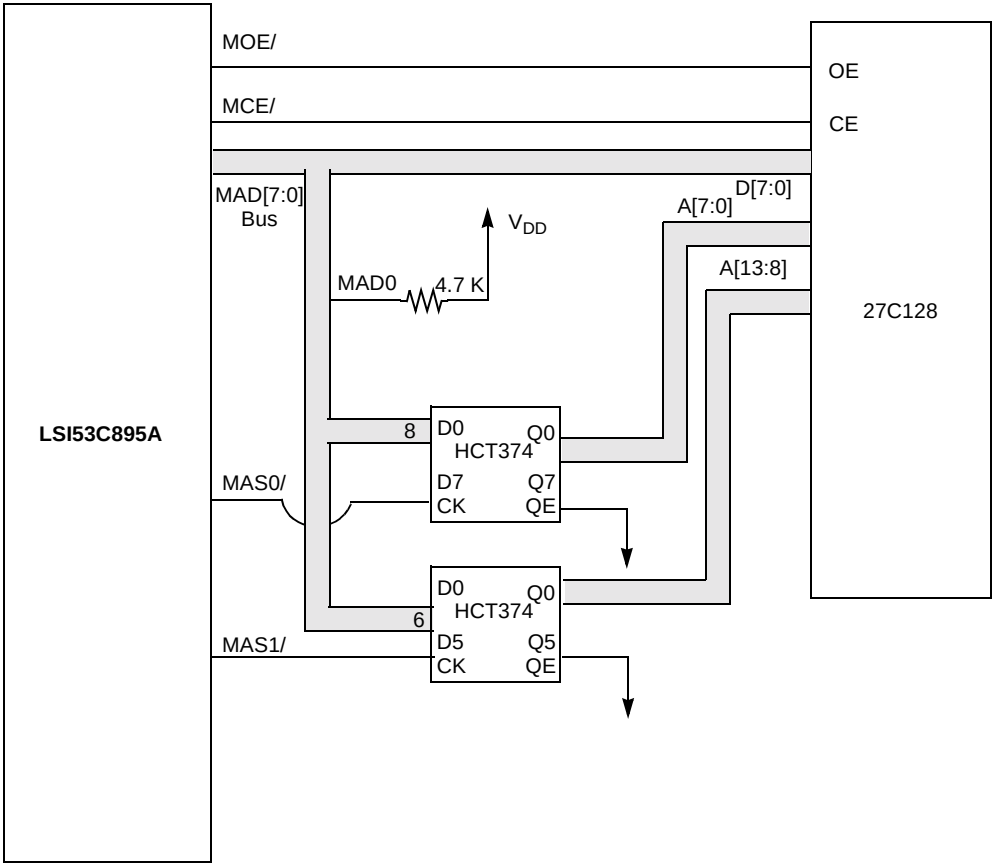
Table A.2 LSI53C895A SCSI Register Map (Cont.)

Register Name	Address	Read/Write	Page
SCSI Timer Zero (STIME0)	0x48	Read/Write	4-85
SCSI Transfer (SXFER)	0x05	Read/Write	4-32
SCSI Wide Residue (SWIDE)	0x45	Read/Write	4-82
Static Block Move Selector (SBMS)	0xB0–0xB3	Read/Write	4-106
Temporary (TEMP)	0x1C–0x1F	Read/Write	4-58
Updated Address (UA)	0xCC–0xCF	Read/Write	4-110

Appendix B
External Memory Interface Diagram Examples

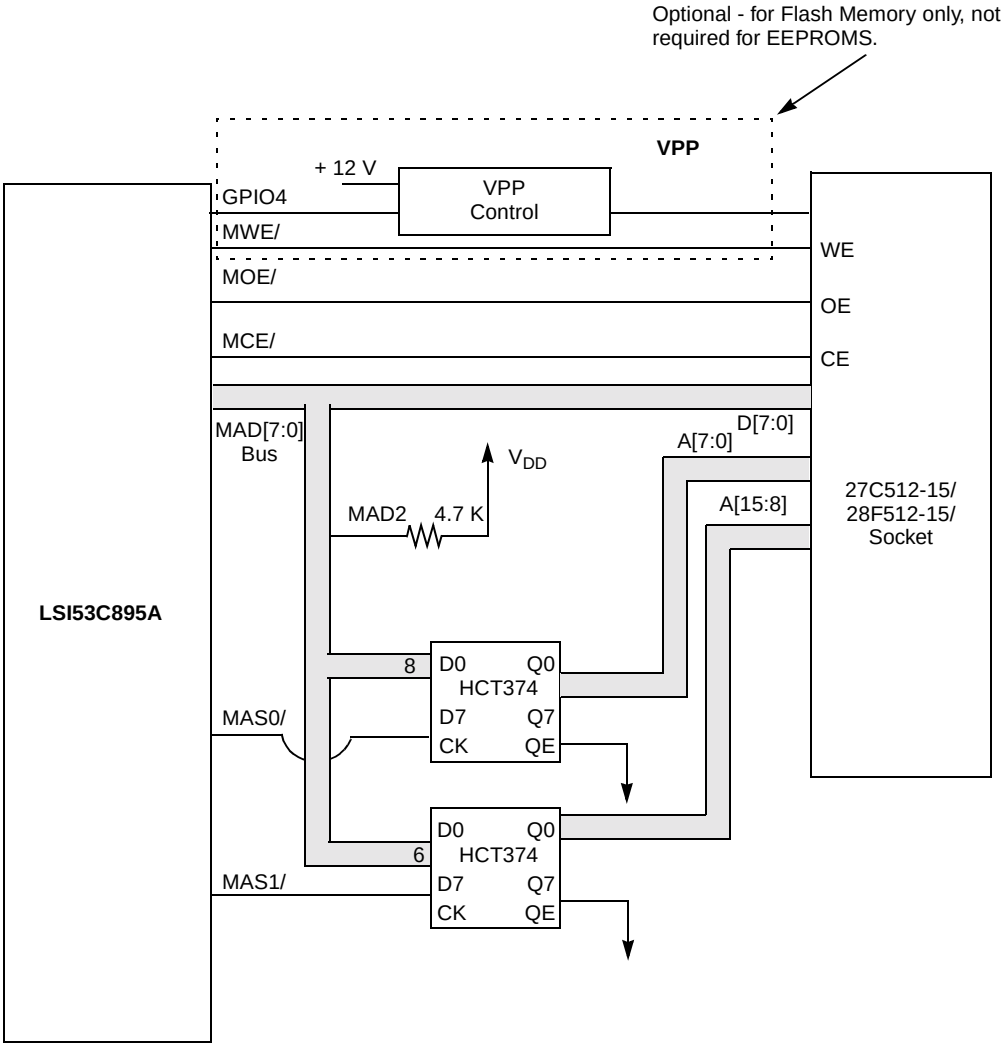
Appendix B has example external memory interface diagrams.

Figure B.1 16 Kbyte Interface with 200 ns Memory



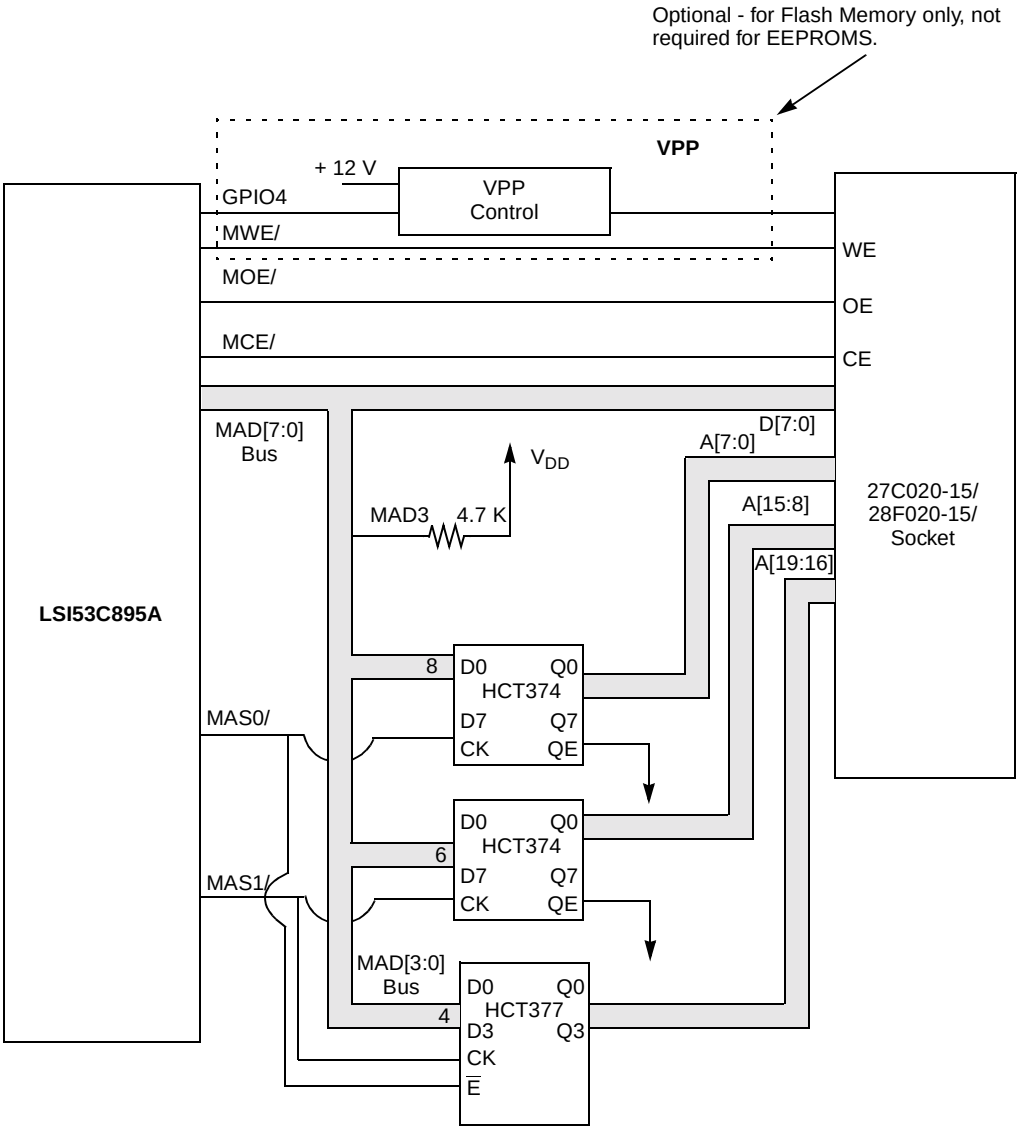
Note: MAD[3:1] pulled LOW internally. MAD bus sense logic enabled for 16 Kbyte of slow memory (200 ns devices @ 33 MHz).

Figure B.2 64 Kbyte Interface with 150 ns Memory



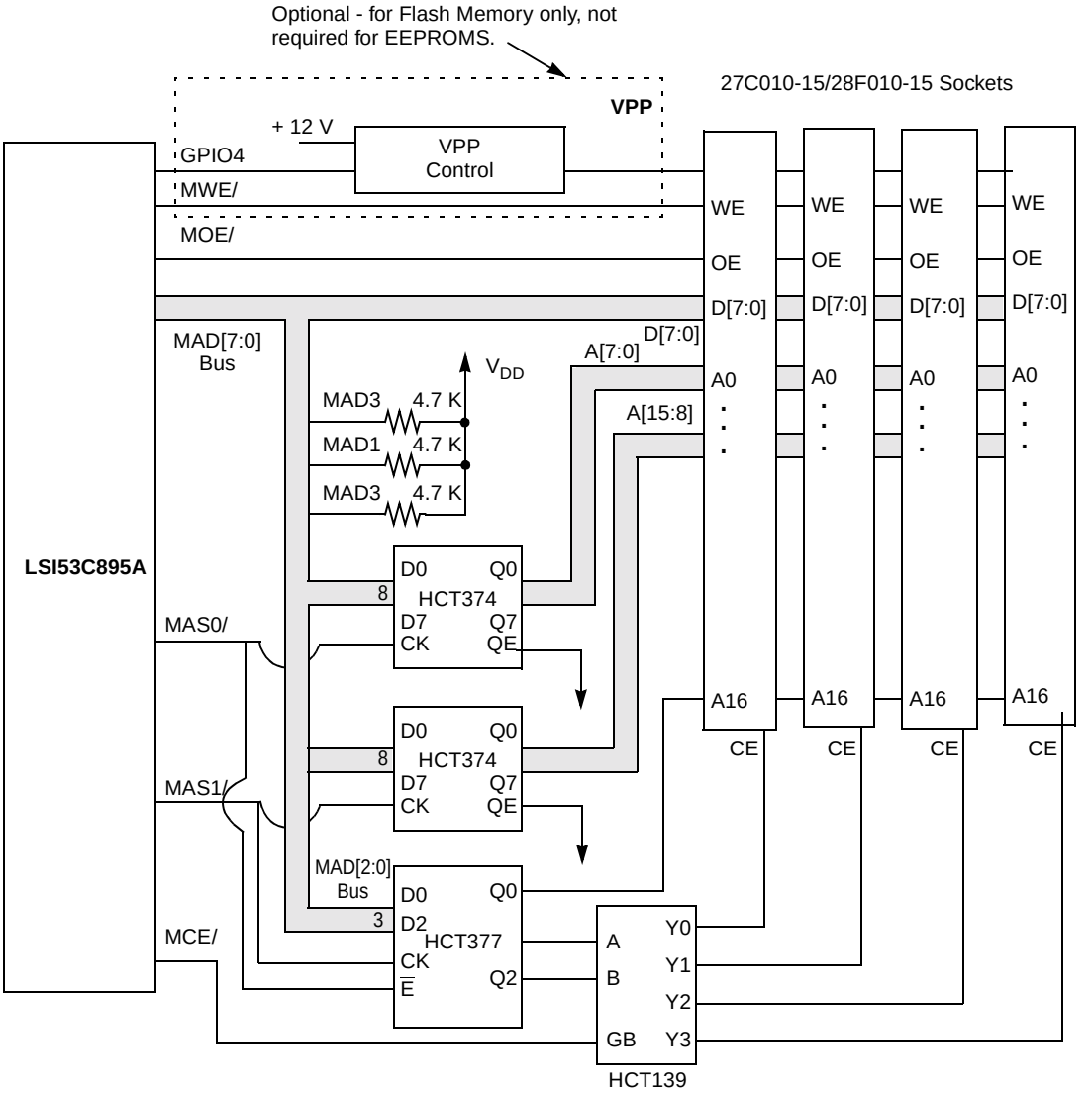
Note: MAD 3, 1, 0 pulled LOW internally. MAD bus sense logic enabled for 64 Kbyte of fast memory (150 ns devices @ 33 MHz).

Figure B.3 128 Kbytes, 256 Kbytes, 512 Kbytes, 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: *LSI53C895A PCI to Ultra2 SCSI 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 (Components and Boards)

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