

# **TECHNICAL MANUAL**

## **LSI53C825A/825AE PCI to SCSI I/O Processor**

*Version 3.1*

**January 2001**

---

This document contains proprietary information of LSI Logic Corporation. The information contained herein is not to be used by or disclosed to third parties without the express written permission of an officer of LSI Logic Corporation.

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 DB14-000159-00, Fourth Edition (January 2001)

This document describes the LSI Logic LSI53C825A/AE PCI to SCSI I/O Processor and will remain the official reference source for all revisions/releases of this product until rescinded by an update.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.lsillogic.com>.**

LSI Logic Corporation reserves the right to make changes to any products herein at any time without notice. LSI Logic does not assume any responsibility or liability arising out of the application or use of any product described herein, except as expressly agreed to in writing by LSI Logic; nor does the purchase or use of a product from LSI Logic convey a license under any patent rights, copyrights, trademark rights, or any other of the intellectual property rights of LSI Logic or third parties.

Copyright © 1998–2001 by LSI Logic Corporation. All rights reserved.

#### TRADEMARK ACKNOWLEDGMENT

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

# Preface

---

This book is the primary reference and technical manual for the LSI Logic LSI53C825A/825AE PCI to SCSI I/O Processor. It contains a complete functional description for the LSI53C825A/825AE and includes complete physical and electrical specifications for the LSI53C825A/825AE.

---

## Audience

This technical manual is intended for system designers and programmers who are using this device to design a SCSI port for PCI-based personal computers, workstations, or embedded applications.

---

## Organization

This document has the following chapters and appendixes:

- [Chapter 1, Introduction](#), includes general information about the LSI53C825A and other members of the LSI53C8XX family of PCI to SCSI I/O Processors.
- [Chapter 2, Functional Description](#), describes the main functional areas of the chip in more detail, including the interfaces to the SCSI bus.
- [Chapter 3, Signal Descriptions](#), describes the chip's connection to the PCI bus, including the PCI commands and configuration registers supported.
- [Chapter 4, Registers](#), contains the pin diagrams and definitions of each signal.
- [Chapter 5, SCSI SCRIPTS Instruction Set](#), describes each bit in the operating registers, organized by address.

- [Chapter 6, Specifications](#), defines all of the SCSI SCRIPTS instructions that are supported by the LSI53C825A.
- [Appendix A, Register Summary](#), contains a register summary.
- [Appendix B, External Memory Interface Diagram Examples](#), contains several example interface drawings to connect the LSI53C825A to an external ROM.

---

## 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.lsillogic.com](http://www.lsillogic.com)

SCSI SCRIPTS™ Processors Programming Guide, Version 2.2,  
Order Number S15044.A

**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
1.0	6/95	Revision 1.0
2.0	3/96	Revision 2.0
3.0	12/97	Revision 3.0
3.1	1/01	Product names changed from SYM to LSI.



# Contents

---

<b>Chapter 1</b>	<b>Introduction</b>	
1.1	General Description	1-1
1.2	Package and Feature Options	1-2
1.2.1	PCI Pad Power-up Sequence	1-3
1.3	TolerANT <sup>®</sup> Technology	1-3
1.4	LSI53C825A Benefits Summary	1-4
1.4.1	SCSI Performance	1-4
1.4.2	PCI Performance	1-5
1.4.3	Integration	1-6
1.4.4	Ease of Use	1-6
1.4.5	Flexibility	1-7
1.4.6	Reliability	1-7
1.4.7	Testability	1-8

---

<b>Chapter 2</b>	<b>Functional Description</b>	
2.1	PCI Addressing	2-1
2.1.1	Configuration Space	2-1
2.1.2	PCI Bus Commands and Functions Supported	2-2
2.1.3	PCI Cache Mode	2-4
2.2	SCSI Functional Description	2-10
2.2.1	SCSI Core	2-10
2.2.2	DMA Core	2-11
2.2.3	SCRIPTS Processor	2-11
2.2.4	Internal SCRIPTS RAM	2-11
2.2.5	SDMS: The Total SCSI Solution	2-12
2.2.6	Prefetching SCRIPTS Instructions	2-13
2.2.7	Opcode Fetch Burst Capability	2-14
2.3	External Memory Interface	2-14
2.4	PCI Cache Mode	2-17

2.4.1	Load and Store Instructions	2-17
2.4.2	3.3 V/5 V PCI Interface	2-17
2.4.3	Additional Access to General Purpose Pins	2-17
2.4.4	JTAG Boundary Scan Testing	2-18
2.4.5	Big and Little Endian Support	2-19
2.4.6	Loopback Mode	2-20
2.4.7	Parity Options	2-21
2.4.8	DMA FIFO	2-23
2.4.9	SCSI Bus Interface	2-27
2.4.10	Select/Reselect During Selection/Reselection	2-33
2.4.11	Synchronous Operation	2-33
2.4.12	Achieving Optimal SCSI Send Rates	2-34
2.4.13	Interrupt Handling	2-35
2.4.14	Chained Block Moves	2-42
2.5	Power Management	2-46
2.5.1	Power State D0	2-46
2.5.2	Power State D3	2-46

---

## Chapter 3

### Signal Descriptions

3.1	PCI Bus Interface Signals	3-6
3.1.1	System Signals	3-6
3.1.2	Address and Data Signals	3-7
3.1.3	Interface Control Signals	3-8
3.1.4	Arbitration Signals	3-9
3.1.5	Error Reporting Signals	3-9
3.1.6	SCSI Bus Interface Signals	3-10
3.1.7	Additional Interface Signals	3-11
3.1.8	External Memory Interface Signals	3-14
3.1.9	JTAG Signals	3-15
3.2	MAD Bus Programming	3-15

---

## Chapter 4

### Registers

4.1	Configuration Registers	4-1
4.2	Operating Registers	4-18



---

<b>Chapter 5</b>	<b>SCSI SCRIPTS Instruction Set</b>	
5.1	Low Level Register Interface Mode	5-1
5.2	High Level SCSI SCRIPTS Mode	5-2
5.2.1	Sample Operation	5-3
5.3	Block Move Instructions	5-6
5.3.1	First Dword	5-6
5.3.2	Second Dword	5-13
5.4	I/O Instruction	5-14
5.4.1	First Dword	5-14
5.4.2	Second Dword	5-23
5.5	Read/Write Instructions	5-24
5.5.1	First Dword	5-24
5.5.2	Second Dword	5-26
5.5.3	Read-Modify-Write Cycles	5-26
5.5.4	Move To/From SFBR Cycles	5-27
5.6	Transfer Control Instructions	5-29
5.6.1	First Dword	5-29
5.6.2	Second Dword	5-36
5.7	Memory Move Instructions	5-36
5.7.1	First Dword	5-37
5.7.2	Read/Write System Memory from SCRIPTS	5-38
5.7.3	Second Dword	5-38
5.7.4	Third Dword	5-38
5.8	Load and Store Instructions	5-40
5.8.1	First Dword	5-41
5.8.2	Second Dword	5-42

---

<b>Chapter 6</b>	<b>Specifications</b>	
6.1	DC Characteristics	6-1
6.2	TolerANT Technology Electrical Characteristics	6-7
6.3	AC Characteristics	6-11
6.4	PCI and External Memory Interface Timing Diagrams	6-13
6.4.1	Target Timing	6-15
6.4.2	Initiator Timing	6-24
6.4.3	External Memory Timing	6-32

6.5	PCI and External Memory Interface Timing	6-44
6.6	SCSI Timing Diagrams	6-45
6.7	Package Drawings	6-52

---

<b>Appendix A</b>	<b>Register Summary</b>
-------------------	-------------------------

---

<b>Appendix B</b>	<b>External Memory Interface Diagram Examples</b>
-------------------	---

---

<b>Index</b>
--------------

---

<b>Customer Feedback</b>
--------------------------

---

**Figures**

1.1	LSI53C825A External Memory Interface	1-9
1.2	LSI53C825A Chip Block Diagram	1-10
2.1	DMA FIFO Sections	2-23
2.2	LSI53C825A Host Interface Data Paths	2-27
2.3	LSI53C825A Differential Wiring Diagram	2-31
2.4	Regulated Termination	2-32
2.5	Determining the Synchronous Transfer Rate	2-35
2.6	Block Move and Chained Block Move Instructions	2-45
3.1	LSI53C825A Pin Diagram	3-2
3.2	LSI53C825AJ Pin Diagram	3-3
3.3	LSI53C825A Functional Signal Grouping	3-5
5.1	SCRIPTS Overview	5-5
5.2	Block Move Instruction Register	5-8
5.3	I/O Instruction Register	5-17
5.4	Read/Write Instruction Register	5-25
5.5	Transfer Control Instruction	5-31
5.6	Memory Move Instruction	5-39
5.7	Load and Store Instruction Format	5-43
6.1	Rise and Fall Time Test Conditions	6-9
6.2	SCSI Input Filtering	6-9
6.3	Hysteresis of SCSI Receivers	6-9
6.4	Input Current as a Function of Input Voltage	6-10
6.5	Output Current as a Function of Output Voltage	6-10

6.6	External Clock	6-11
6.7	Reset Input	6-12
6.8	Interrupt Output	6-13
6.9	PCI Configuration Register Read	6-15
6.10	PCI Configuration Register Write	6-16
6.11	Operating Register/SCRIPTS RAM Read	6-17
6.12	Operating Register/SCRIPTS RAM Write	6-18
6.13	External Memory Read	6-20
6.14	External Memory Write	6-22
6.15	Nonburst Opcode Fetch	6-24
6.16	Burst Opcode Fetch	6-25
6.17	Back-to-Back Read	6-26
6.18	Back-to-Back Write	6-27
6.19	Burst Read	6-28
6.20	Burst Write	6-30
6.21	Read Cycle, Normal/Fast Memory ( $\geq 64$ Kbytes), Single Byte Access	6-32
6.22	Write Cycle, Normal/Fast Memory ( $\geq 64$ Kbytes), Single Byte Access	6-33
6.23	Read Cycle, Normal/Fast Memory ( $\geq 64$ Kbytes), Multiple Byte Access	6-34
6.24	Write Cycle, Normal/Fast Memory ( $\geq 64$ Kbytes), Multiple Byte Access	6-36
6.25	Read Cycle, Slow Memory ( $\geq 64$ Kbytes)	6-38
6.26	Write Cycle, Slow Memory ( $\geq 64$ Kbytes)	6-39
6.27	Read Cycle, Normal/Fast Memory ( $\geq 64$ Kbytes)	6-40
6.28	Write Cycle, Normal/Fast Memory ( $\geq 64$ Kbytes)	6-41
6.29	Read Cycle, Slow Memory ( $\leq 64$ Kbytes)	6-42
6.30	Write Cycle, Slow Memory ( $\leq 64$ Kbytes)	6-43
6.31	Initiator Asynchronous Send	6-45
6.32	Initiator Asynchronous Receive	6-46
6.33	Target Asynchronous Send	6-47
6.34	Target Asynchronous Receive	6-48
6.35	Initiator and Target Synchronous Transfers	6-48
6.36	LSI53C825A 160 Pin PQFP (PF) Mechanical Drawing	6-52
B.1	64 Kbyte Interface with 200 ns Memory	B-1

B.2	64 Kbyte Interface with 150 ns Memory	B-2
B.3	256 Kbyte Interface with 150 ns Memory	B-3
B.4	512 Kbyte Interface with 150 ns Memory	B-4

---

## Tables

2.1	PCI Bus Commands and Encoding Types	2-3
2.2	External Memory Support	2-16
2.3	Bits Used for Parity Control and Generation	2-21
2.4	SCSI Parity Control	2-22
2.5	SCSI Parity Errors and Interrupts	2-23
2.6	Differential Mode	2-28
3.1	LSI53C825A, LSI53C825AJ, LSI53C825AE, and LSI53C825AJE Power and Ground Pins	3-4
3.2	System Signals	3-6
3.3	Address and Data Signals	3-7
3.4	Interface Control Signals	3-8
3.5	Arbitration Signals	3-9
3.6	Error Reporting Signals	3-9
3.7	SCSI Bus Interface Signals	3-10
3.8	Additional Interface Signals	3-11
3.9	External Memory Interface Signals	3-14
3.10	JTAG Signals (LSI53C825AJ, LSI53C825AJE Only)	3-15
3.11	Subsystem Data Configuration Table for the LSI53C825AE (PCI Rev ID 0x26)	3-16
3.12	Subsystem Data Configuration Table for the LSI53C825A (PCI Rev ID 0x14) Revision G Only	3-16
3.13	External Memory Support	3-17
4.1	PCI Configuration Register Map	4-2
4.2	LSI53C825A Register Map	4-19
4.3	Synchronous Clock Conversion Factor	4-29
4.4	Examples of Synchronous Transfer Periods and Rates for SCSI-1	4-33
4.5	Example Transfer Periods and Rates for Fast SCSI-2	4-33
4.6	Maximum Synchronous Offset	4-34
4.7	Timeout Periods	4-82
4.8	Timeout Periods, 50 MHz Clock	4-83
5.1	SCRIPTS Instructions	5-3
5.2	Read/Write Instructions	5-27

6.1	Absolute Maximum Stress Ratings	6-2
6.2	Operating Conditions	6-2
6.3	SCSI Signals—SD[15:0]/, SDP[1:0]/, SREQ/, SACK/	6-3
6.4	SCSI Signals—SMSG, SI_O/, SC_D/, SATN/, SBSY/, SSEL/, SRST/	6-3
6.5	Input Signals—CLK, SCLK, GNT/, IDSEL, RST/, TESTIN, DIFFSENS, BIG_LIT/	6-3
6.6	Capacitance	6-4
6.7	Output Signals—MAC/_TESTOUT, REQ/	6-4
6.8	Output Signals—IRQ/, SDIR[15:0], SDIRP0, SDIRP1, BSYDIR, SELDIR, RSTDIR, TGS, IGS, MAS/[1:0], MCE/, MOE/, MWE/	6-4
6.9	Output Signal—SERR/	6-4
6.10	Bidirectional Signals—AD[31:0], C_BE[3:0], FRAME/, IRDY/, TRDY/, DEVSEL/, STOP/, PERR/, PAR/	6-5
6.11	Bidirectional Signals—GPIO0_FETCH/, GPIO1_MASTER/, GPIO2_MAS2/, GPIO3, GPIO4	6-5
6.12	Bidirectional Signals—MAD[7:0]	6-6
6.13	Input Signals—TDI, TMS, TCK (LSI53C825AJ only)	6-6
6.14	Output Signal—TDO (LSI53C825AJ only)	6-6
6.15	TolerANT Technology Electrical Characteristics	6-8
6.16	External Clock	6-11
6.17	Reset Input	6-12
6.18	Interrupt Output	6-13
6.19	LSI53C825A PCI and External Memory Interface Timing	6-44
6.20	Initiator Asynchronous Send	6-45
6.21	Initiator Asynchronous Receive	6-46
6.22	Target Asynchronous Send	6-47
6.23	Target Asynchronous Receive	6-48
6.24	SCSI-1 Transfers (SE 5.0 Mbytes)	6-49
6.25	SCSI-1 Transfers (Differential, 4.17 Mbytes/s)	6-49
6.26	SCSI-2 Fast Transfers (10.0 Mbytes/s (8-Bit Transfers) or 20.0 Mbytes/s (16-Bit Transfers), 40 MHz Clock)	6-50
6.27	SCSI-2 Fast Transfers 10.0 Mbytes (8-Bit Transfers) or 20.0 Mbytes/s (16-Bit Transfers) 50 MHz Clock	6-50
A.1	Configuration Registers	A-1
A.2	LSI53C825A Register Map	A-2



# Chapter 1

## Introduction

---

This chapter includes general information about the LSI53C825A and other members of the LSI53C8XX family of PCI to SCSI I/O Processors and contains the following sections:

- [Section 1.1, “General Description”](#)
- [Section 1.2, “Package and Feature Options”](#)
- [Section 1.3, “TolerANT® Technology”](#)
- [Section 1.4, “LSI53C825A Benefits Summary”](#)

---

### 1.1 General Description

This manual combines information on the LSI53C825A and LSI53C825AE, which are PCI to SCSI I/O Processors. The LSI53C825AE is a minor modification of the existing LSI53C825A product. It has all the functionality of the LSI53C825A with the addition of features to enable it to comply with the Microsoft PC 97 Hardware Design Guide. Specifically, the LSI53C825AE has a Power Management Support enhancement. Because there are only slight differences between them, the LSI53C825A and LSI53C825AE are referred to as LSI53C825A throughout this technical manual. Only the new enhancements are referred to as LSI53C825AE.

This technical manual assumes the user is familiar with the current and proposed standards for SCSI and PCI. For additional background information on these topics, please refer to the list of reference materials provided in the [Preface](#) of this document.

The LSI53C825A PCI to SCSI I/O Processor brings high-performance I/O solutions to host adapter, workstation, and general computer designs, making it easy to add SCSI to any PCI system. It provides a local

memory bus for local storage of the device's BIOS ROM in flash memory or standard EPROMs. The LSI53C825A supports big and little endian byte addressing to accommodate a variety of data configurations. The LSI53C825A supports programming of local FLASH memory for updates to BIOS or SCRIPTS™ programs.

The LSI53C825A is a pin-for-pin replacement for the LSI53C825 PCI to SCSI I/O processor, although some software enhancements are needed to take advantage of the features in the LSI53C825A. The LSI53C825A performs fast 8-bit or 16-bit SCSI transfers in Single-Ended (SE) or differential mode, and improves performance by optimizing PCI bus utilization.

The LSI53C825A integrates a high-performance SCSI core, a PCI bus master DMA core, and the LSI Logic SCSI SCRIPTS processor to meet the flexibility requirements of SCSI-3 and future SCSI standards. It is designed to implement multithreaded I/O algorithms with a minimum of processor intervention, solving the protocol overhead problems of previous intelligent and nonintelligent adapter designs.

The LSI53C825A is fully supported by the LSI Logic Storage Device Management System (SDMS™), a software package that supports the Advanced SCSI Protocol Interface (ASPI) and the ANSI Common Access Method (CAM). SDMS provides BIOS and driver support for hard disk, tape, removable media products, and CD-ROM under the major PC operating systems.

---

## 1.2 Package and Feature Options

The LSI53C825A is packaged in a 160-pin plastic quad flat pack. The device is also available, as the LSI53C825AJ, with additional pins that support JTAG boundary scan testing. The JTAG boundary scan signals replace the TESTIN, MAC/\_TESTOUT, BIG\_LIT/, and SDIRP1 pins. The devices that have been upgraded to include the power management features are the LSI53C825AE and LSI53C825AJE.



## 1.2.1 PCI Pad Power-up Sequence

This power-up sequence should be followed when separate power supplies are being applied to the VDD-IO and VDD-CORE pins in a chip testing environment. Following this recommended power-up sequence helps prevent potential damage to these devices.

### 1.2.1.1 Description of the Issue

The Universal PCI pad input receiver in this cell library has all devices in a common N well attached to the 5 V core VDD supply. The P channel is powered from the VDD PCI supply.

In the event that the I/O VDD PCI supply goes high prior to the core VDD supply, the parasitic diode between the P channel source and the N well of the device can become forward biased. This creates an excessive current flow between the two nodes, and it causes damage to the device.

### 1.2.1.2 Solution for the Issue

In most system applications and production environments, the two VDD pins power-up simultaneously. The user should know of this potential hazard if using separate power supplies in a testing environment.

Either power-up the Core and I/O VDD PCI simultaneously, or if this is not possible, power-up the Core VDD before powering up the I/O VDD PCI supply.

Note that a power-down situation can have the same effect. The I/O must always power-down prior to the Core.

---

## 1.3 TolerANT<sup>®</sup> Technology

The LSI53C825A features TolerANT technology, which includes active negation on the SCSI drivers and input signal filtering on the SCSI receivers. Active negation drives the SCSI Request, Acknowledge, Data, and Parity signals HIGH rather than 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. TolerANT input signal filtering is a built-in feature of the LSI53C825A and all LSI Logic fast SCSI devices. On the LSI53C825A, the user may select a filtering period of 30 or 60 ns, with bit 1 in the [SCSI Test Two \(STEST2\)](#) register.

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. TolerANT technology is compatible with both the Alternative One and Alternative Two termination schemes proposed by the American National Standards Institute.

---

## 1.4 LSI53C825A Benefits Summary

This section provides an overview of the LSI53C825A features and benefits. It contains information on [SCSI Performance](#), [PCI Performance](#), [Integration](#), [Ease of Use](#), [Flexibility](#), [Reliability](#), and [Testability](#).

### 1.4.1 SCSI Performance

To improve SCSI performance, the LSI53C825A:

- Includes 4 Kbyte internal RAM for SCRIPTS instruction storage
- SCSI synchronous offset increased from 8 to 16 levels
- Supports variable block size and scatter/gather data transfers
- Performs sustained memory-to-memory DMA transfers faster than 47 Mbytes/s (@ 33 MHz)
- Minimizes SCSI I/O start latency
- Performs complex bus sequences without interrupts, including restore data pointers

- Reduces ISR overhead through a unique interrupt status reporting method
- Performs fast and wide SCSI bus transfers in SE and differential mode
  - 10 Mbytes/s asynchronous
- 20 Mbytes/s synchronous Load/Store SCRIPTS instruction increases performance of data transfers to and from chip registers
- Supports target disconnect and later reconnect with no interrupt to the system processor
- Supports multithreaded I/O algorithms in SCSI SCRIPTS with fast I/O context switching
- Expanded Register Move instruction supports additional arithmetic capability
- Complies with PCI Bus Power Management Specification (LSI53C825AE), Revision 1.0

## 1.4.2 PCI Performance

To improve PCI performance, the LSI53C825A:

- Complies with PCI 2.1 specification
- Bursts 2, 4, 8, 16, 32, 64, or 128 Dwords across 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

### 1.4.3 Integration

The LSI53C825A contains these integration features:

- 3.3 V/5 V PCI interface
- Full 32-bit PCI DMA bus master
- Can be used as a third-party PCI bus DMA controller by using Memory-to-Memory Move instructions
- High-performance SCSI core
- Integrated SCRIPTS processor

### 1.4.4 Ease of Use

The LSI53C825A provides ease of use by having:

- Up to one megabyte of add-in memory support for BIOS and SCRIPTS storage
- Direct PCI to SCSI connection
- Reduced SCSI development effort
- Easily adapted to the ASPI or the ANSI CAM, with SDMS software
- Compiler compatible with existing LSI53C7XX and LSI53C8XX family SCRIPTS
- Direct connection to PCI, SCSI SE, and differential buses
- Development tools and sample SCSI SCRIPTS available
- Maskable and pollable interrupts
- Wide SCSI, A or P cable, and up to 16 devices supported
- Three programmable SCSI timers: Select/Reselect, Handshake-to-Handshake, and General Purpose. The time-out period is programmable from 100  $\mu$ s to greater than 25.6 seconds
- SDMS software for complete PC-based operating system support
- Support for relative jumps
- SCSI Selected As ID bits (SSAID) for responding with multiple IDs

## 1.4.5 Flexibility

The LSI53C825A contains these flexibility features:

- High level programming interface (SCSI SCRIPTS)
- Programs local memory bus Flash memory
- Big/little endian support
- Selectable 88-byte or 536-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
- Selectable IRQ pins disable bit
- 32 additional scratch pad registers
- Ability to route system clock to SCSI clock

## 1.4.6 Reliability

The LSI53C825A contains these reliability features:

- 2 kV ESD protection on SCSI signals
- Typical 300 mV SCSI bus hysteresis
- Protection against bus reflections due to impedance mismatches
- Controlled bus assertion times (reduces RFI, improves reliability, and eases FCC certification)
- Latch-up protection greater than 150 mA
- Voltage feed-through protection (minimum leakage current through SCSI pads)
- 25% of pins are power and ground
- Power and ground isolation of I/O pads and internal chip logic

- TolerANT technology provides:
  - Active negation of SCSI Data, Parity, Request, and Acknowledge signals for improved Fast SCSI transfer rates
  - Input signal filtering on SCSI receivers improves data integrity, even in noisy cabling environments
- JTAG Boundary scan support (LSI53C825AJ only)

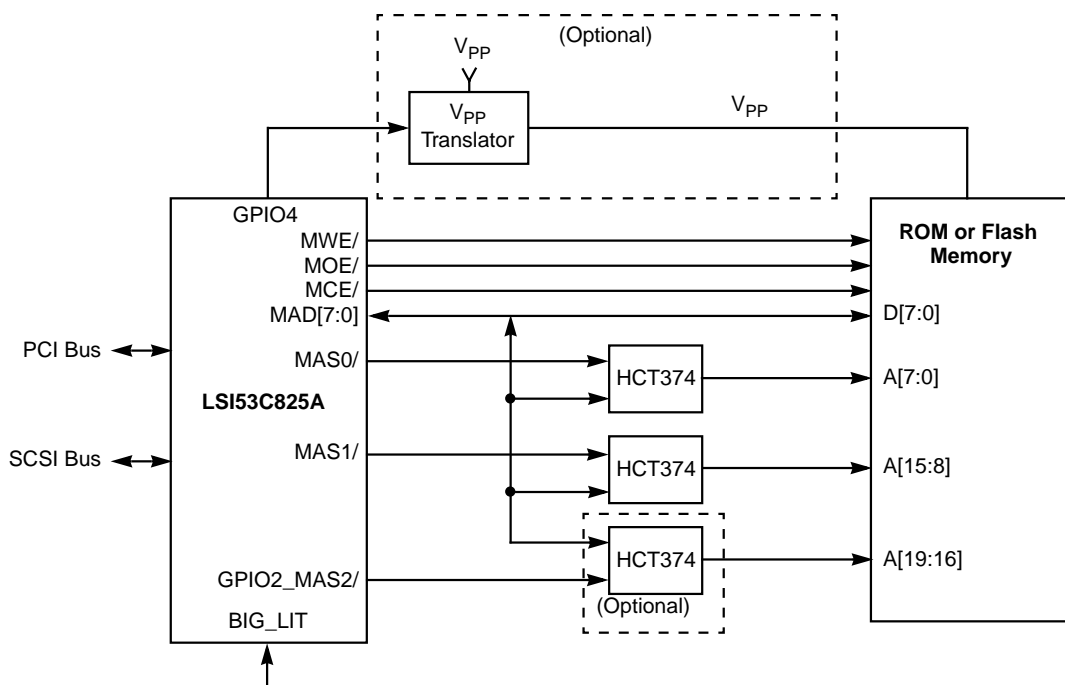
### 1.4.7 Testability

The LSI53C825A contains these testability features:

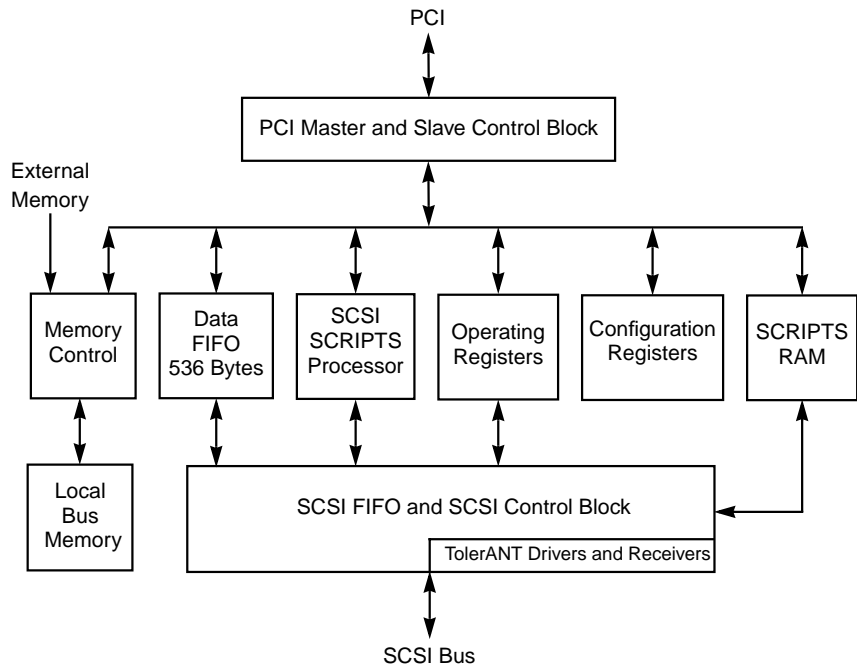
- All SCSI signals accessible through programmed I/O
- SCSI loopback diagnostics
- SCSI bus signal continuity checking
- Support for single-step mode operation
- Test mode (AND tree) to check pin continuity to the board (most package options)
- JTAG Boundary scan support (LSI53C825AJ only)

A system diagram showing the connections of the LSI53C825A with an external ROM or Flash memory is pictured in [Figure 1.1](#). A block diagram of the LSI53C825A is pictured in [Figure 1.2](#).

**Figure 1.1 LSI53C825A External Memory Interface**



**Figure 1.2 LSI53C825A Chip Block Diagram**





# Chapter 2

## Functional Description

---

Chapter 2 is divided into the following sections:

- [Section 2.1, “PCI Addressing”](#)
- [Section 2.2, “SCSI Functional Description”](#)
- [Section 2.3, “External Memory Interface”](#)
- [Section 2.4, “PCI Cache Mode”](#)
- [Section 2.5, “Power Management”](#)

---

## 2.1 PCI Addressing

There are three physical PCI-defined address spaces:

- [PCI Configuration Space](#)
- [I/O Space](#)
- [Memory Space](#)

### 2.1.1 Configuration Space

Configuration space is a contiguous 256 x 8-bit set of addresses dedicated to each “slot” or “stub” on the bus. Decoding C\_BE/[3:0] determines if a PCI cycle is intended to access 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 addresses are used to select a specific 8-bit register. AD[10:8] are decoded as well, but they must be zero or the LSI53C825A does not respond. According to the PCI specification, AD[10:8] are to be used for multifunction devices. The host processor uses the PCI configuration space to initialize the LSI53C825A.

The lower 128 bytes of the LSI53C825A configuration space hold system parameters while the upper 128 bytes map into the LSI53C825A operating registers. For all PCI cycles except configuration cycles, the LSI53C825A registers are located on the 256-byte block boundary defined by the base address assigned through the configured register. The LSI53C825A operating registers are available in both the upper and lower 128-byte portions of the 256-byte space selected.

At initialization time, each PCI device is assigned a base address (in the case of the LSI53C825A, the upper 24 bits of the address are selected) for memory accesses and I/O accesses. On every access, the LSI53C825A 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 LSI53C825A and the low-order eight bits define the register to be accessed. A decode of C\_BE/[3:0] determines which registers and what type of access is to be performed.

**I/O Space** – PCI defines memory space as a contiguous 32-bit memory address that is shared by all system resources, including the LSI53C825A. [Base Address One \(Memory\)](#) determines which 256-byte memory area this device will occupy.

**Memory Space** – PCI defines I/O space as a contiguous 32-bit I/O address that is shared by all system resources, including the LSI53C825A. [Base Address Zero \(I/O\)](#) determines which 256-byte I/O area this device will occupy.

## 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 command encoding and types appear in [Table 2.1](#).

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

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

1. This operation is selectable by bit 2 in the [DMA Mode \(DMODE\)](#) operating register.

2. This operation is selectable by bit 3 in the [DMA Mode \(DMODE\)](#) operating register.

3. This operation is selectable by bit 0 in the [Chip Test Three \(CTEST3\)](#) operating register.

### 2.1.2.1 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.

### 2.1.2.2 I/O Write Command

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

### **2.1.2.3 Memory Read Command**

The Memory Read command reads data from an agent mapped in memory address space. All 32 address bits are decoded.

### **2.1.2.4 Memory Write Command**

The Memory Write command writes data to an agent when mapped in memory address space. All 32 address bits are decoded.

### **2.1.2.5 Memory Read Multiple Command**

The Memory Read Multiple command reads data from an agent mapped in memory address space. All 32 address bits are decoded.

### **2.1.2.6 Memory Read Line Command**

The Memory Read Line command reads data from an agent mapped in memory address space. All 32 address bits are decoded.

### **2.1.2.7 Memory Write and Invalidate Command**

The Memory Write and Invalidate command writes data to an agent when mapped in memory address space. All 32 address bits are decoded.

## **2.1.3 PCI Cache Mode**

The LSI53C825A supports the PCI specification for an 8-bit [Cache Line Size](#) register located in 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 Read Line, Read Multiple, and Write and Invalidate are each software enabled or disabled to allow the user full flexibility in using these commands.

### **2.1.3.1 Support for PCI Cache Line Size Register**

The LSI53C825A supports the PCI specification for an 8-bit [Cache Line Size](#) register in PCI configuration space. It can sense and react to nonaligned addresses corresponding to cache line boundaries.

### 2.1.3.2 Selection of Cache Line Size

The cache logic selects a cache line size based on the values for the burst size in the [DMA Mode \(DMODE\)](#) register, bit 2 in the [Chip Test Five \(CTEST5\)](#) register, and the PCI [Cache Line Size](#) register.

Note: The LSI53C825A does not automatically use the value in the PCI [Cache Line Size](#) register as the cache line size value. The chip scales the value of the [Cache Line Size](#) register down to the nearest binary burst size allowed by the chip (2, 4, 8, 16, 32, 64, or 128), compares this value to the burst size defined by the values of the [DMA Mode \(DMODE\)](#) register and bit 2 of the [Chip Test Five \(CTEST5\)](#) register, then selects the smallest as the value for the cache line size. The LSI53C825A uses this value for all burst data transfers.

### 2.1.3.3 Alignment

The LSI53C825A uses the calculated line size value to monitor the current address for alignment to the cache line size. When it is not aligned, the chip attempts to align to the cache boundary by using a “smart aligning” scheme. This means that it attempts to use the largest burst size possible that is less than the cache line size, to reach the cache boundary quickly with no overflow. This process is a stepping mechanism that steps up to the highest possible burst size based on the current address.

The stepping process begins at a 4 Dword boundary. The LSI53C825A will first try to align to a 4 Dword boundary (0x00, 0x010, 0x020, etc.) by using single Dword transfers (no bursting). Once this boundary is reached the chip evaluates the current alignment to various burst sizes allowed, and selects the largest possible as the next burst size, while not exceeding the cache line size. The chip then issues this burst, and reevaluates the alignment to various burst sizes, again selecting the largest possible while not exceeding the cache line size, as the next burst size. This stepping process continues until the chip reaches the cache line size boundary or runs out of data. Once a cache line boundary is reached, the chip uses the cache line size as the burst size from then on, except in the case of multiples (explained below). The alignment process is finished at this point.

**Example: Cache Line Size - 16, Current Address = 0x01** – The chip is not aligned to a 4 Dword cache boundary (the stepping threshold), so it issues four single Dword transfers (the first is a 3-byte transfer). At address 0x10, the chip is aligned to a 4 Dword boundary, but not aligned to any higher burst size boundaries that are less than the cache line size. So, the part issues a burst of 4. At this point, the address is 0x20, and the chip evaluates that it is aligned not only to a 4 Dword boundary, but also to an 8 Dword boundary. It selects the highest, 8, and burst 8 Dwords. At this point, the address is 0x40, which is a cache line size boundary. Alignment stops, and the burst size from then on is switched to 16.

#### 2.1.3.4 Memory Move Misalignment

The LSI53C825A does not operate in a cache alignment mode when a Memory Move instruction type is issued and the read and write addresses are different distances from the nearest cache line boundary. For example, if the read address is 0x21F and the write address is 0x42F, and the cache line size is 8, the addresses are byte aligned, but they are not the same distance from the nearest cache boundary. The read address is 1 byte from the cache boundary 0x220 and the write address is 17 bytes from the cache boundary 0x440. In this situation, the chip does not align to cache boundaries and operates as a LSI53C825.

#### 2.1.3.5 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 LSI53C825A enables Memory Write and Invalidate cycles when bit 0 in the [Chip Test Three \(CTEST3\)](#) register (WRIE) and bit 4 in the PCI [Command](#) register are set. When the following conditions are met, Memory Write and Invalidate commands are issued:

- The CLSE bit, WRIE bit, and PCI configuration [Command](#) register, bit 4 are set.

- The **Cache Line Size** register contains a legal burst size 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 chip has enough bytes in the **DMA FIFO (DFIFO)** to complete at least one full cache line burst.
- The chip is aligned to a cache line boundary.

When these conditions are met, the LSI53C825A issues a Write and Invalidate command instead of a Memory Write command during all PCI write cycles.

**Multiple Cache Line Transfers** – The 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 the Revision 2.1 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 being that 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, 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 reevaluates the burst size based on the amount of remaining data to transfer and again selects the highest possible multiple of the cache line size, no larger than the DMODE burst size. The most likely scenario of this scheme is that the chip selects the DMODE burst size after alignment, and issue 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 Write and Invalidate command such that when a latency time-out occurs, the LSI53C825A continues to transfer up until 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 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 Write and Invalidate command on the next ownership, in accordance with the PCI specification.

**PCI Target Disconnect** – During a Write and Invalidate transfer, if the target device issues a disconnect the LSI53C825A relinquishes the bus and immediately tries to finish the transfer on another bus ownership. The chip does not issue another Write and Invalidate command on the next ownership unless the address is aligned.

### 2.1.3.6 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 up to a cache line boundary rather than a single memory cycle. The Read Line Mode function that exists in the previous LSI53C8XX chips has been modified in the LSI53C825A to reflect the PCI [Cache Line Size](#) register specifications. The functionality of the Enable Read Line bit (bit 3 in [DMA Mode \(DMODE\)](#)) has been modified to more resemble the Write and Invalidate mode in terms of conditions that must be met before a Read Line command is issued. However, the Read Line option operates exactly like the previous LSI53C8XX chips when cache mode is disabled by a CLSE bit reset or when certain conditions exist in the chip (explained below).

The Read Line mode is enabled by setting bit 3 in the [DMA Mode \(DMODE\)](#) register. If cache mode is disabled, Read Line commands are issued on every read data transfer, except opcode fetches, as in previous LSI53C8XX chips.

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

- The CLSE and Enable Read Line bits are set.
- The [Cache Line Size](#) register contains a legal burst size value in Dwords (2, 4, 8, 16, 32, 64, or 128) that value is less than or equal to the DMODE burst size.



- The number of bytes to be transferred at the time a cache boundary is reached is equal to or greater than the DMODE burst size.
- The chip is aligned to 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.

### 2.1.3.7 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 LSI53C825A supports PCI Read Multiple functionality and issues Read Multiple commands on the PCI bus when the Read Multiple Mode is enabled. This mode is enabled by setting bit 2 of the [DMA Mode \(DMODE\)](#) register (ERMP). If cache mode is enabled, a Read Multiple command is issued on all read cycles, except opcode fetches, when the following conditions are met:

- The CLSE and ERMP bits are set.
- 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 DMODE burst size.
- The number of bytes to be transferred at the time a cache line boundary has been reached must be at least twice the full cache line size.
- The chip is aligned to a cache line boundary.

When these conditions are met, the chip issues a 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 be read is a multiple of the cache line size as allowed for in the Revision 2.1 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 DMODE burst size bits and [Chip Test Five \(CTEST5\)](#), bit 2.

### 2.1.3.8 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.

### 2.1.3.9 Unsupported PCI Commands

The LSI53C825A does not respond to reserved commands, special cycle, dual address cycle, or interrupt acknowledge commands as a slave. It never generates these commands as a master.

---

## 2.2 SCSI Functional Description

The LSI53C825A is composed of three functional blocks: the [SCSI Core](#), the [DMA Core](#), and the [SCRIPTS Processor](#). The LSI53C825A is fully supported by the SDMS, a complete software package that supports the LSI Logic product line of SCSI processors and controllers. The PCI Bus [Power Management](#) support (LSI53C825AE) is discussed at the end of this chapter.

### 2.2.1 SCSI Core

The SCSI core supports the 8-bit or 16-bit data bus. It supports SCSI synchronous transfer rates up to 20 Mbytes/s, and asynchronous transfer rates up to 10 Mbytes/s on a 16-bit wide 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 SCSI-3 requirements.

The SCSI core offers low level register access or a high level control interface. Like first generation SCSI devices, the LSI53C825A SCSI core can be accessed as a register oriented device. The ability to sample and/or assert any signal on the SCSI bus can be used in error recovery and diagnostic procedures. In support of loopback diagnostics, the SCSI core may perform a self-selection and operate as both an initiator and a target.

The LSI53C825A SCSI core 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.

## **2.2.2 DMA Core**

The DMA core is a bus master DMA device that attaches directly to the industry standard PCI Bus. The DMA core is tightly coupled to the SCSI core through the SCRIPTS processor, which supports uninterrupted scatter/gather memory operations.

The LSI53C825A supports 32-bit memory and automatically supports misaligned DMA transfers. A 536-byte FIFO allows the LSI53C825A to support 2, 4, 8, 16, 32, 64, or 128 longword bursts across the PCI bus interface.

## **2.2.3 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 and are executed from 32-bit system RAM. The SCRIPTS processor executes complex SCSI bus sequences independently of the host CPU.

The SCRIPTS processor can begin a SCSI I/O operation in approximately 500 ns. This compares with 2–8 ms required for traditional intelligent host adapters. 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.

## **2.2.4 Internal SCRIPTS RAM**

The LSI53C825A has 4 Kbytes (1024 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 LSI53C825A use the PCI bus, as if they were external accesses. The MAD5 pin enables the 4 K internal RAM. To disable the internal RAM, connect a 4.7 k $\Omega$  resistor between the MAD5 pin and V<sub>SS</sub>.

The RAM can be relocated by the PCI system BIOS anywhere in 32-bit address space. The RAM Base Address register in PCI configuration space contains the base address of the internal RAM. This register is similar to the ROM Base Address register in PCI configuration space. To simplify loading of SCRIPTS instructions, the base address of the RAM will appear 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 (i.e., by the CPU) follow the same timing sequence as a standard slave register access, except that the target wait states required 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 LSI53C825A, see [Chapter 5, “SCSI SCRIPTS Instruction Set.”](#)

## 2.2.5 SDMS: The Total SCSI Solution

For users who do not need to develop custom drivers, LSI Logic provides a total SCSI solution in PC environments with the SDMS. SDMS software provides BIOS driver support for hard disk, tape, and removable media peripherals for the major PC-based operating systems.

SDMS software includes a SCSI BIOS to manage all SCSI functions related to the device. It also provides a series of SCSI device drivers that support most major operating systems. SDMS software supports a multithreaded I/O application programming interface (API) for user developed SCSI applications. SDMS software supports both the ASPI and CAM SCSI software specifications.

## 2.2.6 Prefetching SCRIPTS Instructions

When enabled (by setting the Prefetch Enable bit in the [DMA Control \(DCNTL\)](#) register), the prefetch logic in the LSI53C825A 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 4 Dwords, it disables itself. While the LSI53C825A is prefetching SCRIPTS instructions, the PCI [Cache Line Size](#) register value does not have any effect and the Read Line, Read Multiple, and Write and Invalidate commands are not used.

The LSI53C825A 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 software. When one of these conditions apply, the contents of the prefetch unit are flushed automatically.

- 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 [Section 5.7, “Memory Move Instructions,”](#) in [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 DSP.
- On all Transfer Control instructions when the transfer conditions are met. This is necessary because the next instruction to be executed is not the sequential next instruction in the prefetch unit.
- When the Prefetch Flush bit ([DMA Control \(DCNTL\)](#), 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 in the [DMA Mode \(DMODE\)](#) register (0x38) causes the LSI53C825A to burst in the first two longwords of all instruction fetches. If the instruction is a memory-to-memory move, the third longword is accessed in a separate ownership. If the instruction is an indirect type, the additional longword 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 longwords required, in two bursts of two longwords each.

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

---

## 2.3 External Memory Interface

The LSI53C825A 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. The device also supports Flash ROM updates through the add-in interface and the GPIO4 pin (used to control  $V_{PP}$  the power supply for programming external memory). 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 LSI53C825A, two or three external 8-bit address holding registers (HCT273 or HCT374), and the appropriate memory device. The 4.7 k $\Omega$  pull-down 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), an MTD4P05, and several passive components are also needed. The memory size and speed is determined by pull-down resistors on the 8-bit bidirectional memory bus at power-up. The LSI53C825A senses this bus shortly after the release of the Reset signal and configures the 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 LSI53C825A 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 LSI53C825A. [Table 2.2](#) shows the memory space associated with the possible values of MAD[3:1]. The MAD[3:1] pins are fully defined in [Chapter 4, “Registers.”](#)

**Table 2.2 External Memory 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 k $\Omega$  pull-down resistors on the MAD pins corresponding to the available memory space. For example, to connect to a 32 Kbyte external ROM, use pull-downs on MAD3 and MAD2. If the external memory interface is not used, then no external resistors are necessary since there are internal pull-ups on the MAD bus. The internal pull-up resistors are disabled when external pull-down resistors are detected, to reduce current drain.

The LSI53C825A allows the system to determine the size of the available external memory using the [Expansion ROM Base Address](#) register in 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 3, "Signal Descriptions."](#)

MAD0 is the slow ROM pin. When pulled down, 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. The 12 V power supply for flash memory,  $V_{PP}$ , is enabled and disabled with the GPIO4 pin and the GPIO4 control bit. For more information on the GPIO4 pin, refer to [Chapter 4, "Registers."](#)



---

## 2.4 PCI Cache Mode

The LSI53C825A supports the PCI specification for an 8-bit [Cache Line Size](#) register located in 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 Read Line, Read Multiple, and Write and Invalidate are each software enabled or disabled to allow the user full flexibility in using these commands. For more information on PCI cache mode operations, refer to [Chapter 3, “Signal Descriptions.”](#)

### 2.4.1 Load and Store Instructions

The LSI53C825A supports the Load and Store instruction type, which simplifies the movement of data between memory and the internal chip registers. It also enables the LSI53C825A to transfer bytes to addresses relative to the [Data Structure Address \(DSA\)](#) register. For more information on the Load and Store instructions, refer to [Chapter 5, “SCSI SCRIPTS Instruction Set.”](#)

### 2.4.2 3.3 V/5 V PCI Interface

The LSI53C825A can attach directly to a 3.3 V or a 5 V PCI interface, due to separate  $V_{DD}$  pins for the PCI bus drivers. This allows the devices to be used on the universal board recommended by the PCI Special Interest Group.

### 2.4.3 Additional Access to General Purpose Pins

The LSI53C825A can access the GPIO0 and GPIO1 general purpose pins through register bits in the PCI configuration space, instead of using the [General Purpose Pin Control \(GPCNTL\)](#) register in the operating register space to control these pins. In the LSI Logic SDMS software, the configuration bits control pins as the clock and data lines, respectively.

To access the GPIO[1:0] pins through the configuration space, connect a 4.7 k $\Omega$  resistor between the MAD7 pin and  $V_{SS}$ . MAD7 contains an internal pull-up that is sensed shortly after chip reset. If the pin is sensed high, GPIO[1:0] access is disabled; if it is low, GPIO[1:0] access is enabled. Additionally, if GPIO[1:0] access has been enabled through the

MAD7 pin and if GPIO0 and/or GPIO1 are sensed low after chip reset, GPIO[1:0] access is disabled. If GPIO[1:0] access through configuration space is enabled, the GPIO0 and GPIO1 pins cannot be controlled from the [General Purpose Pin Control \(GPCNTL\)](#) and [General Purpose \(GPREG\)](#) registers, but are observable from the [General Purpose \(GPREG\)](#) register. When GPIO[1:0] access is enabled, the Serial Interface Control register at configuration addresses 0x34–0x35 controls the GPIO0 and GPIO1 pins. For more information on GPIO[1:0] access, refer to the Serial Interface Control register description in [Chapter 3, “Signal Descriptions.”](#) For more information on the GPIO pins, see [Chapter 4, “Registers.”](#) This does not apply to the LSI53C825AE.

Note: The LSI Logic SDMS software controls the GPIO0 and GPIO1 pins using the [General Purpose Pin Control \(GPCNTL\)](#) and [General Purpose \(GPREG\)](#) registers. Therefore, if using SDMS, do not connect a 4.7 k $\Omega$  resistor between MAD7 and  $V_{SS}$ .

## 2.4.4 JTAG Boundary Scan Testing

The LSI53C825AJ includes support for JTAG boundary scan testing in accordance with the IEEE 1149.1 specification, with one exception that is discussed in this section. The device can accept all required boundary scan instructions, as well as the optional CLAMP, HIGH-Z, and IDCODE instructions.

The LSI53C825AJ 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. The 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. This action cannot be controlled by the boundary scan logic, and thus is not compliant with the specification. There are two solutions that resolve this issue:

- 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.

- When RST/ is asserted during boundary scan testing, the expected output on the SCSI pins must be HIGH-Z condition, and not what is contained in the boundary scan data registers for the SCSI pin output cells.

Because of package limitations, the LSI53C825AJ replaces the TESTIN, MAC/\_TESTOUT, BIG\_LIT, and SDIRP1 signals with the JTAG boundary scan signals.

## 2.4.5 Big and Little Endian Support

The LSI53C825A supports both big and little endian byte ordering through pin selection. The LSI53C825AJ operates in little endian mode only (the BIG\_LIT pin is replaced by one of the JTAG boundary scan signals). In big endian mode, the first byte of an aligned SCSI to PCI transfer is routed to lane three and succeeding transfers are routed to descending lanes. This mode of operation also applies to data transfers over the add-in ROM interface. The byte of data accessed at location 0x0000 from memory is routed to lane three, and the data at location 0x0003 is routed to byte lane 0. In little endian mode, the first byte of an aligned SCSI to PCI transfer is routed to lane zero and succeeding transfers are routed to ascending lanes. This mode of operation also applies to the add-in ROM interface. The byte of data accessed at location 0x0000 from memory is routed to lane zero, and the data at location 0x0003 is routed to byte lane 3.

The Big\_Lit pin gives the LSI53C825A the flexibility of operating with either big or little endian byte orientation. Internally, in either mode, the actual byte lanes of the DMA FIFO and registers are not modified. The LSI53C825A supports slave accesses in big or little endian mode.

When a Dword is accessed, no repositioning of the individual bytes is necessary since Dwords are addressed by the address of the least significant byte. SCRIPTS always uses Dwords in 32-bit systems, so compatibility is maintained between systems using different byte orientations. When less than a Dword is accessed, individual bytes must be repositioned. Internally, the LSI53C825A adjusts the byte control logic of the DMA FIFO and register decodes to access the appropriate byte lanes. The registers always appear on the same byte lane, but the address of the register are repositioned.

Big/little endian mode selection has the most effect on individual byte access. Internally, the LSI53C825A adjusts the byte control logic of the DMA FIFO and register decodes to enable the appropriate byte lane. The registers always appear on the same byte lane, but the address of the register are repositioned.

Data to be transferred between system memory and the SCSI bus always starts at address zero and continues through address 'n' – there is no byte ordering in the chip. The first byte in from the SCSI bus goes to address 0, the second to address 1, etc. Going out onto the SCSI bus, address zero is the first byte out on the SCSI bus, address 1 is the second byte, etc. The only difference is that in a little endian system, address 0 is on byte lane 0, and in big endian mode address zero is on byte lane 3.

Correct SCRIPTS are generated if the SCRIPTS compiler is run on a system that has the same byte ordering as the target system. Any SCRIPTS patching in memory must patch the instruction with the byte ordering that the SCRIPTS processor expects.

Software drivers for the LSI53C825A should access registers by their logical name (i.e., SCNTL0) rather than by their address. The logical name should be equated to the register's big endian address in big endian mode (SCNTL0 = 0x03), and its little endian address in little endian mode (SCNTL0 = 0x00). This way, there is no change to the software when moving from one mode to the other; only the equate statement setting the operating modes needs to be changed.

Addressing of registers from within a SCRIPTS instruction is independent of bus mode. Internally, the LSI53C825A always operates in little endian mode.

## 2.4.6 Loopback Mode

The LSI53C825A 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 [Chip Test One \(CTEST1\)](#) register, the LSI53C825A allows control of all SCSI signals, whether the LSI53C825A is operating in initiator or target mode. For more information on this mode of operation, refer to the *SCSI SCRIPTS Processors Programming Guide*.

## 2.4.7 Parity Options

The LSI53C825A 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 Zero \(SCNTL0\)](#) register. [Table 2.5](#) describes the options available when a parity error occurs.

**Table 2.3 Bits Used for Parity Control and Generation**

Bit Name	Location	Description
Assert SATN/ on Parity Errors	<a href="#">SCSI Control Zero (SCNTL0)</a> , Bit 1	Causes the LSI53C825A to automatically assert SATN/ when it detects a parity error while operating as an initiator.
Enable Parity Checking	<a href="#">SCSI Control Zero (SCNTL0)</a> , Bit 3	Enables the LSI53C825A to check for parity errors. The LSI53C825A checks for odd parity.
Assert Even SCSI Parity	<a href="#">SCSI Control One (SCNTL1)</a> , Bit 2	Determines the SCSI parity sense generated by the LSI53C825A to the SCSI bus.
Disable Halt on SATN/ or a Parity Error (Target Mode Only)	<a href="#">SCSI Control One (SCNTL1)</a> , Bit 5	Causes the LSI53C825A not to halt operations when a parity error is detected in target mode.
Enable Parity Error Interrupt	<a href="#">SCSI Interrupt Enable Zero (SIEN0)</a> , Bit 0	Determines whether the LSI53C825A will generate an interrupt when it detects a SCSI parity error.
Parity Error	<a href="#">SCSI Interrupt Status Zero (SIST0)</a> , Bit 0	This status bit is set whenever the LSI53C825A has detected a parity error on the SCSI bus.

**Table 2.3 Bits Used for Parity Control and Generation (Cont.)**

Bit Name	Location	Description
Status of SCSI Parity Signal	<a href="#">SCSI Status Zero (SSTAT0)</a> , Bit 0	This status bit represents the active HIGH current state of the SCSI SDP0 parity signal.
SCSI SDP1 Signal	<a href="#">SCSI Status Two (SSTAT2)</a> , Bit 0	This bit represents the active HIGH current state of the SCSI SDP1 parity signal.
Latched SCSI Parity	<a href="#">SCSI Status Two (SSTAT2)</a> , Bit 3 and <a href="#">SCSI Status One (SSTAT1)</a> , Bit 3	These bits reflect the SCSI odd parity signal corresponding to the data latched into the <a href="#">SCSI Input Data Latch (SIDL)</a> register.
Master Parity Error Enable	<a href="#">Chip Test Four (CTEST4)</a> , Bit 3	Enables parity checking during master data phases.
Master Data Parity Error	<a href="#">DMA Status (DSTAT)</a> , Bit 6	Set when the LSI53C825A as a master detects that a target device has signaled a parity error during a data phase.
Master Data Parity Error Interrupt Enable	<a href="#">DMA Interrupt Enable (DIEN)</a> , Bit 6	By clearing this bit, a Master Data Parity Error will not cause IRQ/ to be asserted, but the status bit will be set in the <a href="#">DMA Status (DSTAT)</a> register.

**Table 2.4 SCSI Parity Control**

EPC	AESP	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. Key: EPC = Enable Parity Checking (bit 3, <a href="#">SCSI Control Zero (SCNTL0)</a> ). ASEP = Assert SCSI Even Parity (bit 2, <a href="#">SCSI Control One (SCNTL1)</a> ). 2. This table only applies when the Enable Parity Checking bit is set.		

**Table 2.5 SCSI Parity Errors and Interrupts**

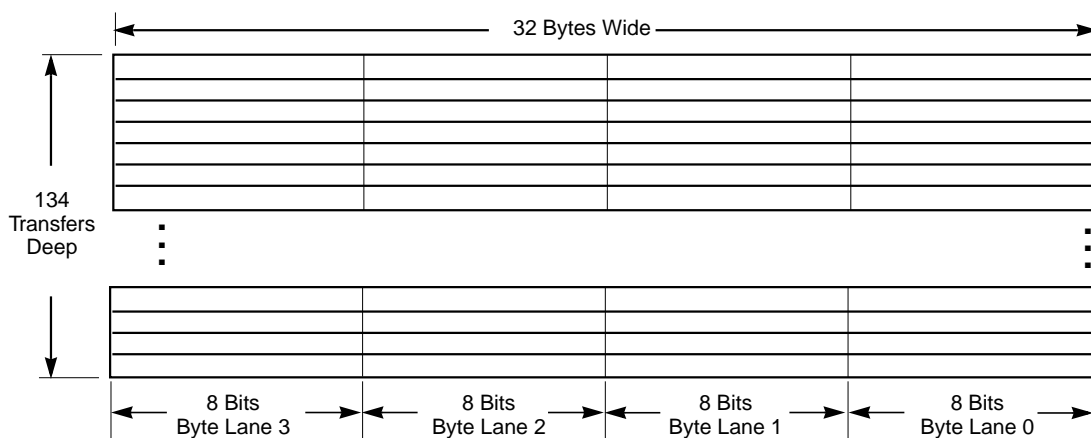
DPH	PAR	Description
0	0	Halts when a parity error occurs in target or initiator mode and will not generate an interrupt.
0	1	Halts when a parity error occurs in target mode and will generate an interrupt in target or initiator mode.
1	0	Does not halt in target mode when a parity error occurs until the end of the transfer. An interrupt will not be generated.
1	1	Does not halt in target mode when a parity error occurs until the end of the transfer. An interrupt will be generated.

Key:  
DHP = Disable Halt on SATN/ or Parity Error (bit 5, [SCSI Control One \(SCNTL1\)](#).  
PAR = Parity Error (bit 0, [SCSI Interrupt Enable Zero \(SIEN0\)](#)).

## 2.4.8 DMA FIFO

The DMA FIFO is 4 bytes wide by 134 transfers deep. The DMA FIFO is illustrated in [Figure 2.1](#). To assure compatibility with older products in the LSI53C8XX family, the user may set the DMA FIFO size to 88 bytes by clearing the DMA FIFO Size bit, bit 5 in the [Chip Test Five \(CTEST5\)](#) register.

**Figure 2.1 DMA FIFO Sections**



### 2.4.8.1 Data Paths

The data path through the LSI53C825A is dependent on whether data is being moved into or out of the chip, and whether SCSI data is being transferred asynchronously or synchronously.

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 88 bytes, 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 [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 88.

If the DMA FIFO size is set to 536 bytes (using bit 5 of the [Chip Test Five \(CTEST5\)](#) register), 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 [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 536.

- 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, then the least significant byte or the most significant byte in the [SCSI Output Data Latch \(SODL\)](#) register is full, respectively. Checking this bit also reveals bytes left in the [SCSI Output Data Latch \(SODL\)](#) register from a Chained Move operation with an odd byte count.

#### Synchronous SCSI Send –

- Step 1. If the DMA FIFO size is set to 88 bytes, 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 [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 88.

If the DMA FIFO size is set to 536 bytes (using bit 5 of the Chip Test Five (CTEST5) register), 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 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 536.

- 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, then the least significant byte or the most significant byte in the SCSI Output Data Latch (SODL) register is full, respectively. Checking this bit also reveals bytes left in the SCSI Output Data Latch (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. If bit 6 is set in the SCSI Status Zero (SSTAT0) or SCSI Status Two (SSTAT2), 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 88 bytes, 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 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 88.

If the DMA FIFO size is set to 536 bytes (using bit 5 of the Chip Test Five (CTEST5) register), 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 [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 536.

- Step 2. Read bit 7 in the [SCSI Status Zero \(SSTAT0\)](#) and [SCSI Status Two \(SSTAT2\)](#) register to determine if any bytes are left in the [SCSI Input Data Latch \(SIDL\)](#) register. If bit 7 is set in the [SCSI Status Zero \(SSTAT0\)](#) or [SCSI Status Two \(SSTAT2\)](#), 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 Control Two \(SCNTL2\)](#), bit 0) to determine whether a byte is left in the [SCSI Wide Residue \(SWIDE\)](#) register.

### **Synchronous SCSI Receive –**

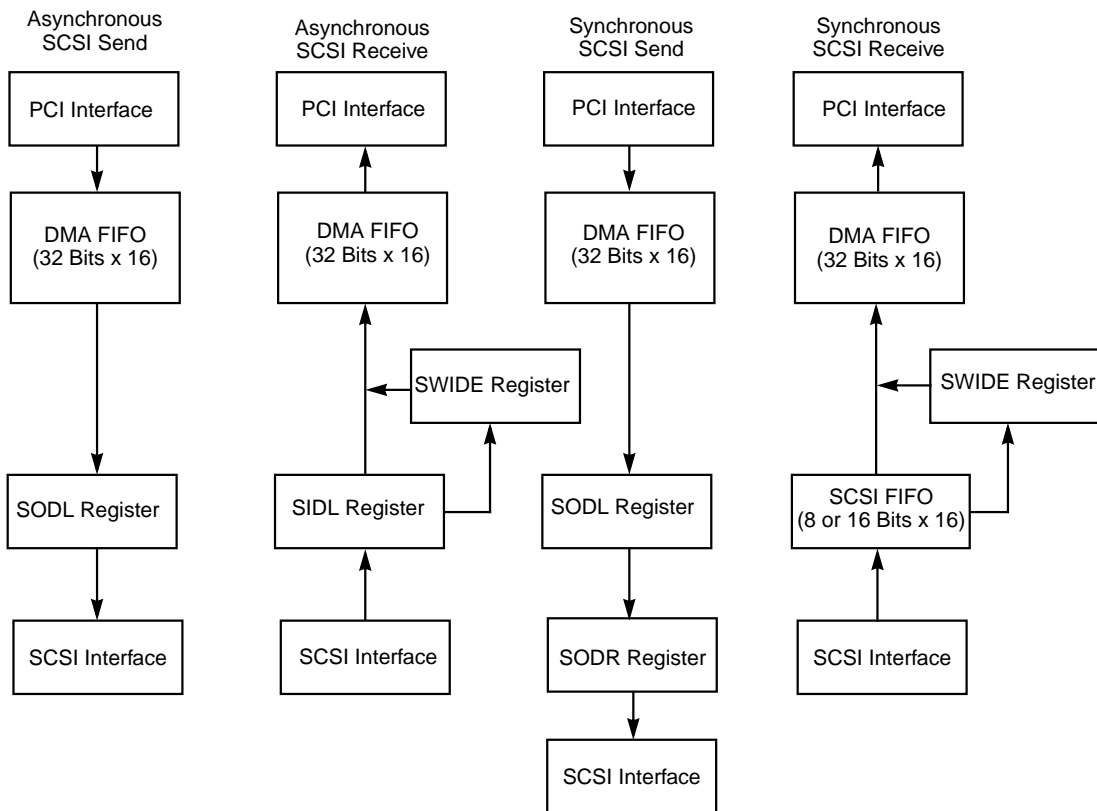
- Step 1. If the DMA FIFO size is set to 88 bytes, subtract the seven least significant bits of the [DMA Byte Counter \(DBC\)](#) register from the 7-bit value of the [DMA FIFO \(DFIFO\)](#) register. AND the result with 0x7F for a byte count between zero and 88.

If the DMA FIFO size is set to 536 bytes (using bit 5 of the [Chip Test Five \(CTEST5\)](#) register), 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 [Chip Test Five \(CTEST5\)](#) register and bits [7:0] of the [DMA FIFO \(DFIFO\)](#) register. AND the result with 0x3FF for a byte count between zero and 536.

- Step 2. Read bits [7:4] of the [SCSI Status One \(SSTAT1\)](#) register and bit 4 of the [SCSI Status Two \(SSTAT2\)](#) register, 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.

[Figure 2.2](#) shows how data is moved to/from the SCSI bus in each of the different modes.

**Figure 2.2 LSI53C825A Host Interface Data Paths**



## 2.4.9 SCSI Bus Interface

The LSI53C825A supports both SE and differential operation.

All SCSI signals are active LOW. The LSI53C825A contains the SE output drivers and can be connected directly to the SCSI bus. Each output is isolated from the power supply to ensure that a powered-down LSI53C825A has no effect on an active SCSI bus (CMOS “voltage feed-through” phenomena). TolerANT technology provides signal filtering at the inputs of SREQ/ and SACK/ to increase immunity to signal reflections.

### 2.4.9.1 Differential Mode

In differential mode, the SDIR[15:0], SDIRP[1:0], IGS, TGS, RSTDIR, BSYDIR, and SELDIR signals control the direction of external differential pair transceivers. The LSI53C825A is placed in differential 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 following signals defined in [Table 2.6](#) are used during differential operation by the LSI53C825A.

**Table 2.6 Differential Mode**

Signal	Function
BSYDIR, SELDIR, RSTDIR	Active HIGH signals used to enable the differential drivers as outputs for SCSI signals BSY/, SEL/, and RST/, respectively.
SDIR[15:0], SDIRP[1:0]	Active HIGH signals used to control direction of the differential drivers for SCSI data and parity lines, respectively.
IGS	Active HIGH signal used to control direction of the differential driver for initiator group signals ATN/ and ACK/.
TGS	Active HIGH signal used to control direction of the differential drivers for target group signals MSG/, C/D/, I/O/, and REQ/.
DIFFSENS	Input to the LSI53C825A used to detect the presence of a SE device on a differential system. If a logical zero is detected on this pin, then it is assumed that an SE device is on the bus and all SCSI outputs will be 3-stated to avoid damage to the transceiver.

See [Figure 2.3](#) for an example differential wiring diagram, in which the LSI53C825A is connected to the TI 75LBC976 differential transceiver. 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  $\Omega$  when the Active Negation portion of LSI Logic TolerANT technology is not enabled. When TolerANT technology is enabled, the recommended resistor value on the REQ/, ACK/, SD[7:0]/, and SDP0/ signals is 1.5 k $\Omega$ . The electrical characteristics of these pins change when TolerANT is enabled, permitting a higher resistor value.

To interface the LSI53C825A to the 75LBC976, connect the DIR pins, as well as IGS and TGS, of the LSI53C825A directly to the transceiver enables (nDE/RE/). These signals control the direction of the channels on the 75LBC976.

The SCSI bidirectional control and data pins (SD[7:0]/, SDP0/, REQ/, ACK/, MSG/, I\_O/, C\_D/, and ATN/) of the LSI53C825A connect to the bidirectional data pins (nA) of the 75LBC976 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, SEL/, BSY/, and RST/ are connected to the 75LBC976 with a pull-down resistor. The pull-down resistors are required when the pins (nA) of the 75LBC976 are configured as inputs. When the data pins are inputs, the resistors provide a bias voltage to both the LSI53C825A pins (SEL/, BSY/, and RST/) and the 75LBC976 data pins. Because the SEL/, BSY/, and RST/ pins on the LSI53C825A are inputs only, this configuration allows for the SEL/, BSY/, and RST/ SCSI signals to be asserted on the SCSI bus. The differential pairs on the SCSI bus are reversed when connected to the 75LBC976, due to the active low nature of the SCSI bus.

**8-Bit/16-Bit SCSI and the Differential Interface** – In an 8-bit SCSI bus, the SD[15:8] pins on the LSI53C825A should be pulled up with a 1.5 k $\Omega$  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. In the LSI53C825AJ, the SDIRP1 pin is replaced by the TCK JTAG signal. If the device is used in a wide differential system, use the SDIRP0 pin to control the direction of the differential transceiver for both the SP0 and SP1 signals. The SDIRP0 signal is capable of driving both direction inputs from a transceiver.

#### 2.4.9.2 Terminator Networks

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. The terminators should be socketed, so that if not needed they may be removed, or there should be a means of disabling them with software.

SE cables can use a 220  $\Omega$  pull-up to the terminator power supply (Term Power) line and a 330  $\Omega$  pull-down to ground. Because of the high-performance nature of the LSI53C825A, Regulated (or Active) termination is recommended.

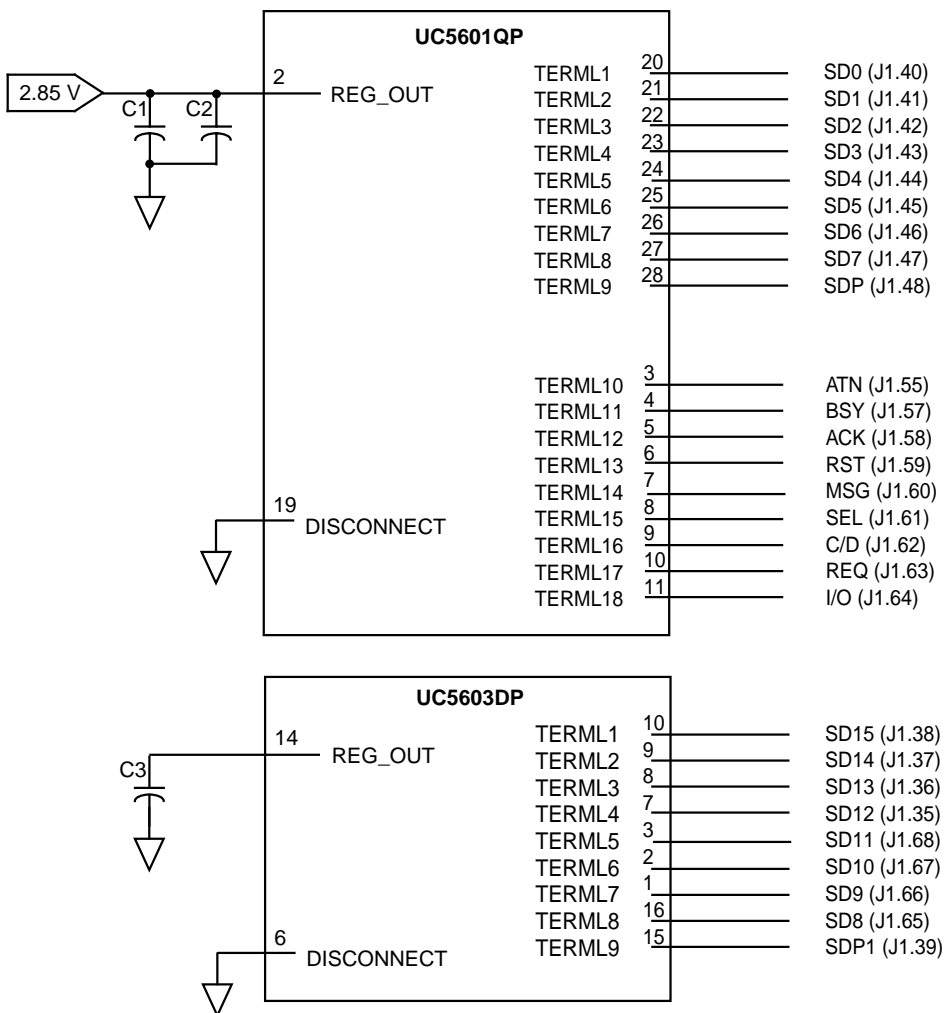
Note: If the LSI53C825A is to be used in a design with only an 8-bit SCSI bus, all 16 data lines still must be terminated or pulled HIGH.

Figure 2.3 is an example differential wiring diagram.

Figure 2.4 shows a Unitrode active terminator. For additional information, refer to the SCSI-2 Specification. TolerANT active negation can be used with either termination network.



**Figure 2.4 Regulated Termination**



Note:

1. C1 - 10  $\mu$ F SMT
2. C2 - 0.1  $\mu$ F SMT
3. C3 - 2.2  $\mu$ F SMT
4. J1 - 68-pin, high density "P" connector



## 2.4.10 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 initiator mode) tries to select a target and is reselected by another. The Select SCRIPTS instruction has an alternate address to which the SCRIPTS jumps 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 (SCID bits 5 and 6, respectively) should both be asserted so that the LSI53C825A may respond as an initiator or as a target. If only selection is enabled, the LSI53C825A 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 LSI53C825A has been selected (bit 5) and reselected (bit 4).

## 2.4.11 Synchronous Operation

The LSI53C825A can transfer synchronous SCSI data in both 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 LSI53C825A can receive data from the SCSI bus at a synchronous transfer period as short as 80 or 160 ns (with a 50 MHz clock), regardless of the transfer period used to send data. The LSI53C825A 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 LSI53C825A can send synchronous data at intervals as short as 100 ns for fast SCSI and 200 ns for SCSI-1.

### 2.4.11.1 Determining the Data Transfer Rate

Synchronous data transfer rates are controlled by bits in two different registers of the LSI53C825A. A brief description of the bits is provided below.

#### 2.4.11.2 SCSI Control Three (SCNTL3) Register, Bits [6:4]

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 50 MHz. The receive rate of synchronous SCSI data is one-fourth of the SCF divider output. For example, if SCLK is 40 MHz and the SCF value is set to divide by one, then the maximum rate at which data can be received is 10 MHz ( $40/(1*4) = 10$ ).

#### 2.4.11.3 SCSI Control Three (SCNTL3) Register, Bits [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.

#### 2.4.11.4 SCSI Transfer (SXFER) Register, Bits [7:5]

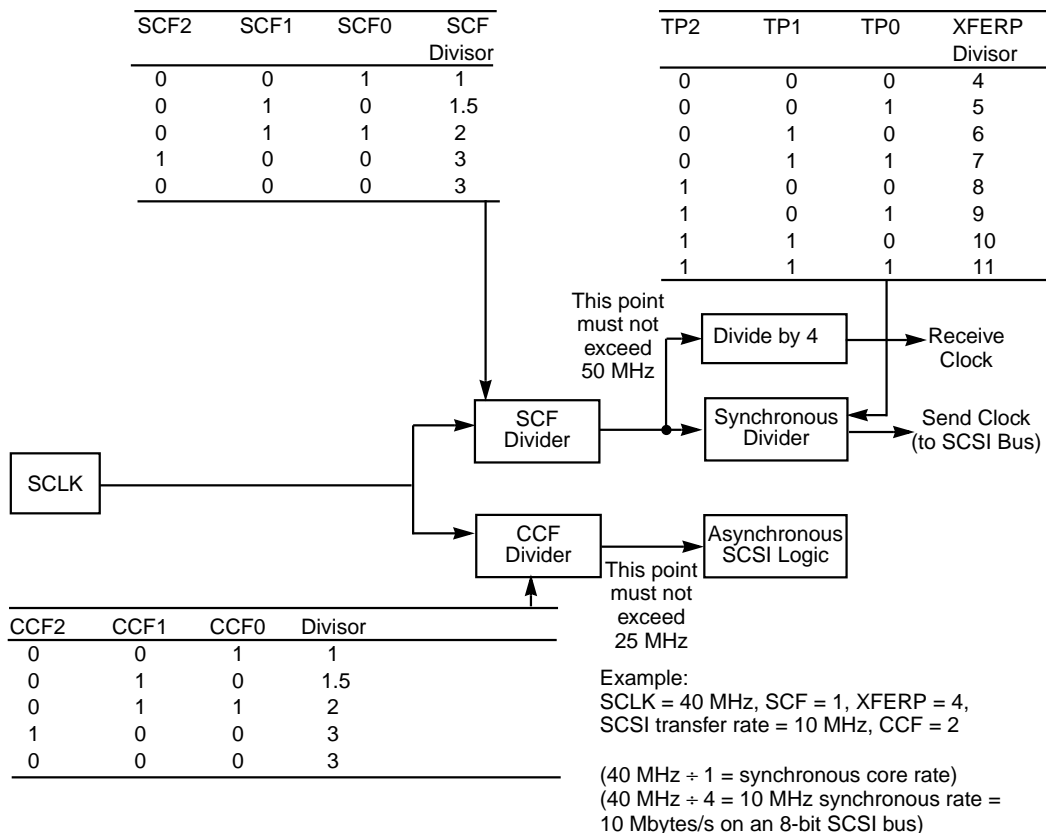
The TP[2:0] divider bits determine the SCSI synchronous transfer period when sending synchronous SCSI data in either initiator or target mode. This value further divides the output from the SCF divider.

### 2.4.12 Achieving Optimal SCSI Send Rates

To achieve optimal synchronous SCSI send timings, the SCF divisor value should be set HIGH, to divide the clock as much as possible before presenting the clock to the TP divider bits in the [SCSI Transfer \(SXFER\)](#) register. The TP[2:0] divider value should be as low as possible. For example, with a 40 MHz clock to achieve a 5 Mbytes/s send rate, the SCF bits can be set to divide by 1 and the TP bits to divide by 8; or the SCF bits can be set to divide by 2 and the TP bits set to divide by 2. Use the second option to achieve optimal SCSI timings.

[Figure 2.5](#) illustrates the clock division factors used in each register, and the role of the register bits in determining the transfer rate.

**Figure 2.5 Determining the Synchronous Transfer Rate**



## 2.4.13 Interrupt Handling

The SCRIPTS processor in the LSI53C825A performs 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 LSI53C825A.

### 2.4.13.1 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 set that indicates 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 LSI53C825A 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.

#### 2.4.13.2 Registers

The registers in the LSI53C825A that are used for detecting or defining interrupts are the [Interrupt Status \(ISTAT\)](#), [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 [Interrupt Status \(ISTAT\)](#) is the only register that can be accessed as a slave during SCRIPTS operation, therefore it is the register that is polled when polled interrupts are used. It is also the first register that should be read when the IRQ/ pin has been asserted in association with a hardware interrupt. 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. If the SIP bit in the ISTAT 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 \(ISTAT\)](#) 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 the 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 LSI53C825A 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 LSI53C825A 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 clear, 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 [Chip Test Three \(CTEST3\)](#).

**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 does 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 this register is set, the IRQ/ pin is not asserted when an interrupt condition occurs. The interrupt is not lost or ignored, but 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 [Interrupt Status \(ISTAT\)](#), this register cannot be accessed except by a SCRIPTS instruction during SCRIPTS execution.

### 2.4.13.3 Fatal vs. Nonfatal Interrupts

A fatal interrupt, as the name implies, always causes SCRIPTS to stop running. All nonfatal interrupts become fatal when they are enabled by setting the appropriate interrupt enable bit. Interrupt masking will be discussed in [Section 2.4.13.4, “Masking.”](#) All DMA interrupts (indicated by the DIP bit in [Interrupt Status \(ISTAT\)](#) and one or more bits in [DMA Status \(DSTAT\)](#) being set) are fatal.

Some SCSI interrupts (indicated by the SIP bit in the ISTAT and one or more bits in [SCSI Interrupt Status Zero \(SIST0\)](#) or [SCSI Interrupt Status One \(SIST1\)](#) being set) are nonfatal. When the LSI53C825A is operating in 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 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 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 LSI53C825A has been selected or reselected (SEL or RSL set), when the initiator has asserted 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.

#### **2.4.13.4 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 Initiator or Target mode.

If a nonfatal interrupt is masked and that condition occurs, the SCRIPTS do not stop, the appropriate bit in the SIST0 or SIST1 is still set, the SIP bit in the ISTAT is not set, and the IRQ/ pin is not asserted. See [Section 2.4.13.3, "Fatal vs. Nonfatal Interrupts,"](#) for a list of the nonfatal interrupts.

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 \(ISTAT\)](#) is set, but the IRQ/ pin is not asserted.

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

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

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

#### **2.4.13.5 Stacked Interrupts**

The LSI53C825A stacks interrupts if they occur one after the other. If the SIP or DIP bits in the [Interrupt Status \(ISTAT\)](#) 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 will move into the [SCSI Interrupt Status Zero \(SIST0\)](#), [SCSI Interrupt Status One \(SIST1\)](#), and [DMA Status \(DSTAT\)](#). 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 the 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 right into the SIST0 or

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.

#### 2.4.13.6 Halting in an Orderly Fashion

When an interrupt occurs, the LSI53C825A attempts to halt in an orderly fashion.

- If the interrupt occurs in the middle of an instruction fetch, the fetch is completed, except in the case of a Bus Fault. Execution does not begin, but the DSP points to the next instruction since it is updated when the current instruction is fetched.
- If the DMA direction is a write to memory and a SCSI interrupt occurs, the LSI53C825A attempts to flush the [DMA FIFO \(DFIFO\)](#) to memory before halting. Under any other circumstances only the current cycle is completed before halting, so the DFE bit in [DMA Status \(DSTAT\)](#) should be checked to see if any data remains in the DMA FIFO.
- SCSI SREQ/SACK handshakes that have begun are completed before halting.
- The LSI53C825A 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 DSP is updated to the transfer address before halting.
- All other instructions may halt before completion.

#### 2.4.13.7 Sample Interrupt Service Routine

The following is a sample of an interrupt service routine for the LSI53C825A. It can be repeated during polling or should be called when the IRQ/ pin is asserted during hardware interrupts.

1. Read [Interrupt Status \(ISTAT\)](#).
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 the [DMA Status \(DSTAT\)](#) to clear the interrupt condition and get the DMA interrupt status. The bits in the 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 [SCSI Interrupt Status Zero \(SIST0\)](#), [SCSI Interrupt Status One \(SIST1\)](#), and [DMA Status \(DSTAT\)](#) 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 ISR. It is recommended that the DMA interrupt is serviced before the SCSI interrupt, because a serious DMA interrupt condition could influence how the SCSI interrupt is acted upon.
6. When using polled interrupts, go back to Step 1 before leaving the 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.

## 2.4.14 Chained Block Moves

Since the LSI53C825A 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.

### 2.4.14.1 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 [SCSI Output Data Latch \(SODL\)](#) 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. Additionally, this bit can be used by the microprocessor or SCRIPTS for error detection and recovery purposes.

### 2.4.14.2 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. The bit can also be used by the microprocessor or SCRIPTS for error detection and recovery purposes.

#### **2.4.14.3 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.

#### **2.4.14.4 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.

#### **2.4.14.5 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.

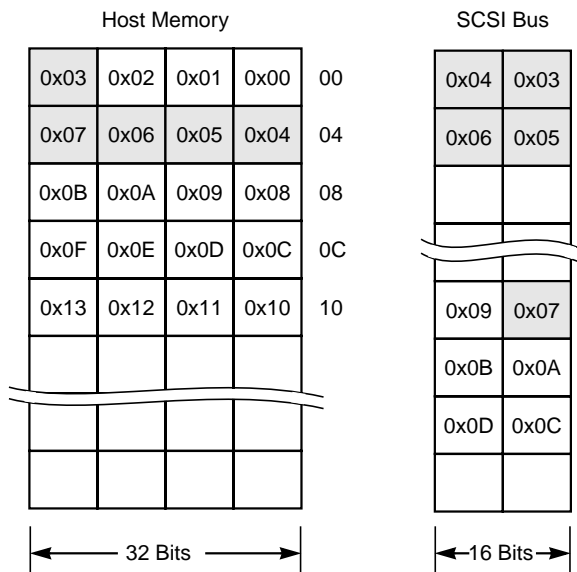
For receive data (Data-In for initiator or Data-Out for target), a chained Block Move instruction indicates that if a partial transfer occurred at the end of the instruction, the WSR flag is set. The high-order byte of the last SCSI transfer is stored in the [SCSI Wide Residue \(SWIDE\)](#) register rather than transferred to memory. The contents of the [SCSI Wide Residue \(SWIDE\)](#) 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 will be 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 (Nth – 1) Block Move instructions should be Chained Block Moves.

The Chained Block Move instruction is illustrated in [Figure 2.6](#).

**Figure 2.6 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 combined 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.

---

## 2.5 Power Management

This feature complies with the PCI Bus Power Management Interface Specification, Revision 1.0. The PCI Function Power States are defined in that specification: D0, D1, D2, and D3. D0 and D3 are required by specification, and D1 and D2 are optional. 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 power states for the SCSI function are independently controlled through two power state bits that are located in the PCI Configuration Space register 0x44. The bits are encoded as:

00b	D0
01b	Reserved
10b	Reserved
11b	D3

Power states D1 and D2 are not discussed because they have not been implemented as a new feature.

The Power states – D0 and D3 – are described below in conjunction with each SCSI function. Power state actions are separate for each function.

### 2.5.1 Power State D0

Power state D0 is the maximum power state and is the power-up default state for each function.

### 2.5.2 Power State D3

Power state D3 is the minimum power state, which includes subsettings called D3hot and D3cold. The devices are considered to be in power state D3cold when power is removed from them. D3cold can transition to D0 by applying Vcc and resetting the device. D3hot allows the device to transition to D0 using software. To obtain power reduction in D3hot, the SCSI clock and the SCSI clock doubler Phase Lock Loop (PLL) are disabled. Furthermore, the function's soft reset is continually asserted while in power state D3, which clears all pending interrupts and 3-states the SCSI bus. In addition, the function's PCI [Command](#) register is cleared.

# Chapter 3

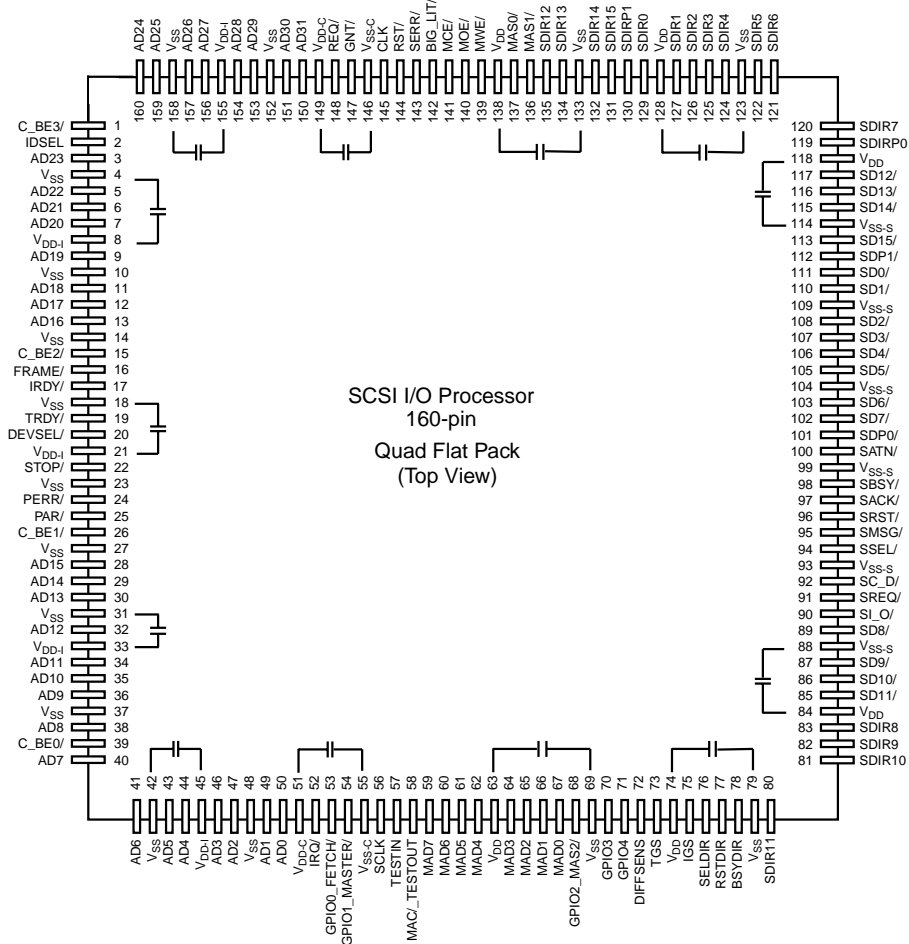
## Signal Descriptions

---

This chapter presents the LSI53C825A pin configuration and signal definitions using tables and illustrations. [Figure 3.1](#) through [Figure 3.2](#) are the pin diagrams for all versions of the LSI53C825A and [Figure 3.3](#) is the functional signal grouping. The pin definitions are presented in [Table 3.1](#) through [Table 3.10](#). The LSI53C825A is a pin-for-pin replacement for the LSI53C825. This chapter is divided into the following sections:

- [Section 3.1, “PCI Bus Interface Signals”](#)
- [Section 3.2, “MAD Bus Programming”](#)

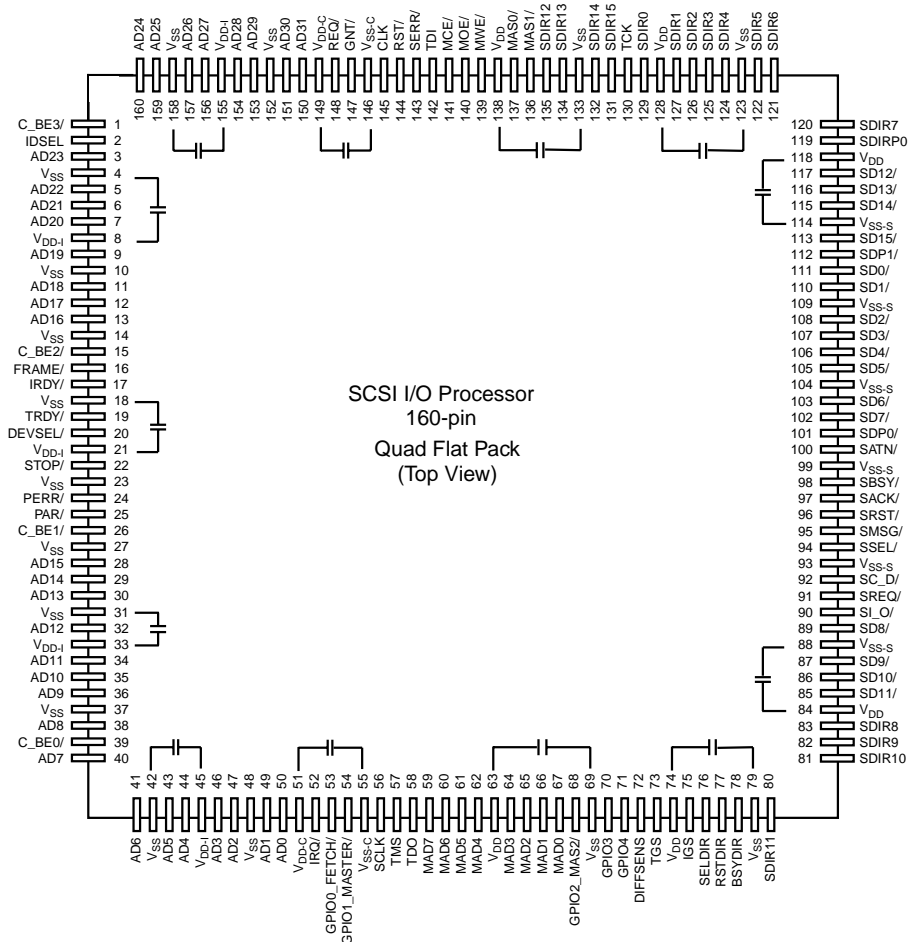
**Figure 3.1 LSI53C825A Pin Diagram**



Note: The decoupling capacitor arrangement shown above is recommended to maximize the benefits of the internal split ground system. Capacitor values between 0.01 and 0.1 $\mu$ F should provide adequate noise isolation. Because of the number of high current drivers on the LSI53C825A, a multilayer PC board with power and ground planes is required.



**Figure 3.2 LSI53C825AJ Pin Diagram**



The PCI/SCSI pin definitions are organized into the following functional groups: System, Address/Data, Interface Control, Arbitration, Error Reporting, SCSI, and Optional Interface. A slash (/) at the end of the 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.

There are four signal type definitions:

- I** Input, a standard input-only signal.
- O** Output, a standard output driver (typically a Totem Pole Output).
- 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.

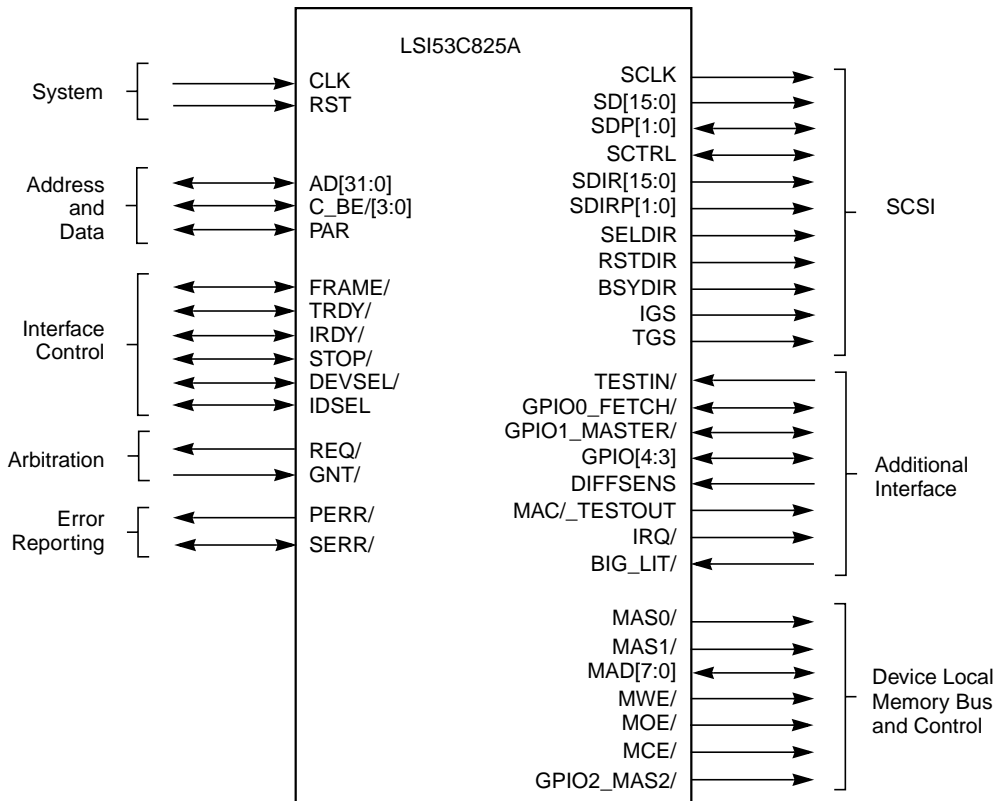
**Table 3.1 LSI53C825A, LSI53C825AJ, LSI53C825AE, and LSI53C825AJE Power and Ground Pins**

Symbol	Pin No.	Description
$V_{SS}$	4, 10, 14, 18, 23, 27, 31, 37, 42, 48, 69, 79, 123, 133, 152, 158	Ground to the PCI I/O pins
$V_{DD}$	8, 33, 45, 63, 74, 84, 118, 128, 138, 155	Power supplies to the Standard I/O pins
$V_{DD-I}^1$	8, 21, 33, 45, 155	$V_{DD}$ pad for PCI I/O pins
$V_{SS-S}$	88, 93, 99, 104, 109, 114	Ground to the SCSI bus I/O pins
$V_{SS-C}$	55, 146	Ground to the internal logic core
$V_{DD-C}$	51, 149	Power supplies to the internal logic core

1. These pins can accept a VDD source of 3.3 or 5 Volts. All other VDD pins must be supplied 5 Volts.

Figure 3.3 is the functional signal grouping for the LSI53C825A.

**Figure 3.3 LSI53C825A Functional Signal Grouping**



---

## 3.1 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](#), [SCSI Bus Interface Signals](#), [Additional Interface Signals](#), [External Memory Interface Signals](#), and [JTAG Signals](#).

### 3.1.1 System Signals

[Table 3.2](#) describes the signals for the System Signals group:

**Table 3.2     System Signals**

Name	Pin No.	Type	Description
CLK	145	I	<b>Clock</b> 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. This clock can optionally be used as the SCSI core clock; however, the LSI53C825A is not able to achieve Fast SCSI transfer rates.
RST/	144	I	<b>Reset</b> 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.

## 3.1.2 Address and Data Signals

Table 3.3 describes the signals for the Address and Data Signals group:

**Table 3.3 Address and Data Signals**

Name	Pin No.	Type	Description
AD[31:0]	150, 151, 153, 154, 156, 157, 159, 160, 3, 5, 6, 7, 9, 11, 12, 13, 28, 29, 30, 32, 34, 35, 36, 38, 40, 41, 43, 44, 46, 47, 49, 50	T/S	Physical longword <b>Address and Data</b> are multiplexed on the same PCI pins. During the first clock of a transaction, AD[31:0] contain a physical address. During subsequent clocks, AD[31:0] contain data. A bus transaction consists of an address phase, followed by one or more data phases. 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]/	1, 15, 26, 39	T/S	Bus <b>Command and Byte Enables</b> 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.
PAR	25	T/S	<b>Parity</b> is the even parity bit that protects the AD[31:0] and C_BE[3:0]/ lines. During address phase, both the address and command bits are covered. During data phase, both data and byte enables are covered.

### 3.1.3 Interface Control Signals

Table 3.4 describes the signals for the Interface Control Signals group:

**Table 3.4 Interface Control Signals**

Name	Pin No.	Type	Description
FRAME/	16	S/T/S	<b>Cycle Frame</b> is driven by the current master to indicate the beginning and duration of an access. FRAME/ is asserted to indicate a bus transaction is beginning. While FRAME/ is asserted, data transfers continue. When FRAME/ is deasserted, the transaction is in the final data phase or the bus is idle.
TRDY/	19	S/T/S	<b>Target Ready</b> 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 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 the target is prepared to accept data. Wait cycles are inserted until both IRDY/ and TRDY/ are asserted together.
IRDY/	17	S/T/S	<b>Initiator Ready</b> indicates the initiating agent's (bus master's) ability to complete the current data phase of the transaction. This signal 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 the master is prepared to accept data. Wait cycles are inserted until both IRDY/ and TRDY/ are asserted together.
STOP/	22	S/T/S	<b>Stop</b> indicates that the selected target is requesting the master to stop the current transaction.
DEVSEL/	20	S/T/S	<b>Device Select</b> 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	2	I	<b>Initialization Device Select</b> is used as a chip select in place of the upper 24 address lines during configuration read and write transactions.

### 3.1.4 Arbitration Signals

Table 3.5 describes the signals for the Arbitration Signals group:

**Table 3.5 Arbitration Signals**

Name	Pin No.	Type	Description
REQ/	148	O	<b>Request</b> indicates to the arbiter that this agent desires use of the PCI bus. This is a point-to-point signal. Every master has its own REQ/.
GNT/	147	I	<b>Grant</b> 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/.

### 3.1.5 Error Reporting Signals

Table 3.6 describes the signals for the Error Reporting Signals group:

**Table 3.6 Error Reporting Signals**

Name	Pin No.	Type	Description
PERR/	24	S/T/S	<b>Parity Error</b> may be pulsed active by an agent that detects a data parity error. PERR/ can be used by any agent to signal data corruptions.
SERR/	143	O	<b>System Error</b> is an open drain output pin used to report address parity errors. 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.

### 3.1.6 SCSI Bus Interface Signals

Table 3.7 describes the SCSI Bus Interface Signals group:

**Table 3.7 SCSI Bus Interface Signals**

Name	Pin No.	Type	Description
SCLK	56	I	<b>SCLK</b> 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.
SD[15:0]/ SDP[1:0]/	113, 115, 116, 117, 85, 86, 87, 89, 102, 103, 105, 106, 107, 108, 110, 111, 112, 101	I/O	<b>SCSI Data</b> includes the following data lines and parity signals: SD[15:0]/ (16-bit SCSI data bus), and SDP[1:0]/ (SCSI data parity bits).
SCTRL/	92, 90, 95, 91, 97, 98, 100, 96, 94	I/O	<b>SCSI Control</b> includes the following signals: SC_D/ SCSI phase line, command/data SI_O/ SCSI phase line, input/output SMMSG/ SCSI phase line, message SREQ/ Data handshake signal from target device SACK/ Data handshake signal from initiator device SBSY/ SCSI bus arbitration signal, busy SATN/ SCSI Attention, the initiator is requesting a message out phase SRST/ SCSI bus reset SSEL/ SCSI bus arbitration signal, select device
SDIR[15:0]	131, 132, 134, 135, 80, 81, 82, 83, 120, 121, 122, 124, 125, 126, 127, 129	O	Driver direction control for SCSI data lines.
SDIRP[1:0] (SDIPR1 not available on LSI53C825AJ)	130, 119	O	Driver direction control for SCSI parity signals. In the LSI53C825AJ, this pin is replaced by the TCK JTAG signal. If the device is used in a wide differential system, use the SDIRP0 pin to control the direction of the differential transceiver for both the SP0 and SP1 signals. The SDIRP0 signal is capable of driving both direction inputs from a transceiver.
SELDIR	76	O	Driver Enable Control for SCSI SEL/ signal.



**Table 3.7 SCSI Bus Interface Signals (Cont.)**

Name	Pin No.	Type	Description
RSTDIR	77	O	Driver Enable Control for SCSI RST/ signal.
BSYDIR	78	O	Driver Enable Control for SCSI BSY/ signal.
IGS	75	O	Direction Control for initiator driver group.
TGS	73	O	Direction Control for target driver group.

### 3.1.7 Additional Interface Signals

Table 3.8 describes the signals for the Additional Interface Signals group:

**Table 3.8 Additional Interface Signals**

Name	Pin No.	Type	Description
TESTIN (Not available on LSI53C825AJ)	57, NA	I	<b>Test In.</b> When this pin is driven LOW, the LSI53C825A connects all inputs and outputs to an “AND tree.” The SCSI control signals and data lines are not connected to the “AND tree.” The output of the “AND tree” is connected to the Test Out pin. This allows manufacturers to verify chip connectivity and determine exactly which pins are not properly attached. When the TESTIN pin is driven LOW, internal pull-ups are enabled on all input, output, and bidirectional pins, all outputs and bidirectional signals are 3-stated, and the MAC/_TESTOUT pin is enabled. Connectivity can be tested by driving one of the LSI53C825A pins LOW. The MAC/_TESTOUT pin should respond by also driving LOW.
GPIO0_FETCH/	53/70/N5	I/O	<b>General Purpose I/O</b> pin. Optionally, when driven LOW, this pin indicates that the next bus request will be for an opcode fetch. This pin powers up as a general purpose input. This pin has two specific purposes in the LSI Logic SDMS software. SDMS software uses it to toggle SCSI device LEDs, turning on the LED whenever the LSI53C825A is on the SCSI bus. SDMS software drives this pin LOW to turn on the LED, or drives it HIGH to turn off the LED. This signal can also be used as data I/O for serial EEPROM access. In this case it is used with the GPIO0 pin, which serves as a clock, and the pin can be controlled from PCI configuration register 0x35 or observed from the <a href="#">General Purpose (GPREG)</a> register, at address 0x07.

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

Name	Pin No.	Type	Description
GPIO1_MASTER/	54	I/O	<b>General purpose I/O</b> pin. Optionally, when driven LOW, indicates that the LSI53C825A is bus master. This pin powers up as a general purpose input. LSI Logic SDMS software supports use of this signal in serial EEPROM applications, when enabled, in combination with the GPIO0 pin. When this signal is used as a clock for serial EEPROM access, the GPIO1 pin serves as data, and the pin is controlled from PCI configuration register 0x35.
GPIO[4:3]	71, 70	I/O	<b>General purpose I/O</b> pins. GPIO4 powers up as an output. It can be used as the enable line for $V_{PP}$ the 12 Volt power supply to the external flash memory interface. GPIO3 powers up as an input. LSI Logic SDMS software uses GPIO3 to detect a differential board. If the pin is pulled LOW externally, the board will be configured by SDMS software as a differential board. If it is pulled HIGH or left floating, SDMS software will configure it as a SE board. The LSI Logic PCI to SCSI host adapters use the GPIO4 pin in the process of flashing a new SDMS software ROM.
DIFFSENS	72	I	The <b>Differential Sense</b> pin detects the presence of a SE device on a differential system. When external differential transceivers are used and a zero is detected on this pin, all chip SCSI outputs will be 3-stated to avoid damage to the transceivers. This pin should be tied HIGH during SE operation. The normal value of this pin is 1.
MAC/_TESTOUT (Not available on LSI53C825AJ)	58, NA	T/S	<b>Memory Access Control.</b> This pin can be programmed to indicate local or system memory accesses (non-PCI applications). It is also used to test the connectivity of the LSI53C825A signals using an "AND tree" scheme. The MAC/_TESTOUT pin is only driven as the Test Out function when the TESTIN/ pin is driven LOW.
IRQ/	52/69/M5	O	<b>Interrupt.</b> 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 programmed as either open drain with an internal weak pull-up or, optionally, as a totem pole driver. Refer to the description of <a href="#">DMA Control (DCNTL)</a> register, bit 3, for additional information.

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

Name	Pin No.	Type	Description
BIG_LIT/ (Not available on LSI53C825AJ)	142, NA	I	<b>Big_Little Endian Select.</b> When this pin is driven LOW, the LSI53C825A routes the first byte of an aligned SCSI to PCI transfer to byte lane zero of the PCI bus and subsequent bytes received are routed to ascending lanes. An aligned PCI to SCSI transfer routes PCI byte lane zero onto the SCSI bus first, and transfers ascending byte lanes in order. When this pin is driven HIGH, the LSI53C825A routes the first byte of an aligned SCSI to PCI transfer to byte lane three of the PCI bus and subsequent bytes received are routed to descending lanes. An aligned PCI to SCSI transfer routes PCI byte lane three onto the SCSI bus first and transfer descending byte lanes in order. This mode of operation also applies to the external memory interface. When this pin is driven in little endian mode and the chip is performing a read from external memory, the byte of data accessed at location 0x00000 is routed to PCI byte lane zero and the data accessed at location 0x00003 is routed to PCI byte lane three. When the chip is performing a write to flash memory, PCI byte lane zero is routed to location 0x00000 and ascending byte lanes are routed to subsequent memory locations. When this pin is driven in big endian mode and the chip is performing a read from external memory, the byte of data accessed at location 0x00000 is routed to PCI byte lane three and the data accessed at location 0x00003 is routed to byte lane zero. When the chip is performing a write to flash memory, PCI byte lane three is routed to location 0x00000 and descending byte lanes is routed to subsequent memory locations.

### 3.1.8 External Memory Interface Signals

[Table 3.9](#) describes the signals for the External Memory Interface Signals group:

**Table 3.9 External Memory Interface Signals**

Name	Pin No.	Type	Description
MAS0/	137/179/A8	O	<b>Memory Address Strobe 0.</b> This pin is used to latch in the least significant address byte of an external EPROM or flash memory. Since the LSI53C825A 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.
MAS1/	136/178/B8	O	<b>Memory Address Strobe 1.</b> This pin is used to latch in the address byte corresponding to address bits [15:8] of an external EPROM or flash memory. Since the LSI53C825A 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]	59, 60, 61, 62, 64, 65, 66, 67	I/O	<b>Memory Address/Data Bus.</b> This bus is used in conjunction with the memory address strobe pins and external address latches to assemble up to a 20-bit address for an external EPROM or flash memory. This bus will put out the most significant byte first and finish with the least significant bits. It is also used to write data to a flash memory or read data into the chip from external EPROM/flash memory. See <a href="#">Section 3.2, "MAD Bus Programming,"</a> for more details.
MWE/	139/181/C7	O	<b>Memory Write Enable.</b> This pin is used as a write enable signal to an external flash memory.
MOE/	140/182/B7	O	<b>Memory Output Enable.</b> This pin is used as an output enable signal to an external EPROM or flash memory during read operations.
MCE/	141/183/A7	O	<b>Memory Chip Enable.</b> This pin is used as a chip enable signal to an external EPROM or flash memory device.
GPIO2_ MAS2/	68/87/J8	I/O	<b>General Purpose I/O</b> pin. Optionally, this pin is used as a Memory Address Strobe 2 if an external memory with more than 16 bits of addressing is specified by the pull-down resistors at power-up and bit 0 in the <a href="#">Expansion ROM Base Address</a> register is set.

### 3.1.9 JTAG Signals

[Table 3.10](#) describes the signals for the JTAG Signals group:

**Table 3.10 JTAG Signals (LSI53C825AJ, LSI53C825AJE Only)**

Name	Pin No.	Type	Description
TCK	130/130	–	<b>Test Clock</b> pin for JTAG boundary scan.
TMS	57/57	–	<b>Test Mode Select</b> pin for JTAG boundary scan.
TDI	142/142	–	<b>Test Data In</b> pin for JTAG boundary scan.
TDO	58/58	–	<b>Test Data Out</b> pin for JTAG boundary scan.

---

## 3.2 MAD Bus Programming

The MAD[7:0] pins, in addition to serving as the address/data bus for the local memory interface, are also used to program power-up options for the chip. A particular option is programmed by connecting a 4.7 k $\Omega$  resistor between the appropriate MAD(x) pin and Vss. The pull-down resistors require that HC or HCT external components are used for a memory interface.

- **MAD[7]** – has no functionality. Do not place a pull-down resistor on this pin.
- **MAD[6]** – Subsystem Data configuration. Please refer to the [Table 3.11](#) for the different configurations.
- **MAD[5]** – SCRIPTS RAM disable. Connecting a 4.7 k $\Omega$  resistor between MAD[5] and Vss disables SCRIPTS RAM.
- **MAD[4]** – Subsystem Data configuration. Please refer to the [Table 3.11](#) and [Table 3.12](#) below for the different configurations.

**Table 3.11 Subsystem Data Configuration Table for the LSI53C825AE (PCI Rev ID 0x26)**

Mode MAD Pins	Offset	Normal 4-hi, 6-hi	Read/Write 4-hi, 6-lo	Reserved 4-low, 6-hi	LSI Logic 4-low, 6-lo
Vendor ID	0x00	0x1000	0x1000	–	0x1000
Device ID	0x02	0x000F	0x000F	–	0x000F
Subsystem Vendor ID	0x2C	0x1000	0x0000	–	0x0000
Subsystem ID	0x2E	0x1000	0x0000	–	0x0000

**Table 3.12 Subsystem Data Configuration Table for the LSI53C825A (PCI Rev ID 0x14) Revision G Only<sup>1</sup>**

Mode MAD Pins	Offset	Normal 4-hi, 6-hi	Read/Write 4-hi, 6-lo	Reserved 4-low, 6-hi	LSI Logic 4-low, 6-lo
Vendor ID	0x00	0x1000	0x1000	–	0x1000
Device ID	0x02	0x000F	0x000F	–	0x000F
Subsystem Vendor ID	0x2C	0x0000	0x0000	–	0x1000
Subsystem ID	0x2E	0x0000	0x0000	–	0x1000

1. The chip revisions before Revision G of the LSI53C825A (PCI Rev ID 0x14) do not support different Subsystem Data Configurations. The [Subsystem ID \(SSID\)](#) and [Subsystem Vendor ID \(SSVID\)](#) registers are hardwired to zero values.

- **MAD[3:1]** – used to set the size of the external expansion ROM device attached. Encoding for these pins are listed in [Table 3.13](#).

**Table 3.13 External Memory Support**

<b>MAD[3:1]</b>	<b>Available Memory Space</b>
000	16 Kbytes
001	32 Kbytes
010	64 Kbytes
011	128 Kbytes
100	256 Kbytes
101	512 Kbytes
110	1024 Kbytes
111	No external memory present

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

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





# Chapter 4

## Registers

---

This chapter describes all LSI53C825A registers and is divided into the following sections:

- [Section 4.1, “Configuration Registers”](#)
- [Section 4.2, “Operating Registers”](#)

---

### 4.1 Configuration Registers

The Configuration registers are accessible only by the system BIOS during PCI configuration cycles, and they are not available to the user at any time. These registers can be accessed by SCRIPTS or the host processor. The lower 128 bytes hold configuration data while the upper 128 bytes hold the LSI53C825A operating registers, which are described in [Chapter 5, “SCSI SCRIPTS Instruction Set.”](#) Please note that the information about lower and upper bytes only applies to the LSI53C825A and not the LSI53C825AE.

Note: The configuration register descriptions provide general information only, to indicate which PCI configuration addresses are supported in the LSI53C825A. For detailed information, refer to the PCI Specification.

[Table 4.1](#) shows the PCI configuration registers implemented by the LSI53C825A/825AE.

All PCI-compliant devices, such as the LSI53C825A, must support the [Vendor ID](#), [Device ID](#), [Command](#), and [Status](#) registers. Support of other PCI-compliant registers is optional. In the LSI53C825A, 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 LSI53C825A 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	0x08
Not Supported	Header Type	Latency Timer	Cache Line Size	0x0C
Base Address Zero (I/O) <sup>1</sup> SCSI Operating Registers				0x10
Base Address One (Memory) <sup>2</sup> SCSI Operating Registers				0x14
Base Address Two (Memory) SCRIPTS RAM <sup>3</sup>				0x18
Not Supported				0x1C
Not Supported				0x20
Not Supported				0x24
Reserved				0x28
Subsystem ID (SSID)		Subsystem Vendor ID (SSVID)		0x2C
Expansion ROM Base Address <sup>4</sup>				0x30
Reserved			Capability Pointer	0x34
Reserved				0x38
Max_Lat	Min_Gnt	Interrupt Pin	Interrupt Line	0x3C
Power Management Capabilities		Next Item Pointer	Capability ID	0x40
Data	Bridge Support Extension	Power Management Control/Status		0x44

1. I/O Base is supported.

2. Memory Base is supported.

3. This register powers up enabled and can be disabled by pull-down resistors on the MAD5 pin.

4. If expansion memory is enabled through pull-down resistors on the MAD[7:0] bus.

Note: Addresses 0x40–7F are not defined for the LSI53C825A. Addresses 0x48–7F are not defined for the LSI53C825AE. All unsupported registers are not writable and return all zeros when read. Reserved registers also return zeros when read.

**Register: 0x00****Vendor ID****Read Only**

15															0
VID															
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0

**VID****Vendor ID****[15:0]**

This field identifies the manufacturer of the device. The Vendor ID is 0x1000.

**Register: 0x02****Device ID****Read Only**

15															0
DID															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

This field identifies the particular device. The LSI53C825A device ID is 0x0003. This value is the same as in the LSI53C825, since the LSI53C825A is a drop in replacement. The devices are uniquely identified in the upper nibble of the [Revision ID](#) register.

**Register: 0x04****Command****Read/Write**

15							9	8	7	6	5	4	3	2	1	0
R							SE	R	EPER	R	WIE	R	EBM	EMS	EIS	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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 LSI53C825A is logically disconnected from the PCI bus for all accesses except configuration accesses.

In the LSI53C825A, bits 3 through 5 and bit 7 and 9 are not implemented. Bits 10 through 15 are reserved.

<b>R</b>	<b>Reserved</b>	<b>[15:9]</b>
<b>SE</b>	<b>SERR/Enable</b> This bit enables the SERR/ driver. SERR/ is disabled when this bit is clear. The default value of this bit is zero. This bit and bit 6 must be set to report address parity errors.	<b>8</b>
<b>R</b>	<b>Reserved</b>	<b>7</b>
<b>EPER</b>	<b>Enable Parity Error Response</b> This bit allows the LSI53C825A to detect parity errors on the PCI bus and report these errors to the system. Only data parity checking is enabled. The LSI53C825A always generates parity for the PCI bus.	<b>6</b>
<b>R</b>	<b>Reserved</b>	<b>5</b>
<b>WIE</b>	<b>Write and Invalidate Mode</b> This bit, when set, causes Memory Write and Invalidate cycles to be issued on the PCI bus after certain conditions have been met. For more information on these conditions, refer to <a href="#">Section 2.1.2.7, "Memory Write and Invalidate Command."</a> To enable Write and Invalidate Mode, bit 10 in the <a href="#">Chip Test Three (CTEST3)</a> register (operating register set) must also be set.	<b>4</b>
<b>R</b>	<b>Reserved</b>	<b>3</b>
<b>EBM</b>	<b>Enable Bus Mastering</b> This bit controls the ability of the LSI53C825A to act as a master on the PCI bus. A value of zero disables the device from generating PCI bus master accesses. A value of one allows the LSI53C825A to behave as a bus master. The LSI53C825A must be a bus master in order to fetch SCRIPTS instructions and transfer data.	<b>2</b>
<b>EMS</b>	<b>Enable Memory Space</b> This bit controls the ability of the LSI53C825A to respond to Memory Space accesses. A value of zero disables the device response. A value of one allows the LSI53C825A to respond to Memory Space accesses at the address specified by <a href="#">Base Address One (Memory)</a> .	<b>1</b>

**EIS**                      **Enable I/O Space**                      **0**

This bit controls the LSI53C825A response to I/O space accesses. A value of zero disables the response. A value of one allows the LSI53C825A to respond to I/O space accesses at the address specified in [Base Address One \(Memory\)](#).

**Register: 0x06**  
**Status**  
**Read/Write**

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

The [Status](#) register is used to record status information for PCI bus related events.

In the LSI53C825A, bits 0 through 4 are reserved and bits 5, 6, 7, and 11 are not implemented by the LSI53C825A.

Reads to this register behave normally. Writes are slightly different in that bits can be cleared, but not set. A bit is reset 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 LSI53C825A whenever it detects a data parity error, even if parity error handling is disabled.

**SSE**                      **Signaled System Error**                      **14**  
This bit is set whenever a 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 with a target abort.

R	Reserved	11								
DT[1:0]	<b>DEVSEL/ Timing</b> These bits encode the timing of DEVSEL/. These are encoded as:	[10:9]								
	<table><tr><td>0b00</td><td>Fast</td></tr><tr><td>0b01</td><td>Medium</td></tr><tr><td>0b10</td><td>Slow</td></tr><tr><td>0b11</td><td>Reserved</td></tr></table>	0b00	Fast	0b01	Medium	0b10	Slow	0b11	Reserved	
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. In the LSI53C825A, 0b01 is supported.									
DPR	<b>Data Parity Reported</b> This bit is set when the following three conditions are met:	8								
	<ul style="list-style-type: none"><li>• The bus agent asserted PERR/ itself or observed PERR/ asserted and;</li><li>• The agent setting this bit acted as the bus master for the operation in which the error occurred and;</li><li>• The Parity Error Response bit in the <a href="#">Command</a> register is set.</li></ul>									
R	Reserved	[7:5]								
NC	<b>New Capabilities (NC)</b> This bit is set to indicate a list of extended capabilities such as PCI Power Management. This bit is Read Only, and applies to the LSI53C825AE only.	4								
R	Reserved	[3:0]								

**Register: 0x08****Revision ID****Read Only**

7					0		
RID							
LSI53C825AE							
0	0	1	0	0	1	1	0
LSI53C825A							
0	0	0	1	0	1	0	0

**RID****Revision ID****[7:0]**

This register specifies a device specific revision identifier. For revision A of the LSI53C825AE, the value of this register is 0x26.

**Register: 0x09****Class Code****Read Only**

23																									0
CC[23:0]																									
0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

**CC[23:0]****Class Code****[23:0]**

This 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. Cache mode is enabled and disabled by the Cache Line Size Enable (CLSE) bit, bit 7 in the [DMA Control \(DCNTL\)](#) register. Setting this bit causes the LSI53C825A to align to cache line boundaries before allowing any bursting, except during Memory Moves in which the read and write addresses are not aligned to a burst size boundary. For more information on this register, see [Section 2.1.3.1, "Support for PCI Cache Line Size Register."](#)

## Register: 0x0D

### Latency Timer

Read/Write

7							0
LT							
0	0	0	0	0	0	0	0

LT[7:0]

### 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 LSI53C825A 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 LSI53C825A:

$$\text{Latency} = 2 + (\text{Burst Size} * (\text{typical wait states} + 1)).$$

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[7:0]****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. The value of this register is 0x00.

**Register: 0x10****Base Address Zero (I/O)****Read/Write**

31																															0									
BARZ																																								
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	x	x	x	x	x	x	x	x	x	x	1

**BARZ****Base Address Register Zero (I/O)****[31:0]**

This 32-bit register has bit zero hardwired to one. Bit 1 is reserved and must return a zero on all reads, and the other bits are used to map the device into I/O space.

**Register: 0x14****Base Address One (Memory)****Read/Write**

31																															0			
BARO																																		
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	x	x	x	x	0

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

This register has bit 0 hardwired to zero. For detailed information on the operation of this register, refer to the PCI Specification.

**Register: 0x18****RAM Base Address Two (Memory) SCRIPTS RAM****Read/Write**

31																															0			
BART																																		
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	x	x	x	x	0

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

This register holds the memory base address of the 4 Kbyte internal RAM. The user can read this register through the [Scratch Register B \(SCRATCHB\)](#) register in the operating register set when bit 3 of the [Chip Test Two \(CTEST2\)](#) register is set.

**Register: 0x2C****Subsystem Vendor ID (SSVID)****Read Only**

15															0
SSVID															
LSI53C825AE															
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
LSI53C825A															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

This register supports subsystem identification, which has a default value of 0x1000 in the LSI53C825AE ([Section 3.2, “MAD Bus Programming”](#)). To write to this register, connect a 4.7 kΩ resistor between the MAD[6] pin and V<sub>SS</sub> and leave the MAD[4] pin unconnected. The MAD[6] and MAD[4] pins have internal pull-up resistors and are sensed shortly after the deassertion of chip reset. In revisions before Rev. G of the LSI53C825A, the MAD[6] and MAD[4] pins do not support the SSID and SSVID configurations, and only values of 0x0000 can be found in the Subsystem Data register.

**Register: 0x2E**  
**Subsystem ID (SSID)**  
**Read Only**

15															0	
SSID																
LSI53C825AE																
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
LSI53C825A																
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

This register supports subsystem identification, which has a default value of 0x1000 in the LSI53C825AE ([Section 3.2, “MAD Bus Programming”](#)). To write to this register, connect a 4.7 kΩ resistor between the MAD[6] pin and V<sub>SS</sub> and leave the MAD[4] pin unconnected. The MAD[6] and MAD[4] pins have internal pull-up resistors and are sensed shortly after the deassertion of chip reset. In revisions before Revision G of the LSI53C825A, the MAD[6] and MAD[4] pins do not support the SSID and SSVID configurations, and only values of 0x0000 can be found in the Subsystem Data register.

**Register: 0x30**  
**Expansion ROM Base Address**  
**Read/Write**

31																															0	
ERBA																																
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	0	0	0

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

This four-byte register handles the base address and size information for expansion ROM. It functions exactly like the [Base Address Zero \(I/O\)](#) and [Base Address 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 can be used with or without an expansion ROM depending on the system configuration. To access the external memory interface, the Memory Space bit in the [Command](#) register must also be set.

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 LSI53C825A will respond 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, will return ones in the upper 17 bits.

**Register: 0x34**  
**Capability Pointer**  
**Read Only**

7				0			
CP							
0	1	0	0	0	0	0	0

**CP**                      **Capabilities Pointer**                      **[7:0]**  
This register provides an offset into the function's PCI Configuration Space for the location of the first item in the capabilities linked list. Only the LSI53C825AE sets this register to 0x40. The capability pointer replaces the General Purpose Pin Control register in earlier revisions of the LSI53C825A.

**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 will write the routing information into this register as it initiates and configures the system. The value in this register tells which input of the system interrupt controller(s) the device's interrupt pin has been 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[7:0]****Interrupt Pin****[7:0]**

This register tells 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 these registers is in units of 0.25 microseconds. The LSI53C25A 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 these registers is in units of 0.25 microseconds. The LSI53C825A 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. Only the LSI53C825AE sets this register to 0x01.

**Register: 0x41**  
**Next Item Pointer**  
**Read Only**

7								0
NIP								
0	0	0	0	0	0	0	0	

**NIP** **Next\_Item\_Ptr** **[7:0]**  
This register describes the location of the next item in the function's capability list. The default value for this register is 0x00, indicating that power management is the last capability in the linked list of extended capabilities. This register applies to the LSI53C825AE only.

**Register: 0x42**  
**Power Management Capabilities**  
**Read Only**

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

This register applies to the LSI53C825AE only and indicates the power management capabilities.

**PMES[4:0]** **PME Support** **[15:11]**  
This field is always set to 00000b because the LSI53C825AE does not provide a PME signal.

**D2S** **D2 Support** **10**  
This device does not support the D2 power management state.

**D1S** **D1 Support** **9**  
This device does not support the D1 power management state.

<b>R</b>	<b>Reserved</b>	<b>[8:6]</b>
<b>DSI</b>	<b>Device Specific Initialization</b> This bit is cleared to indicate that the LSI53C825A requires no special initialization before the generic class device driver is able to use it.	<b>5</b>
<b>APS</b>	<b>Auxiliary Power Source</b> Because the device does not provide a PME signal, this bit always returns a 0. This indicates that no auxiliary power source is required to support the PME signal in the D3cold power management state.	<b>4</b>
<b>PMEC</b>	<b>PME Clock</b> This bit always returns a zero value because the devices do not provide a PME signal.	<b>3</b>
<b>VER[2:0]</b>	<b>Version</b> This field is set of 001b to indicate that the device complies with Revision 1.0 of the PCI Power Management Interface Specification.	<b>[2:0]</b>

## Register: 0x44

### Power Management Control/Status

Read Write

15	14	13	12				9	8	7					2	1	0
PST	DSCL			DSL T				PEN		R					PWS	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register applies to the LSI53C825AE only and indicates the power management control and status descriptions.

<b>PST</b>	<b>PME Status</b> The LSI53C825A always returns a zero for this bit, indicating that PME signal generation is not supported from D3cold.	<b>15</b>
<b>DSCL</b>	<b>Data Scale</b> The LSI53C825A does not support the <a href="#">Data</a> register. Therefore these two bits are always set to 00b.	<b>[14:13]</b>
<b>DSL T</b>	<b>Data Select</b> This device does not support the <a href="#">Data</a> register. Therefore these four bits are always set to 0000b.	<b>[12:9]</b>



<b>PEN</b>	<b>PME Enable</b>	<b>8</b>
	The LSI53C825A always returns a zero for this bit to indicate that PME assertion is disabled.	
<b>R</b>	<b>Reserved</b>	<b>[7:2]</b>
<b>PWS</b>	<b>Power State</b>	<b>[1:0]</b>
	Bits [1:0] are used to determine the current power state for the LSI53C825A. They are used to place the LSI53C825A in a new power state. Power states are defined as:	
	<hr/>	
	0b00	D0
	0b01	Reserved
	0b10	Reserved
	0b11	D3 hot
	<hr/>	

**Register: 0x46**  
**Bridge Support Extensions (PMCSR\_BSE)**  
**Read Only**

7							0
BSE							
0	0	0	0	0	0	0	0

<b>BSE[7:0]</b>	<b>Bridge Support Extensions</b>	<b>[7:0]</b>
	This register applies to the LSI53C825A only and can support PCI bridge specific functionality, if required. The default value always returns 0x00.	

**Register: 0x47**  
**Data**  
**Read Only**

7							0
DATA							
0	0	0	0	0	0	0	0

<b>DATA[7:0]</b>	<b>Data</b>	<b>[7:0]</b>
	This register applies to the LSI53C825AE only and provides an optional mechanism for the function to report state dependent operating data. The LSI53C825AE returns 0x00 as the default value.	

---

## 4.2 Operating Registers

This section contains descriptions of all LSI53C825A operating registers. [Table 4.2](#), the register map, lists registers by operating and configuration addresses. The terms “set” and “assert” are used to refer to bits that are programmed to a binary one. Similarly, the terms “deassert,” “clear,” and “reset” are used to refer to bits that are programmed to a binary zero. Any bits marked as reserved should always be written to zero; mask all information read from them. Reserved bit functions may be changed at any time. Unless otherwise indicated, all bits in registers are active high, 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.

Note: The only register that the host CPU can access while the LSI53C825A is executing SCRIPTS is the [Interrupt Status \(ISTAT\)](#) register; attempts to access other registers will 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.

The LSI53C825A cannot fetch SCRIPTS instructions from the operating register space. Instructions must be fetched from system memory or the internal SCRIPTS RAM.

**Table 4.2 LSI53C825A Register Map**

31	16 15			0
SCNTL3	SCNTL2	SCNTL1	SCNTL0	0x00
GPREG	SDID	SXFER	SCID	0x04
SBCL	SSID	SOCL	SFBR	0x08
SSTAT2	SSTAT1	SSTAT0	DSTAT	0x0C
DSA				0x10
Reserved			ISTAT	0x14
CTEST3	CTEST2	CTEST1	Reserved	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
GPCNTL	MACNTL	SWIDE	SLPAR	0x44
RESPID1	RESPID0	STIME1	STIME0	0x48
STEST3	STEST2	STEST1	STEST0	0x4C
Reserved		SIDL		0x50
Reserved		SODL		0x54
Reserved		SBDL		0x58
SCRATCH B				0x5C
SCRATCH C				0x60
SCRATCH D				0x64
SCRATCH E				0x68
SCRATCH F				0x6C
SCRATCH G				0x70
SCRATCH H				0x74
SCRATCH I				0x78
SCRATCH J				0x7C

**Register: 0x00 (0x80)**  
**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 LSI53C825A 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 LSI53C825A 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 LSI53C825A has won arbitration.
4. Once the LSI53C825A has won arbitration, SSEL/ must be asserted using the [SCSI Output Control Latch \(SOCL\)](#) for a bus clear plus a bus settle delay (1.2  $\mu$ s) before a low level selection can be performed.

### Full Arbitration, Selection/Reselection

1. The LSI53C825A 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 LSI53C825A detects a higher priority ID, the LSI53C825A deasserts BSY, deasserts its ID, and waits until the next bus free state to try arbitration again.
4. The LSI53C825A repeats arbitration until it wins control of the SCSI bus. When it has won, the Won Arbitration bit is set in the [SCSI Status Zero \(SSTAT0\)](#) register, bit 2.
5. The LSI53C825A 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 LSI53C825A 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 LSI53C825A 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. An arbitration sequence should not be started if the connected (CON) bit in the [SCSI Control One \(SCNTL1\)](#) register, bit 4, indicates that the LSI53C825A is already connected to the SCSI bus. This bit is automatically cleared when the arbitration sequence is complete. If a sequence aborts, check bit 4 in the [SCSI Control One \(SCNTL1\)](#) register to verify that the LSI53C825A did not connect to the SCSI bus.

<b>WATN</b>	<b>Select with SATN/ on a Start Sequence</b> <b>4</b> When this bit is set and the LSI53C825A is in the initiator mode, the SATN/ signal is asserted during LSI53C825A selection of a SCSI target device. This is to inform the target that the LSI53C825A has a message to send. If a selection time-out occurs while attempting to select a target device, SATN/ deasserts at the same time SSEL/ deasserts. When this bit is clear, the SATN/ signal is not asserted during selection. When executing SCSI SCRIPTS, this bit is controlled by the SCRIPTS processor, but it may be set manually in low level mode.
<b>EPC</b>	<b>Enable Parity Checking</b> <b>3</b> 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. Parity is also checked as data goes from the SCSI FIFO to the DMA FIFO. If a parity error is detected, bit 0 of the <a href="#">SCSI Interrupt Status Zero (SIST0)</a> register is set and an interrupt may be generated.  If the LSI53C825A is operating in initiator mode and a parity error is detected, SATN/ can optionally be asserted, but the transfer continues until the target changes phase. When this bit is cleared, parity errors are not reported.
<b>R</b>	<b>Reserved</b> <b>2</b>
<b>AAP</b>	<b>Assert SATN/ on Parity Error</b> <b>1</b> When this bit is set, the LSI53C825A 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. The Enable Parity Checking bit must also be set for the LSI53C825A 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 LSI53C825A. 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 LSI53C825A 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 (0x81)**  
**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 LSI53C825A drives the contents of the [SCSI Output Data Latch \(SODL\)](#) onto the SCSI data bus. When the LSI53C825A is an initiator, the SCSI I/O signal must be inactive to assert the SODL contents onto the SCSI bus. When the LSI53C825A 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 LSI53C825A 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.

<b>DHP</b>	<b>Disable Halt on Parity Error or ATN (Target Only)</b> 5 <p>The DHP bit is only defined for target mode. When this bit is cleared, the LSI53C825A 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 LSI53C825A may transfer up to three additional bytes before halting to synchronize between internal core cells. During synchronous operation, the LSI53C825A transfers data until there are no outstanding synchronous offsets. If the LSI53C825A is receiving data, any data residing in the DMA FIFO is sent to memory before halting.</p> <p>When this bit is set, the LSI53C825A does not halt the SCSI transfer when SATN/ or a parity error is received.</p>
<b>CON</b>	<b>Connected</b> 4 <p>This bit is automatically set any time the LSI53C825A is connected to the SCSI bus as an initiator or as a target. It is set after the LSI53C825A 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 LSI53C825A is not connected to the SCSI bus.</p> <p>The CPU can force a connected or disconnected condition by setting or clearing this bit. This feature is used primarily during loopback mode.</p>
<b>RST</b>	<b>Assert SCSI RST/ Signal</b> 3 <p>Setting this bit asserts the SRST/ signal. The SRST/ output remains asserted until this bit is cleared. The 25 <math>\mu</math>s minimum assertion time defined in the SCSI specification must be timed out by the controlling microprocessor or a SCRIPTS loop.</p>
<b>AESP</b>	<b>Assert Even SCSI Parity (force bad parity)</b> 2 <p>When this bit is set, the LSI53C825A 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 LSI53C825A 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.</p>



## IARB

### Immediate Arbitration

1

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

Arbitration is retried until won. At that point, the LSI53C825A holds BSY and SEL asserted, and waits for a select or reselect sequence. The Immediate Arbitration bit is reset automatically when the selection or reselection sequence is completed, or times out.

An unexpected disconnect condition will clear IARB without 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 \(ISTAT\)](#) register. Then one of two things eventually happens:

- The Won Arbitration bit ([SCSI Status Zero \(SSTAT0\)](#), bit 2) is set. In this case, the Immediate Arbitration bit needs to be reset. This will complete the abort sequence and disconnect the LSI53C825A from the SCSI bus. If it is not acceptable to go to Bus Free phase immediately following the arbitration phase, a low level selection may be performed instead.
- The abort completes because the LSI53C825A loses arbitration. This can be detected by the Immediate Arbitration bit being cleared. 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 (0x82)**  
**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 [SCSI Control Two \(SCNTL2\)](#)) before the SCSI core expects a disconnect to occur, normally prior to sending an Abort, Abort Tag, Bus Device Reset, Clear Queue or Release Recovery message, or before deasserting SACK/ after receiving a Disconnect command or Command Complete message.
- CHM

**Chained Mode** 6

This bit determines whether or not the SCSI core is programmed for chained SCSI mode. This bit is automatically set by the Chained Block Move (CHMOV) SCRIPTS instruction and is automatically cleared by the Block Move SCRIPTS instruction (MOVE).

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 LSI53C825A 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.

<b>SLPMD</b>	<b>SLPAR Mode</b> <span style="float: right;"><b>5</b></span> If this bit is clear, the <a href="#">SCSI Longitudinal Parity (SLPAR)</a> register functions like the LSI53C825. If this bit is set, the SLPAR register reflects the high or low byte of the SLPAR word, depending on the state of <a href="#">SCSI Control Two (SCNTL2)</a> , bit 4. It also allows a seed value to be written to the <a href="#">SCSI Longitudinal Parity (SLPAR)</a> register.
<b>SLPHBEN</b>	<b>SLPAR High Byte Enable</b> <span style="float: right;"><b>4</b></span> If this bit is clear, the low byte of the SLPAR word is present in the <a href="#">SCSI Longitudinal Parity (SLPAR)</a> register. If this bit is set, the high byte of the SLPAR word is present in the <a href="#">SCSI Longitudinal Parity (SLPAR)</a> register.
<b>WSS</b>	<b>Wide SCSI Send</b> <span style="float: right;"><b>3</b></span> 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 <a href="#">SCSI Output Data Latch (SODL)</a> 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.
<b>VUE0</b>	<b>Vendor Unique Enhancements, Bit 0</b> <span style="float: right;"><b>2</b></span> 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. The default for this bit is reset.
<b>VUE1</b>	<b>Vendor Unique Enhancement, Bit 1</b> <span style="float: right;"><b>1</b></span> 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 (0x83) SCSI Control Three (SCNTL3) Read/Write

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

R

Reserved

7

## SCF[2:0]

### Synchronous Clock Conversion Factor

[6:4]

These bits select a factor by which the frequency of SCLK is divided before being presented to the synchronous SCSI control logic. 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 [SCSI Transfer \(SXFER\)](#) register for the valid combinations.

Note: For additional information on how the synchronous transfer rate is determined, refer to [Chapter 2, “Functional Description.”](#)

**EWS                      Enable Wide SCSI                      3**

When this bit is cleared, all information transfer phases are assumed to be eight bits, transmitted on SD[7:0]/ and SDP0/. When this bit is asserted, data transfers are 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.

Clearing this bit also clears the Wide SCSI Receive bit in the [SCSI Control Two \(SCNTL2\)](#) register, which indicates the presence of a valid data byte in the [SCSI Wide Residue \(SWIDE\)](#) register.

**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 [Table 4.3](#). All other combinations are reserved.

**Table 4.3      Synchronous Clock Conversion Factor**

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	Reserved	–
1	1	0	Reserved	–
1	1	1	Reserved	–

Note: It is important that these bits are set to the proper values to guarantee that the LSI53C825A meets the SCSI timings as defined by the ANSI specification.

For additional information on how the synchronous transfer rate is determined, refer to [Chapter 2, “Functional Description.”](#)

**Register: 0x04 (0x84)**  
**SCSI Chip ID (SCID)**  
**Read/Write**

7	6	5	4	3	2	1	0
R	RRE	SRE	R	ENC[3:0]			
x	0	0	x	0	0	0	0

<b>R</b>	<b>Reserved</b>	<b>7</b>
<b>RRE</b>	<b>Enable Response to Reselection</b>	<b>6</b>
	When this bit is set, the LSI53C825A is enabled to respond to bus-initiated reselection at the chip ID in the <a href="#">Response ID Zero (RESPID0)</a> and <a href="#">Response ID One (RESPID1)</a> registers. Note that the LSI53C825A does not automatically reconfigure itself to the initiator mode as a result of being reselected.	
<b>SRE</b>	<b>Enable Response to Selection</b>	<b>5</b>
	When this bit is set, the LSI53C825A is able to respond to bus-initiated selection at the chip ID in the <a href="#">Response ID Zero (RESPID0)</a> and <a href="#">Response ID One (RESPID1)</a> registers. Note that the LSI53C825A does not automatically reconfigure itself to target mode as a result of being selected.	
<b>R</b>	<b>Reserved</b>	<b>4</b>
<b>ENC[3:0]</b>	<b>Encoded Chip SCSI ID</b>	<b>[3:0]</b>
	These bits are used to store the LSI53C825A encoded SCSI ID. This is the ID which the chip asserts when arbitrating for the SCSI bus. The IDs that the LSI53C825A responds to when selected or reselected are configured in the <a href="#">Response ID Zero (RESPID0)</a> and <a href="#">Response ID One (RESPID1)</a> 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 (0x85)**  
**SCSI Transfer (SXFER)**  
Read/Write

7			5			4			0		
TP[2:0]			MO[4:0]								
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.

For additional information on how the synchronous transfer rate is determined, refer to [Chapter 2, "Functional Description."](#)

**TP[2:0] SCSI Synchronous Transfer Period [7:5]**

These bits determine the SCSI synchronous transfer period used by the LSI53C825A 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 LSI53C825A should use when transferring SCSI data is determined in the following example:

The LSI53C825A is connected to a hard disk which can transfer data at 10 Mbytes/s synchronously. The LSI53C825A SCLK is running at 40 MHz. The synchronous transfer period (SXFERP) is found as follows:

$$\text{SXFERP} = \text{Period} / \text{SSCP} + \text{ExtCC}$$

$$\text{Period} = 1 \div \text{Frequency} = 1 \div 10 \text{ Mbytes/s} = 100 \text{ ns}$$

$$\text{SSCP} = 1 \div \text{SSCF} = 1 \div 40 \text{ MHz} = 25 \text{ ns}$$

(This SCSI synchronous core clock is determined in [SCSI Control Three \(SCNTL3\)](#), bits [6:4], ExtCC = 1 if [SCSI Control One \(SCNTL1\)](#), bit 7 is asserted and the LSI53C825A is sending data. ExtCC = 0 if the LSI53C825A is receiving data.)

$$\text{SXFERP} = 100 \div 25 = 4$$

Where

<b>SXFERP</b>	Synchronous transfer period
<b>SSCP</b>	SCSI synchronous core period
<b>SSCF</b>	SCSI synchronous core frequency
<b>ExtCC</b>	Extra clock cycle of data setup



Table 4.4 and Table 4.5 show examples of possible bit combinations.

**Table 4.4 Examples of Synchronous Transfer Periods and Rates for SCSI-1**

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

**Table 4.5 Example Transfer Periods and Rates for Fast SCSI-2**

CLK (MHz)	SCSI CLK ÷ SCNTL3 Bits [6:4]	XFERP	Synch. Transfer Period (ns)	Synch. Transfer Rate (Mbytes)	Synch. Receive Rate (Mbytes)
66.67	1.5	4	90	11.11	11.11
66.67	1.5	5	112.5	8.88	11.11
50	1	4	80	12.5	12.5
50	1	5	100	10.0	12.5
40	1	4	100	10.0	10.0
37.50	1	4	106.67	9.375	9.375
33.33	1	4	120	8.33	8.33
25	1	4	160	6.25	6.25
20	1	4	200	5	5
16.67	1	4	240	4.17	4.17

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

These bits describe the maximum SCSI synchronous offset used by the LSI53C825A when transferring synchronous SCSI data in either the initiator or target mode. [Table 4.6](#) describes the possible combinations and their relationship to the synchronous data offset used by the LSI53C825A. These bits determine the LSI53C825A method of transfer for Data In and Data Out phases only; all other information transfers occur asynchronously.

**Table 4.6 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	x	x	x	1	Reserved
1	x	x	1	x	Reserved
1	x	1	x	x	Reserved
1	1	x	x	x	Reserved

**Register: 0x06 (0x86)**  
**SCSI Destination ID (SDID)**  
Read/Write

7	4	3	0
R			
ENC[3:0]			
x	x	x	x
0	0	0	0

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

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

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

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

**Register: 0x07 (0x87)**  
**General Purpose (GPREG)**  
Read/Write

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

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

**GPIO[4:0]** **General Purpose I/O** **[4:0]**

These bits are programmed through the [General Purpose Pin Control \(GPCNTL\)](#) register as inputs, outputs, or to perform special functions. As an output, these pins can be used to enable or disable external terminators. It is also possible to program these signals as live inputs and sense them through a SCRIPTS register to register Move

Instruction. GPIO[3:0] default as inputs and GPIO4 defaults as an output pin. When configured as inputs, an internal pull-down is enabled.

GPIO4 can be used to enable or disable  $V_{PP}$  the 12 V power supply to the external flash memory. This bit powers up with the power to the external memory disabled.

The GPIO[1:0] signals can also be controlled from PCI configuration register 0x35. They may be read, but not controlled, from this register.

LSI Logic software uses GPIO3 to detect a differential board. If the pin is pulled low externally, the board will be configured by SDMS software as a differential board. If it is pulled high or left floating, SDMS software will configure it as an SE board. The LSI Logic PCI to SCSI host adapters use the GPIO4 pin in the process of flashing a new SDMS software ROM.

LSI Logic software uses the GPIO0 pin to toggle SCSI device LEDs, turning on the LED whenever the LSI53C825A is on the SCSI bus. SDMS software drives this pin low to turn on the LED, or drives it high to turn off the LED.

SDMS software uses the GPIO[1:0] pins to support serial EEPROM access. When serial EEPROM access is enabled, GPIO1 is used as a clock and GPIO0 is used as data. The pins are controlled from PCI configuration register 0x35. They may be read, but not controlled, from this register.

**Register: 0x08 (0x88)**  
**SCSI First Byte Received (SFBR)**  
Read/Write

7						0	
IB							
0	0	0	0	0	0	0	0

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

This register contains the first byte received in any asynchronous information transfer phase. For example, when a LSI53C825A is operating in the initiator mode, this register contains the first byte received in the Message-In, Status, and Data-In phases.

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

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

The [SCSI First Byte Received \(SFBR\)](#) is not writable using the CPU, and therefore not by a Memory Move. However, it can be loaded using SCRIPTS Read/Write operations. To load the SFBR with a byte stored in system memory, the byte must first be moved to an intermediate LSI53C825A register (such as the SCRATCH register), and then to the SFBR.

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

**Register: 0x09 (0x89)**  
**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

<b>REQ</b>	<b>Assert SCSI REQ/ Signal</b>	<b>7</b>
<b>ACK</b>	<b>Assert SCSI ACK/ Signal</b>	<b>6</b>
<b>BSY</b>	<b>Assert SCSI BSY/ Signal</b>	<b>5</b>
<b>SEL</b>	<b>Assert SCSI SEL/ Signal</b>	<b>4</b>
<b>ATN</b>	<b>Assert SCSI ATN/ Signal</b>	<b>3</b>
<b>MSG</b>	<b>Assert SCSI MSG/ Signal</b>	<b>2</b>
<b>C/D</b>	<b>Assert SCSI C_D/ Signal</b>	<b>1</b>
<b>I/O</b>	<b>Assert SCSI I_O/ Signal</b>	<b>0</b>

This register is used primarily for diagnostic testing or programmed I/O operation. It is controlled by the SCRIPTS processor when executing SCSI SCRIPTS. [SCSI Output Control Latch \(SOCL\)](#) is used only when transferring data using programmed I/O. Some bits are set or cleared when executing SCSI SCRIPTS. Do not write to the register once the LSI53C825A starts executing normal SCSI SCRIPTS.

**Register: 0x0A (0x09)**  
**SCSI Selector ID (SSID)**  
**Read Only**

7	6	4	3	0			
VAL	R			ENID[3:0]			
0	x	x	x	0	0	0	0

<b>VAL</b>	<b>SCSI Valid</b>	<b>7</b>
	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.	
<b>R</b>	<b>Reserved</b>	<b>[6:4]</b>
<b>ENID[3:0]</b>	<b>Encoded Destination SCSI ID</b>	<b>[3:0]</b>
	Reading the <a href="#">SCSI Destination ID (SDID)</a> register immediately after the LSI53C825A 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 (0x8B)**  
**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

<b>REQ</b>	<b>Assert SCSI REQ/ Signal</b>	<b>7</b>
<b>ACK</b>	<b>Assert SCSI ACK/ Signal</b>	<b>6</b>
<b>BSY</b>	<b>Assert SCSI BSY/ Signal</b>	<b>5</b>
<b>SEL</b>	<b>Assert SCSI SEL/ Signal</b>	<b>4</b>
<b>ATN</b>	<b>Assert SCSI ATN/ Signal</b>	<b>3</b>

<b>MSG</b>	<b>Assert SCSI MSG/ Signal</b>	<b>2</b>
<b>C/D</b>	<b>Assert SCSI C_D/ Signal</b>	<b>1</b>
<b>I/O</b>	<b>Assert SCSI I_O/ Signal</b>	<b>0</b>

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

### Register: 0x0C (0x8C)

#### 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 LSI53C825A stack interrupts). The DIP bit in the [Interrupt Status \(ISTAT\)](#) register is also cleared. It is possible to mask DMA interrupt conditions individually through the [DMA Interrupt Enable \(DIEN\)](#) register.

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

<b>DFE</b>	<b>DMA FIFO Empty</b>	<b>7</b>
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.		



<b>MDPE</b>	<b>Master Data Parity Error</b> <span style="float: right;"><b>6</b></span> This bit is set when the LSI53C825A 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 <a href="#">Chip Test Four (CTEST4)</a> ).
<b>BF</b>	<b>Bus Fault</b> <span style="float: right;"><b>5</b></span> This bit is set when a PCI bus fault condition is detected. A PCI bus fault can only occur when the LSI53C825A is bus master, and is defined as a cycle that ends with a Bad Address or Target Abort Condition.
<b>ABRT</b>	<b>Aborted</b> <span style="float: right;"><b>4</b></span> 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 <a href="#">Interrupt Status (ISTAT)</a> register.
<b>SSI</b>	<b>Single Step Interrupt</b> <span style="float: right;"><b>3</b></span> If the Single Step Mode bit in the <a href="#">DMA Control (DCNTL)</a> register is set, this bit is set and an interrupt generated after successful execution of each SCRIPTS instruction.
<b>SIR</b>	<b>SCRIPTS Interrupt Instruction Received</b> <span style="float: right;"><b>2</b></span> This status bit is set whenever an interrupt instruction is evaluated as true.
<b>R</b>	<b>Reserved</b> <span style="float: right;"><b>1</b></span>
<b>IID</b>	<b>Illegal Instruction Detected</b> <span style="float: right;"><b>0</b></span> This status bit is set any time an illegal or reserved instruction opcode is detected, whether the LSI53C825A is operating in single step mode or automatically executing SCSI SCRIPTS.  Any of the following conditions during instruction execution also sets this bit: <ul style="list-style-type: none"> <li>• The LSI53C825A is executing a Wait Disconnect instruction and the SCSI REQ line is asserted without a disconnect occurring.</li> <li>• A Block Move instruction is executed with 0x000000 loaded into the <a href="#">DMA Byte Counter (DBC)</a> register, indicating there are zero bytes to move.</li> </ul>

- 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 LSI53C825A 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 and Store instruction is issued with the memory address mapped to the operating registers of the chip, not including ROM or RAM.
- A Load and Store instruction is issued when the register address is not aligned with the memory address.
- A Load and Store instruction is issued with bit 5 in the [DMA Command \(DCMD\)](#) register cleared or bits 3 or 2 set.
- A Load and Store instruction when the count value in the [DMA Byte Counter \(DBC\)](#) register is not set at 1 to 4.
- A Load and Store instruction attempts to cross a Dword boundary.
- A Memory Move instruction is executed with one of the reserved bits in the [DMA Command \(DCMD\)](#) register set.
- A Memory Move instruction is executed with the source and destination addresses not aligned.

**Register: 0x0D (0x8D)**  
**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 \(SIDL\)](#) register before being sent to the DMA FIFO and then to the host bus. The [SCSI Input Data Latch \(SIDL\)](#) register contains SCSI data received asynchronously. Synchronous data received does not flow through this register.
- 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 [SCSI Output Data Latch \(SODL\)](#) register is the interface between the DMA logic and the SCSI bus. In synchronous mode, data is transferred from the host bus to the [SCSI Output Data Latch \(SODL\)](#) 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 [SCSI Output Data Latch \(SODL\)](#) 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.

<b>AIP</b>	<b>Arbitration in Progress</b>	<b>4</b>
	Arbitration in Progress (AIP = 1) indicates that the LSI53C825A has detected a Bus Free condition, asserted BSY, and asserted its SCSI ID onto the SCSI bus.	
<b>LOA</b>	<b>Lost Arbitration</b>	<b>3</b>
	When set, LOA indicates that the LSI53C825A has detected a bus free condition, arbitrated for the SCSI bus, and lost arbitration due to another SCSI device asserting the SEL/ signal.	
<b>WOA</b>	<b>Won Arbitration</b>	<b>2</b>
	When set, WOA indicates that the LSI53C825A has detected a Bus Free condition, arbitrated for the SCSI bus and won arbitration. The arbitration mode selected in the <a href="#">SCSI Control Zero (SCNTL0)</a> register must be full arbitration and selection to set this bit.	
<b>RST/</b>	<b>SCSI RST/ Signal</b>	<b>1</b>
	This bit reports the current status of the SCSI RST/ signal, and the RST signal (bit 6) in the <a href="#">Interrupt Status (ISTAT)</a> register. This bit is not latched and may change as it is read.	
<b>SDP0/</b>	<b>SCSI SDP0/ Parity Signal</b>	<b>0</b>
	This bit represents the present state of the SCSI SDP0/ parity signal. This signal is not latched and may change as it is read.	

**Register: 0x0E (0x8E)**  
**SCSI Status One (SSTAT1)**  
Read Only

7		4		3	2	1	0
FF[3:0]				SDPOL	MSG	C/D	I/O
0	0	0	0	x	x	x	x

<b>FF[3:0]</b>	<b>FIFO Flags</b>	<b>[7:4]</b>
	These four bits, along with <a href="#">SCSI Status Two (SSTAT2)</a> , bit 4, define the number of bytes or words that currently reside in the LSI53C825A SCSI synchronous data FIFO. These bits are not latched and they will change as data moves through the FIFO.	

	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

<b>SDP0L</b>	<b>Latched SCSI Parity</b>	<b>3</b>
	This bit reflects the SCSI parity signal (SDP0/), corresponding to the data latched in the <a href="#">SCSI Input Data Latch (SIDL)</a> . It changes when a new byte is latched into the least significant byte of the <a href="#">SCSI Input Data Latch (SIDL)</a> register. This bit is active high, in other words, it is set when the parity signal is active.	
<b>MSG</b>	<b>SCSI MSG/ Signal</b>	<b>2</b>
<b>C/D</b>	<b>SCSI C_D/ Signal</b>	<b>1</b>
<b>I/O</b>	<b>SCSI I_O/ Signal</b>	<b>0</b>
	These SCSI phase status bits 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 (0x8F)**  
**SCSI Status Two (SSTAT2)**  
**Read Only**

7	6	5	4	3	2	1	0
ILF	ORF1	OLF1	FF4	SPL1	DIFF	LDSC	SDP1
0	0	0	0	0	0	0	0

- 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 \(SIDL\)](#) register before being sent to the DMA FIFO and then to the host bus. The [SCSI Input Data Latch \(SIDL\)](#) register contains SCSI data received asynchronously. Synchronous data received does not flow through this register.
- 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 [SCSI Output Data Latch \(SODL\)](#) register is the interface between the DMA logic and the SCSI bus. In synchronous mode, data is transferred from the host bus to the [SCSI Output Data Latch \(SODL\)](#) 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 [SCSI Output Data Latch \(SODL\)](#) 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.

<b>FF4</b>	<b>FIFO Flags, Bit 4</b>	<b>4</b>
	This is the most significant bit in the SCSI FIFO Flags field, with the rest of the bits in <a href="#">SCSI Status One (SSTAT1)</a> . For a complete description of this field, see the definition for <a href="#">SCSI Status One (SSTAT1)</a> , bits [7:4].	
<b>SPL1</b>	<b>Latched SCSI parity for SD[15:8]</b>	<b>3</b>
	This active HIGH bit reflects the SCSI odd parity signal corresponding to the data latched into the most significant byte in the <a href="#">SCSI Input Data Latch (SIDL)</a> register.	
<b>DIFF</b>	<b>Diffsens Mismatch</b>	<b>2</b>
	If this bit is reset, the correct cable type has been connected for the differential operation. If this bit is set, a SE cable has been connected to the device's DIFFSENS pin.	
<b>LDSC</b>	<b>Last Disconnect</b>	<b>1</b>
	This bit is used in conjunction with the Connected (CON) bit in <a href="#">SCSI Control One (SCNTL1)</a> . It allows the user to detect the case in which a target device disconnects, and then some SCSI device selects or reselects the LSI53C825A. 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 <a href="#">SCSI Control One (SCNTL1)</a> is off. This bit is cleared when a Block Move instruction is executed while the Connected bit in <a href="#">SCSI Control One (SCNTL1)</a> is on.	
<b>SDP1</b>	<b>SCSI SDP1 Parity Signal</b>	<b>0</b>
	This bit represents the present state of the SCSI SDP1/parity signal. It is unlatched and may change as it is read.	

## Registers: 0x10–0x13 (0x90–0x93)

### Data Structure Address (DSA)

Read/Write

31	DSA[31:0]																												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	0	0

#### DSA

#### Data Structure Address

[31:0]

This 32-bit register contains the base address used for all table indirect calculations. The [Data Structure Address \(DSA\)](#) 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 is preserved. The power-up value of this register is indeterminate.

## Register: 0x14 (0x94)

### Interrupt Status (ISTAT)

Read/Write

7	6	5	4	3	2	1	0
ABRT	SRST	SIGP	SEM	CON	INTF	SIP	DIP
0	0	0	0	0	0	0	0

This is the only register that is accessible by the host CPU while a LSI53C825A 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. For more information on interrupt handling refer to [Chapter 2, "Functional Description."](#)

#### ABRT

#### Abort Operation

7

Setting this bit aborts the current operation under execution by the LSI53C825A. 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 \(ISTAT\)](#) register.
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.

<b>SRST</b>	<b>Software Reset</b> <span style="float: right;"><b>6</b></span> Setting this bit resets the LSI53C825A. 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).
<b>SIGP</b>	<b>Signal Process</b> <span style="float: right;"><b>5</b></span> SIGP is a R/W bit that is writable at any time, and polled and reset using <a href="#">Chip Test Two (CTEST2)</a> . 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.
<b>SEM</b>	<b>Semaphore</b> <span style="float: right;"><b>4</b></span> The SCRIPTS processor may set this bit using a SCRIPTS register write instruction. An external processor may also set it while the LSI53C825A is executing a SCRIPTS operation. This bit enables the LSI53C825A to

notify an external processor of a predefined condition while SCRIPTS are running. The external processor may also notify the LSI53C825A 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 LSI53C825A is connected to the SCSI bus as an initiator or as a target. It is set after successfully completing selection or when the LSI53C825A responds to a bus-initiated selection or reselection. It is also set after the SCSI function wins arbitration when operating in low level mode. When this bit is cleared, the LSI53C825A 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 \(ISTAT\)](#) register is read it is 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 is not set, do not attempt to read the other chip status registers. An interrupt-on-the-fly must be cleared before servicing any other interrupts indicated by SIP or DIP.

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 LSI53C825A. 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 LSI53C825A is selected
- The LSI53C825A 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

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 LSI53C825A. 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: 0x18 (0x98)****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]**

This was a general purpose read/write register in previous LSI53C8XX family chips. Although it is still a read/write register, LSI Logic reserves the right to use these bits for future LSI53C8XX family enhancements.

**Register: 0x19 (0x99)****Chip Test One (CTEST1)****Read Only**

7				0			
FMT[3:0]				FFL[3:0]			
1	1	1	1	0	0	0	0

**FMT[3:0]****Byte Empty in DMA FIFO****[7:4]**

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 is 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.

**FFL[3:0]****Byte Full in DMA FIFO****[3:0]**

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 (0x9A)**  
**Chip Test Two (CTEST2)**  
**Read/Write**

7	6	5	4	3	2	1	0
DDIR	SIGP	CIO	CM	SRTCH	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 \(ISTAT\)](#) register (bit 5). The SIGP bit is used to signal a running SCRIPTS instruction. When this register is read, the SIGP bit in the 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.

Note: Bits 4 and 5 may be set if the chip is dual-mapped.

**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 dual-mapped.

**SRTCH** **SCRATCHA/B Operation** **3**  
This bit controls the operation of the [Scratch Register A \(SCRATCHA\)](#) and [Scratch Register B \(SCRATCHB\)](#) registers. When it is set, SCRATCHB contains the RAM base address value from the PCI configuration RAM Base Address register. This is the base address for the 4 Kbyte internal RAM. In addition, the [Scratch Register A \(SCRATCHA\)](#) register displays the memory-mapped based address of the chip operating registers. When this

bit is clear, the [Scratch Register A \(SCRATCHA\)](#) and [Scratch Register B \(SCRATCHB\)](#) 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.

<b>TEOP</b>	<b>SCSI True End of Process</b>	<b>2</b>
	This bit indicates the status of the LSI53C825A internal TEOP signal. The TEOP signal acknowledges the completion of a transfer through the SCSI portion of the LSI53C825A. When this bit is set, TEOP is active. When this bit is cleared, TEOP is inactive.	
<b>DREQ</b>	<b>Data Request Status</b>	<b>1</b>
	This bit indicates the status of the LSI53C825A internal Data Request signal (DREQ). When this bit is set, DREQ is active. When this bit is cleared, DREQ is inactive.	
<b>DACK</b>	<b>Data Acknowledge Status</b>	<b>0</b>
	This bit indicates the status of the LSI53C825A internal Data Acknowledge signal (DACK/). When this bit is set, DACK/ is inactive. When this bit is cleared, DACK/ is active.	

**Register: 0x1B (0x9B)**  
**Chip Test Three (CTEST3)**  
**Read/Write**

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

<b>V[3:0]</b>	<b>Chip Revision Level</b>	<b>[7:4]</b>
	These bits identify the chip revision level for software purposes. It should have the same value as the lower nibble of the PCI <a href="#">Revision ID</a> register, at address 0x08 in the configuration space.	
<b>FLF</b>	<b>Flush DMA FIFO</b>	<b>3</b>
	When this bit is set, data residing in the DMA FIFO is transferred to memory, starting at the address in the <a href="#">DMA</a>	

[Next Address \(DNAD\)](#) register. The internal DMAWR signal, controlled by the [Chip Test Five \(CTEST5\)](#) register, determines the direction of the transfer. This bit is not self-clearing; clear it once the data is successfully transferred by the LSI53C825A.

Note: Polling of FIFO flags is allowed during flush operations.

<b>CLF</b>	<b>Clear DMA FIFO</b>	<b>2</b>
	When this bit is set, all data pointers for the DMA FIFO are cleared. Any data in the FIFO is lost. After the LSI53C825A 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.

<b>FM</b>	<b>Fetch Pin Mode</b>	<b>1</b>
	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.	

<b>WRIE</b>	<b>Write and Invalidate Enable</b>	<b>0</b>
	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 <a href="#">Command</a> register must also be set in order for the chip to generate Write and Invalidate commands.	

## Registers: 0x1C–0x1F (0x9C–0x9F)

Temporary (TEMP)

Read/Write

31																															0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
TEMP																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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

**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 LSI53C825A 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 (0xA0)

DMA FIFO (DFIFO)

Read/Write

7							0
BO							
x	0	0	0	0	0	0	0

**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:

Step 1. If the DMA FIFO size is set to 88 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.

If the DMA FIFO size is set to 536 bytes (using bit 5 of the [Chip Test Five \(CTEST5\)](#) register), 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 [Chip Test Five \(CTEST5\)](#) register and bits [7:0] of the [DMA FIFO \(DFIFO\)](#) register.

Step 2. If the DMA FIFO size is set to 88 bytes, AND the result with 0x7F for a byte count between zero and 64. If the DMA FIFO size is set to 536 bytes, AND the result with 0x3FF for a byte count between zero and 536.

Note: If trying to calculate the total number of bytes in both the DMA FIFO and SCSI logic, see [Section 2.4.8.1, “Data Paths,”](#) in [Chapter 2, “Functional Description.”](#)

**Register: 0x21 (0xA1)**  
**Chip Test Four (CTEST4)**  
**Read/Write**

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

**BDIS**

**Burst Disable**

**7**

When set, this bit causes the LSI53C825A 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.

<b>ZMOD</b>	<b>High Impedance Mode</b> <span style="float: right;"><b>6</b></span> Setting this bit causes the LSI53C825A to place all output and bidirectional pins into a high impedance state. In order to read data out of the LSI53C825A, this bit must be cleared. This bit is intended for board level testing only. Do not set this bit during normal system operation.
<b>ZSD</b>	<b>SCSI Data High Impedance</b> <span style="float: right;"><b>5</b></span> Setting this bit causes the LSI53C825A 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.
<b>SRTM</b>	<b>Shadow Register Test Mode</b> <span style="float: right;"><b>4</b></span> 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 <a href="#">Temporary (TEMP)</a> and <a href="#">Data Structure Address (DSA)</a> 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 <a href="#">Data Structure Address (DSA)</a> and <a href="#">Temporary (TEMP)</a> 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.
<b>MPEE</b>	<b>Master Parity Error Enable</b> <span style="float: right;"><b>3</b></span> Setting this bit enables parity checking during master data phases. A parity error during a bus master read is detected by the LSI53C825A. A parity error during a bus master write is detected by the target, and the LSI53C825A is informed of the error by the PERR/ pin being asserted by the target. When this bit is cleared, the LSI53C825A does not interrupt if a master parity error occurs. This bit is cleared at power-up.

**FBL[2:0]**      **FIFO Byte Control**      **[2:0]**

FBL3	FBL2	FBL1	FBL0	DMA FIFO Byte Lane	Pins
0	x	x	x	Disabled	N/A
1	0	0	0	0	D[7:0]
1	0	0	1	1	D[15:8]
1	0	1	0	2	D[23:16]
1	0	1	1	3	D[31:24]

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 (0xA2)

### 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	x	x	x

#### **ADCK**      **Clock Address Incrementor**      **7**

Setting this bit increments the address pointer contained in the [DMA Next Address \(DNAD\)](#) register. The [DMA Next Address \(DNAD\)](#) register is incremented based on the DNAD contents and the current [DMA Byte Counter \(DBC\)](#) value. This bit automatically clears itself after incrementing the [DMA Next Address \(DNAD\)](#) register.

#### **BBCK**      **Clock Byte Counter**      **6**

Setting this bit decrements the byte count contained in the 24-bit [DMA Byte Counter \(DBC\)](#) register. It is decremented based on the DBC contents and the current [DMA Next Address \(DNAD\)](#) value. This bit automatically

clears itself after decrementing the [DMA Byte Counter \(DBC\)](#) register.

<b>DFS</b>	<b>DMA FIFO Size</b> <span style="float: right;"><b>5</b></span> This bit controls the size of the DMA FIFO. When clear, the DMA FIFO appears as only 88 bytes deep. When set, the DMA FIFO size increases to 536 bytes. Using an 88-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.
<b>MASR</b>	<b>Master Control for Set or Reset Pulses</b> <span style="float: right;"><b>4</b></span> 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.
<b>DDIR</b>	<b>DMA Direction</b> <span style="float: right;"><b>3</b></span> 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 DMAWR signal indicates that data is transferred from the SCSI bus to the host bus. Deasserting the DMAWR signal transfers data from the host bus to the SCSI bus.
<b>BL2</b>	<b>Burst Length Bit 2</b> <span style="float: right;"><b>2</b></span> This bit works with bits 6 and 7 (BL[1:0]) in the <a href="#">DMA Mode (DMODE)</a> , <a href="#">0x38 (0xB8)</a> register to determine the burst length. For complete definitions of this field, refer to the descriptions of <a href="#">DMA Mode (DMODE)</a> , bits 6 and 7. This bit is disabled if an 88-byte FIFO is selected by clearing the DMA FIFO Size bit.
<b>BO[9:8]</b>	<b>DMA FIFO Byte Offset Counter, Bits [9:8]</b> <span style="float: right;"><b>[1:0]</b></span> These are the upper two bits of the DFBOC. The DFBOC consists of these bits and the <a href="#">DMA FIFO (DFIFO)</a> , bits [7:0].

**Register: 0x23 (0xA3)**  
**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 [Chip Test Four \(CTEST4\)](#) 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 to this register only when testing the DMA FIFO using the [Chip Test Four \(CTEST4\)](#) register. Writing to this register while the test mode is not enabled produces unexpected results.

**Register: 0x24–0x26 (0xA4–0xA6)**  
**DMA Byte Counter (DBC)**  
Read/Write

23																									0
DBC																									
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	

**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 LSI53C825A. 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 0xFFFFFFFF. If the instruction is a Block Move and a value of 0x000000 is loaded into the [DMA Byte Counter \(DBC\)](#) register, an illegal instruction interrupt occurs if the LSI53C825A is not in the target mode, Command phase.

The [DMA Byte Counter \(DBC\)](#) register is also used to hold the least significant 24 bits of the first Dword of a SCRIPTS fetch, and to hold the offset value during table indirect I/O SCRIPTS. The power-up value of this register is indeterminate.

**Register: 0x27 (0xA7)**  
**DMA Command (DCMD)**  
**Read/Write**

7				0			
DCMD							
x	x	x	x	x	x	x	x

**DCMD** **DMA Command** **[7:0]**  
This 8-bit register determines the instruction for the LSI53C825A to execute. This register has a different format for each instruction. For a complete description see [Chapter 5, “SCSI SCRIPTS Instruction Set.”](#)

**Register: 0x28–0x2B (0xA8–0xAB)**  
**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.

This register should not be used to determine data addresses during a Phase Mismatch interrupt, as its value is not always correct for this use. The [DMA Byte Counter \(DBC\)](#), [DMA FIFO \(DFIFO\)](#), and [DMA SCRIPTS Pointer Save \(DSPS\)](#) registers should be used to calculate residual byte counts and addresses as described in the Data Paths section in [Chapter 2, “Functional Description.”](#)

## Registers: 0x2C–0x2F (0xAC–0xAF)

### DMA SCRIPTS Pointer (DSP)

Read/Write

31	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	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 the single step mode, there is a single step interrupt after each instruction is executed. The [DMA SCRIPTS Pointer \(DSP\)](#) register does not need to be written with the next address, 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 (0xB0–0xB3)

### DMA SCRIPTS Pointer Save (DSPS)

Read/Write

31																													0				
DSPS																																	
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	x	x	x	x

**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 (0xB4–0xB7)

### Scratch Register A (SCRATCHA)

Read/Write

31	SCRATCHA																												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

#### 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 LSI53C825A cannot fetch SCRIPTS instructions from this location. When bit 3 in the [Chip Test Two \(CTEST2\)](#) register is set, this register contains the memory mapped base address of the operating registers. Setting [Chip Test Two \(CTEST2\)](#), bit 3 only causes the base address to appear in this register; any information that was previously in the register will remain intact. Any writes to this register while [Chip Test Two \(CTEST2\)](#), bit 3 is set will pass through to the actual [Scratch Register A \(SCRATCHA\)](#) register. The power-up value of this register is indeterminate.

## Register: 0x38 (0xB8)

### DMA Mode (DMODE)

Read/Write

7	6	5	4	3	2	1	0
BL[1:0]		SIOM	DIOM	ER	ERMP	BOF	MAN
0	0	0	0	0	0	0	0

#### BL[1:0]      Burst Length      [7:6]

These bits control the maximum number of Dwords transferred per bus ownership, regardless of whether the transfers are back-to-back, burst, or a combination of both. This value is also independent of the width (64 or 32 bits) of the data transfer on the PCI bus. The LSI53C825A 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 LSI53C825A inserts a “fairness delay” of four CLKs between burst transfers (as set in BL[1:0]) 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.

<b>BL2</b> <b>(CTEST5 bit 2)</b>	<b>BL1</b>	<b>BL0</b>	<b>Burst Length</b>
0	0	0	2-transfer burst
0	0	1	4-transfer burst
0	1	0	8-transfer burst
0	1	1	16-transfer burst
1	0	0	32-transfer burst <sup>1</sup>
1	0	1	64-transfer burst <sup>1</sup>
1	1	0	128-transfer burst <sup>1</sup>
1	1	1	Reserved

1. Only valid if the FIFO size is set to 536 bytes.

## 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 a LSI53C825A is I/O mapped. Bits 4 and 5 of the [Chip Test Two \(CTEST2\)](#) register are used to determine the configuration status of the LSI53C825A.

## 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 a LSI53C825A is I/O mapped. Bits 4 and 5 of the [Chip Test Two \(CTEST2\)](#) register are used to determine the configuration status of the LSI53C825A.

<b>ERL</b>	<b>Enable Read Line</b> <span style="float: right;"><b>3</b></span> This bit enables a PCI Read Line command. If the PCI cache mode is enabled by setting bits in the PCI <a href="#">Cache Line Size</a> register, this chip issues a Read Line command on all read cycles if other conditions are met. For more information on these conditions, refer to <a href="#">Chapter 3, “Signal Descriptions.”</a>
<b>ERMP</b>	<b>Enable Read Multiple</b> <span style="float: right;"><b>2</b></span> This bit, when set, causes Read Multiple commands to be issued on the PCI bus after certain conditions have been met. These conditions are described in <a href="#">Chapter 3, “Signal Descriptions.”</a>
<b>BOF</b>	<b>Burst Opcode Fetch Enable</b> <span style="float: right;"><b>1</b></span> Setting this bit causes the LSI53C825A 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. This bit has no effect if SCRIPTS instruction prefetching is enabled.
<b>MAN</b>	<b>Manual Start Mode</b> <span style="float: right;"><b>0</b></span> Setting this bit prevents the LSI53C825A from automatically fetching and executing SCSI SCRIPTS when the <a href="#">DMA SCRIPTS Pointer (DSP)</a> register is written. When this bit is set, the Start DMA bit in the <a href="#">DMA Control (DCNTL)</a> register must be set to begin SCRIPTS execution. Clearing this bit causes the LSI53C825A to automatically begin fetching and executing SCSI SCRIPTS when the <a href="#">DMA SCRIPTS Pointer (DSP)</a> register is written. This bit normally is not used for SCSI SCRIPTS operations.

**Register: 0x39 (0xB9)**  
**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

<b>R</b>	<b>Reserved</b>	<b>7</b>
<b>MDPE</b>	<b>Master Data Parity Error</b>	<b>6</b>
<b>BF</b>	<b>Bus Fault</b>	<b>5</b>
<b>ABRT</b>	<b>Aborted</b>	<b>4</b>
<b>SSI</b>	<b>Single Step Interrupt</b>	<b>3</b>
<b>SIR</b>	<b>SCRIPTS Interrupt Instruction Received</b>	<b>2</b>
<b>R</b>	<b>Reserved</b>	<b>1</b>
<b>IID</b>	<b>Illegal Instruction Detected</b>	<b>0</b>

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 [DMA Status \(DSTAT\)](#) register. Masking an interrupt does not prevent setting the [Interrupt Status \(ISTAT\)](#) 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 \(ISTAT\)](#) SIP or DIP bit is set.)

The LSI53C825A IRQ/ output is latched; once asserted, it remains asserted until the interrupt is cleared by reading the appropriate status register. Masking an interrupt after the IRQ/ output is asserted does not cause IRQ/ to be deasserted.

For more information on interrupts, see [Chapter 2, "Functional Description."](#)

**Register: 0x3A (0xBA)**  
**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 (0xBB)**  
**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 LSI53C825A 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 LSI53C825A monitors the cache line size using the [DMA Mode \(DMODE\)](#) register.

**PFF**      **Prefetch Flush** **6**

Setting this bit causes the prefetch unit to flush its contents. This bit clears after the flush is complete.

**PFEN**      **Prefetch Enable** **5**

Setting this bit enables the prefetch unit if the burst size is equal to or greater than four. For more information on SCRIPTS instruction prefetching, see [Chapter 2, "Functional Description."](#)

<b>SSM</b>	<b>Single Step Mode</b> <span style="float: right;"><b>4</b></span> Setting this bit causes the LSI53C825A to stop after executing each SCRIPTS instruction, and generate a single step interrupt. When this bit is cleared the LSI53C825A does not stop after each instruction. It continues fetching and executing instructions until an interrupt condition occurs. For normal SCSI SCRIPTS operation, keep this bit cleared. To restart the LSI53C825A after it generates a SCRIPTS Step interrupt, read the <a href="#">Interrupt Status (ISTAT)</a> and <a href="#">DMA Status (DSTAT)</a> registers to recognize and clear the interrupt. Then set the START DMA bit in this register.
<b>IRQM</b>	<b>IRQ Mode</b> <span style="float: right;"><b>3</b></span> 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.
<b>STD</b>	<b>Start DMA Operation</b> <span style="float: right;"><b>2</b></span> The LSI53C825A fetches a SCSI SCRIPTS instruction from the address contained in the <a href="#">DMA SCRIPTS Pointer (DSP)</a> register when this bit is set. This bit is required if the LSI53C825A is in one of the following modes: <ul style="list-style-type: none"> <li>• Manual start mode – Bit 0 in the <a href="#">DMA Mode (DMODE)</a> register is set</li> <li>• Single step mode – Bit 4 in the <a href="#">DMA Control (DCNTL)</a> register is set</li> </ul> <p>When the LSI53C825A 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 LSI53C825A is in single step mode, set the Start DMA bit to restart execution of SCRIPTS after a single step interrupt.</p>
<b>IRQD</b>	<b>IRQ Disable</b> <span style="float: right;"><b>1</b></span> Setting this bit disables the IRQ pin. Clearing the bit enables normal operation. As with any other register other than <a href="#">Interrupt Status (ISTAT)</a> , 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 <a href="#">Chapter 2, "Functional Description."</a>

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

When the COM bit is cleared, the LSI53C825A 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.

When this bit is set, the ID is stored only in the [SCSI Destination ID \(SDID\)](#) register, protecting the SFBR from being overwritten if a selection/reselection occurs during a DMA register-to-register operation. This bit is not affected by a software reset.

**Registers: 0x3C–0x3F (0xBC–0xBF)****Adder Sum Output (ADDER)****Read Only**

31	ADDER																														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	x	x

**ADDER****Adder Sum Output****[31:0]**

This register contains the output of the internal adder, and is used primarily for test purposes. The power-up value for this register is indeterminate.

**Register: 0x40 (0xC0)****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.”](#)

<b>M/A</b>	<b>SCSI Phase Mismatch - Initiator Mode; SCSI ATN Condition - Target Mode</b> <b>7</b> 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 <a href="#">SCSI Output Control Latch (SOCL)</a> register. This expected phase is automatically written by SCSI SCRIPTS. In the target mode, this bit is set when the initiator asserts SATN/. See the Disable Halt on Parity Error or SATN/ Condition bit in the <a href="#">SCSI Control One (SCNTL1)</a> register for more information on when this status is actually raised.
<b>CMP</b>	<b>Function Complete</b> <b>6</b> Indicates full arbitration and selection sequence is completed.
<b>SEL</b>	<b>Selected</b> <b>5</b> Indicates the LSI53C825A is selected by a SCSI initiator device. Set the Enable Response to Selection bit in the <a href="#">SCSI Chip ID (SCID)</a> register for this to occur.
<b>RSL</b>	<b>Reselected</b> <b>4</b> Indicates the LSI53C825A is reselected by a SCSI target device. Set the Enable Response to Reselection bit in the <a href="#">SCSI Chip ID (SCID)</a> register for this to occur.
<b>SGE</b>	<b>SCSI Gross Error</b> <b>3</b> The following conditions are considered SCSI Gross Errors: <ul style="list-style-type: none"> <li>• Data underflow – reading the SCSI FIFO when no data is present.</li> <li>• Data overflow – writing to the SCSI FIFO while it is full.</li> <li>• Offset underflow – receiving a SACK/ pulse in the target mode before the corresponding SREQ/ is sent.</li> <li>• Offset overflow – receiving a SREQ/ pulse in the initiator mode, and exceeding the maximum offset (defined by the MO[3:0] bits in the <a href="#">SCSI Transfer (SXFER)</a> register).</li> <li>• A phase change in the initiator mode, with an outstanding SREQ/SACK offset.</li> </ul>



- Residual data in SCSI FIFO – starting a transfer other than synchronous data receive with data left in the SCSI synchronous receive FIFO.

<b>UDC</b>	<b>Unexpected Disconnect</b>	<b>2</b>
	This condition only occurs in the initiator mode. It happens when the target to which the LSI53C825A is connected disconnects from the SCSI bus unexpectedly. See the SCSI Disconnect Unexpected bit in the <a href="#">SCSI Control Two (SCNTL2)</a> register for more information on expected versus unexpected disconnects. Any disconnect in the low level mode causes this condition.	
<b>RST</b>	<b>SCSI Reset Condition</b>	<b>1</b>
	Indicates assertion of the SRST/ signal by the LSI53C825A or any other SCSI device. This condition is edge-triggered, so multiple interrupts cannot occur because of a single SRST/ pulse.	
<b>PAR</b>	<b>SCSI Parity Error</b>	<b>0</b>
	Indicates detection by the LSI53C825A of a parity error while receiving or sending SCSI data. See the Disable Halt on Parity Error or SATN/ Condition bits in the <a href="#">SCSI Control One (SCNTL1)</a> register for more information on when this condition is actually raised.	

### Register: 0x41 (0xC1)

#### SCSI Interrupt Enable One (SIEN1)

Read/Write

7	6	5	4	3	2	1	0
R					STO	GEN	HTH
x	x	x	x	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.”](#)

<b>R</b>	<b>Reserved</b>	<b>[7:3]</b>
<b>STO</b>	<b>Selection or Reselection Time-out</b> The SCSI device which the LSI53C825A is attempting to select or reselect does not respond within the programmed time-out period. See the description of the <a href="#">SCSI Timer Zero (STIME0)</a> register bits [3:0] for more information on the time-out timer.	<b>2</b>
<b>GEN</b>	<b>General Purpose Timer Expired</b> The general purpose timer is expired. The time measured is the time between enabling and disabling of the timer. See the description of the <a href="#">SCSI Timer One (STIME1)</a> register, bits [3:0], for more information on the general purpose timer.	<b>1</b>
<b>HTH</b>	<b>Handshake-to-Handshake Timer Expired</b> 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 <a href="#">SCSI Timer Zero (STIME0)</a> register, bits [7:4], for more information on the handshake-to-handshake timer.	<b>0</b>

**Register: 0x42 (0xC2)**  
**SCSI Interrupt Status Zero (SIST0)**  
**Read Only**

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

Reading the [SCSI Interrupt Status Zero \(SIST0\)](#) register returns the status of the various interrupt conditions, whether they are enabled in the [SCSI Interrupt Enable Zero \(SIEN0\)](#) register or not. Each bit set indicates occurrence of the corresponding condition. Reading the SIST0 clears the interrupt status.

Reading this register clears any bits that are set at the time the register is read, but does not necessarily clear the register because additional interrupts may be pending (the LSI53C825A stack interrupts). SCSI interrupt conditions are individually masked through the [SCSI Interrupt Enable Zero \(SIEN0\)](#) register.

When performing consecutive 8-bit reads of the [DMA Status \(DSTAT\)](#), [SCSI Interrupt Status Zero \(SIST0\)](#), and [SCSI Interrupt Status One \(SIST1\)](#) registers (in any order), insert a delay equivalent to 12 clock periods between the reads to ensure the interrupts clear properly. Also, if reading the registers when both the [Interrupt Status \(ISTAT\)](#) SIP and DIP bits may not be set, read the [SCSI Interrupt Status Zero \(SIST0\)](#) and [SCSI Interrupt Status One \(SIST1\)](#) registers before the [DMA Status \(DSTAT\)](#) register to avoid missing a SCSI interrupt. For more information on interrupts refer to [Chapter 2, “Functional Description.”](#)

<b>M/A</b>	<b>Initiator Mode: Phase Mismatch; Target Mode: SATN/ Active</b>	<b>7</b>
	In the initiator mode, this bit is set if the SCSI phase asserted by the target does not match the instruction. The phase is sampled when SREQ/ is asserted by the target. In the target mode, this bit is set when the SATN/ signal is asserted by the initiator.	
<b>CMP</b>	<b>Function Complete</b>	<b>6</b>
	This bit is set when an arbitration only or full arbitration sequence is completed.	
<b>SEL</b>	<b>Selected</b>	<b>5</b>
	This bit is set when the LSI53C825A is selected by another SCSI device. The Enable Response to Selection bit must be set in the <a href="#">SCSI Chip ID (SCID)</a> register (and the <a href="#">Response ID Zero (RESPID0)</a> and <a href="#">Response ID One (RESPID1)</a> registers must hold the chip's ID) for the LSI53C825A to respond to selection attempts.	
<b>RSL</b>	<b>Reselected</b>	<b>4</b>
	This bit is set when the LSI53C825A is reselected by another SCSI device. The Enable Response to Reselection bit must be set in the <a href="#">SCSI Chip ID (SCID)</a> register (and the <a href="#">Response ID Zero (RESPID0)</a> and <a href="#">Response ID One (RESPID1)</a> registers must hold the chip's ID) for the LSI53C825A to respond to reselection attempts.	
<b>SGE</b>	<b>SCSI Gross Error</b>	<b>3</b>
	This bit is set when the LSI53C825A encounters a SCSI Gross Error Condition. The following conditions can result in a SCSI Gross Error Condition:	
	<ul style="list-style-type: none"> <li>• Data Underflow – reading the SCSI FIFO when no data is present.</li> </ul>	

- Data Overflow – writing too many bytes to the SCSI FIFO, or the synchronous offset causes overwriting the SCSI FIFO.
- Offset Underflow – the LSI53C825A is operating in the 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 LSI53C825A 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 LSI53C825A is operating in the initiator mode and the target device unexpectedly disconnects from the SCSI bus. This bit is only valid when the LSI53C825A operates in the initiator mode. When the SCSI function operates in the low level mode, any disconnect causes an interrupt, even a valid SCSI disconnect. This bit is also set if a selection time-out occurs (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 LSI53C825A detects an active SRST/ signal, whether the reset is generated external to the chip or caused by the Assert SRST/ bit in the [SCSI Control One \(SCNTL1\)](#) register. This SCSI reset detection logic is edge-sensitive, so that multiple interrupts are not generated for a single assertion of the SRST/ signal.

## PAR

### Parity Error

0

This bit is set when the LSI53C825A 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 LSI53C825A always generates parity when sending SCSI data.

**Register: 0x43 (0xC3)**  
**SCSI Interrupt Status One (SIST1)**  
**Read Only**

7					3	2	1	0
R					STO	GEN	HTH	
x	x	x	x	x	0	0	0	

Reading the [SCSI Interrupt Status One \(SIST1\)](#) 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	[7:3]
STO	<b>Selection or Reselection Time-Out</b>	<b>2</b>
	The SCSI device which the LSI53C825A is attempting to select or reselect does not respond within the programmed time-out period. See the description of the <a href="#">SCSI Timer Zero (STIME0)</a> register, bits [3:0], for more information on the time-out timer.	
GEN	<b>General Purpose Timer Expired</b>	<b>1</b>
	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 <a href="#">SCSI Timer One (STIME1)</a> register, bits [3:0], for more information on the general purpose timer.	
HTH	<b>Handshake-to-Handshake Timer Expired</b>	<b>0</b>
	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 <a href="#">SCSI Timer Zero (STIME0)</a> register, bits [7:4], for more information on the handshake-to-handshake timer.	

**Register: 0x44 (0xC4)**  
**SCSI Longitudinal Parity (SLPAR)**  
**Read/Write**

7				0			
SLPAR							
x	x	x	x	x	x	x	x

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

The [SCSI Longitudinal Parity \(SLPAR\)](#) register consists of two multiplexed bytes; other register bit settings determine what is displayed at this memory location at any given time. When bit 5 in the [SCSI Control Two \(SCNTL2\)](#) (SLPMD) register is cleared, the chip XORs the high and low bytes of the [SCSI Longitudinal Parity \(SLPAR\)](#) register together to give a single-byte value which is displayed in the [SCSI Longitudinal Parity \(SLPAR\)](#) register. If the SLPMD bit is set, then the [SCSI Longitudinal Parity \(SLPAR\)](#) register shows either the high byte or the low byte of the SLPAR word. The SLPAR High Byte Enable bit, [SCSI Control Two \(SCNTL2\)](#), bit 4, determines which byte of the [SCSI Longitudinal Parity \(SLPAR\)](#) register is visible on the [SCSI Longitudinal Parity \(SLPAR\)](#) register at any given time. If this bit is cleared, the [SCSI Longitudinal Parity \(SLPAR\)](#) register contains the low byte of the SLPAR word; if it is set, the [SCSI Longitudinal Parity \(SLPAR\)](#) register contains the high byte of the SLPAR word.

This register performs a bitwise 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 >>> 10010110
4. 10010110	00000000

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

The [SCSI Longitudinal Parity \(SLPAR\)](#) register is also used to generate the check bytes for SCSI send operations. If the [SCSI Longitudinal Parity \(SLPAR\)](#) 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.

**Register: 0x45 (0xC5)**  
**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 (0xC6)**  
**Memory Access Control (MACNTL)**  
**Read/Write**

7	6	5	4	3	2	1	0
TYP[3:0]				DWR	DRD	PSCPT	SCPTS
0	1	1	0	0	0	0	0

**TYP[3:0]      Chip Type      [7:4]**

These bits identify the chip type for software purposes. This technical manual applies to devices that have these bits set to 0x06.

Bits 3 through 0 of this register are used to determine if an external bus master access is to local or far memory. When bits 3 through 0 are set, the corresponding access is considered local and the MAC/\_TESTOUT pin is driven high. When these bits are clear, the corresponding access is to far memory and the MAC/\_TESTOUT pin is driven low. This function is enabled after a Transfer Control SCRIPTS instruction is executed.

**DWR      DataWR      3**

This bit is used to define if a data write is considered local memory access.

**DRD      DataRD      2**

This bit is used to define if data write is considered 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 local memory access.

**SCPTS      SCRIPTS      0**

This bit is used to define if a SCRIPTS fetch is considered local memory access.



**Register: 0x47 (0xC7)**  
**General Purpose Pin Control (GPCNTL)**  
Read/Write

7	6	5	4	2	1	0
ME	FE	R	GPIO[4:2]			GPIO[1:0]
0	0	x	0	1	1	1

This register is used to determine if the pins controlled by the [General Purpose \(GPREG\)](#) register are inputs or outputs. Bits [4:0] in GPCNTL correspond to bits [4:0] in the [General Purpose \(GPREG\)](#) register. When the bits are enabled as inputs, an internal pull-up is also enabled.

<b>ME</b>	<b>Master Enable</b>	<b>7</b>
	The internal bus master signal is presented on GPIO1 if this bit is set, regardless of the state of bit 1 (GPIO1_EN).	
<b>FE</b>	<b>Fetch Enable</b>	<b>6</b>
	The internal opcode fetch signal is presented on GPIO0 if this bit is set, regardless of the state of bit 0 (GPIO0_EN).	
<b>R</b>	<b>Reserved</b>	<b>5</b>
<b>GPIO[4:2]</b>	<b>GPIO4_EN–GPIO2_EN (GPIO Enable)</b>	<b>[4:2]</b>
	General purpose control bits, corresponding to bits [4:2] in the <a href="#">General Purpose (GPREG)</a> register and pins 60, 59, and 57. GPIO4 powers up as a general purpose output, and GPIO[3:2] power-up as general purpose inputs.	
<b>GPIO[1:0]</b>	<b>GPIO1_EN–GPIO0_EN (GPIO Enable)</b>	<b>[1:0]</b>
	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 (0xC8)**  
**SCSI Timer Zero (STIME0)**  
Read/Write

7				4		3		0	
HTH[3:0]						SEL[3:0]			
0	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 the 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. [Table 4.7](#) 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."](#)

**Table 4.7     Timeout Periods**

HTH[7:4], SEL[3:0], GEN[3:0] <sup>1</sup>	Minimum Timeout (40 or 160 MHz)	Minimum Timeout (50 MHz)
0000	Disabled	Disabled
0001	125 $\mu$ s	100 $\mu$ s
0010	250 $\mu$ s	200 $\mu$ s
0011	500 $\mu$ s	400 $\mu$ s
0100	1 ms	800 $\mu$ s
0101	2 ms	1.6 ms
0110	4 ms	3.2 ms
0111	8 ms	6.4 ms
1000	16 ms	12.8 ms
1001	32 ms	25.6 ms
1010	64 ms	51.2 ms
1011	128 ms	102.4 ms
1100	256 ms	204.8 ms
1101	512 ms	409.6 ms
1110	1.024 s	819.2 ms
1111	2.048 s	1.6384 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.

**SEL[3:0]****Selection Time-Out****[3:0]**

These bits select the SCSI selection/reselection time-out period. When this timing (plus the 200  $\mu$ s 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 (0xC9)**  
**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

<b>R</b>	<b>Reserved</b>	<b>7</b>
<b>HTHBA</b>	<b>Handshake-to-Handshake Timer Bus Activity Enable</b>	<b>6</b>
	Setting this bit causes this timer to begin testing for SCSI REQ/ and ACK/ activity as soon as SBSY/ is asserted, regardless of the agents participating in the transfer.	
<b>GENSF</b>	<b>General Purpose Timer Scale Factor</b>	<b>5</b>
	Setting this bit causes this timer to shift by a factor of 16. See <a href="#">Table 4.8</a> for Timeout Periods, 50 MHz clock.	

**Table 4.8 Timeout Periods, 50 MHz Clock**

HTH[7:4], SEL[3:0], GEN[3:0] <sup>1</sup>	Minimum Timeout (50 MHz Clock) <sup>2</sup>	
	GENSF = 0	GENSF = 1
0000	Disabled	Disabled
0001	100 $\mu$ s	1.6 ms
0010	200 $\mu$ s	3.2 ms
0011	400 $\mu$ s	6.4 ms
0100	800 $\mu$ s	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

**Table 4.8 Timeout Periods, 50 MHz Clock (Cont.)**

HTH[7:4], SEL[3:0], GEN[3:0] <sup>1</sup>	Minimum Timeout (50 MHz Clock) <sup>2</sup>	
	GENSF = 0	GENSF = 1
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.
2. 50 MHz clock is not supported for Ultra2 SCSI operation.

**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. See [Chapter 2, “Functional Description,”](#) for an explanation of how interrupts are generated when the timers expire.

**Register: 0x4A (0xCA)**  
**Response ID Zero (RESPID0)**  
Read/Write

7							0
ID							
x	x	x	x	x	x	x	x

**RESPID0**      **Response ID Zero**      **[7:0]**

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 [Response ID One \(RESPID1\)](#) 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 [Response ID One \(RESPID1\)](#) and [Response ID Zero \(RESPID0\)](#) registers. However, the chip can arbitrate with only one ID value in the [SCSI Chip ID \(SCID\)](#) register.

**Register: 0x4B (0xCB)**  
**Response ID One (RESPID1)**  
Read/Write

15						8	
ID							
x	x	x	x	x	x	x	x

**RESPID1**      **Response ID One**      **[15:8]**

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 [Response ID One \(RESPID1\)](#) 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 [Response ID One](#)

(RESPID1) and [Response ID Zero \(RESPID0\)](#) registers. However, the chip can arbitrate with only one ID value in the [SCSI Chip ID \(SCID\)](#) register.

## Register: 0x4C (0xCC)

### SCSI Test Zero (STEST0)

Read Only

7	6	5	4	3	2	1	0
SSAID[3:0]				SLT	ART	SOZ	SOM
0	0	0	0	0	x	1	1

#### SSAID[3:0] SCSI Selected As ID [7:4]

These bits contain the encoded value of the SCSI ID that the LSI53C825A is selected or reselected as during a SCSI selection or reselection phase. These bits are read only and contain the encoded value of 0–15 possible IDs that could be used to select the LSI53C825A. During a SCSI selection or reselection phase when a valid ID is put on the bus, and the LSI53C825A responds to that ID, the “selected as” ID is written into these bits. These bits are used with [Response ID Zero \(RESPID0\)](#) and [Response ID One \(RESPID1\)](#) registers to allow response to multiple IDs on the bus.

#### SLT Selection Response Logic Test 3

This bit is set when the LSI53C825A 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 LSI53C825A 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 LSI53C825A 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 LSI53C825A SCSI functioning as an initiator, is waiting for the target

to request data transfers. If the LSI53C825A SCSI is functioning as a target, then the initiator has sent the offset number of acknowledges.

SOM

SCSI Synchronous Offset Maximum0

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 LSI53C825A SCSI is functioning as a target, and is waiting for the initiator to acknowledge the data transfers. If the LSI53C825A SCSI is functioning as an initiator, then the target has sent the offset number of requests.

**Register: 0x4D (0xCD)**  
**SCSI Test One (STEST1)**  
**Read/Write**

7	6	5	0				
SCLK	SISO	R					
0	0	x	x	x	x	x	x

SCLK

SCSI Clock7

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

SISO

SCSI Isolation Mode6

This bit allows the LSI53C825A 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[5:0]



**Register: 0x4E (0xCE)**  
**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 LSI53C825A 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. If a SCSI gross error occurs, set this bit. This bit automatically clears itself after resetting the synchronous offset.
- DIF** **SCSI Differential Mode** **5**  
Setting this bit allows the LSI53C825A to interface properly to external differential transceivers. Its only real effect is to 3-state the SBSY/, SSEL/, and SRST/ pads so that they can be used as pure inputs. Clearing this bit enables SE mode operation. This bit should be set in the initialization routine if the differential pair interface is used.
- SLB** **SCSI Loopback Mode** **4**  
Setting this bit allows the LSI53C825A to perform SCSI loopback diagnostics. That is, it enables the SCSI core to simultaneously perform as both the initiator and the target.

<b>SZM</b>	<b>SCSI High Impedance Mode</b> <b>3</b> 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.
<b>AWS</b>	<b>Always Wide SCSI</b> <b>2</b> When this bit is set, all SCSI information transfers are done in the 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.
<b>EXT</b>	<b>Extend SREQ/SACK/ Filtering</b> <b>1</b> 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.
	<p><u>Note:</u> Never set this bit during Fast SCSI (greater than 5 Mbytes transfers per second) operations, because a valid assertion could be treated as a glitch.</p>
<b>LOW</b>	<b>SCSI Low level Mode</b> <b>0</b> Setting this bit places the LSI53C825A in 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 <a href="#">SCSI Control Zero (SCNTL0)</a> register. SCSI bus transfers are performed by manually asserting and polling SCSI signals. Clearing this bit allows instructions to be executed in the SCSI SCRIPTS mode.
	<p><u>Note:</u> It is not necessary to set this bit for access to the SCSI bit-level registers (<a href="#">SCSI Output Data Latch (SODL)</a>, <a href="#">SCSI Bus Control Lines (SBCL)</a>, and input registers).</p>

**Register: 0x4F (0xCF)**  
**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	0	0	0	0

**TE** **TolerANT Enable** **7**

Setting this bit enables the active negation portion of LSI Logic TolerANT technology. Active negation causes the SCSI Request, Acknowledge, Data, and Parity signals to be actively deasserted, instead of relying on external pull-ups, when the LSI53C825A is driving these signals. Active deassertion of these signals occurs only when the LSI53C825A 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, "Introduction."](#)

**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 following table.

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<sub>DD</sub> during a power-down mode.

<b>DSI</b>	<b>Disable Single Initiator Response</b> <span style="float: right;"><b>4</b></span> If this bit is set, the LSI53C825A ignores all bus-initiated selection attempts that employ the single-initiator option from SCSI-1. In order to select the LSI53C825A while this bit is set, the LSI53C825A 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.
<b>S16</b>	<b>16-Bit System</b> <span style="float: right;"><b>3</b></span> If this bit is set, all devices in the SCSI system implementation are assumed to be 16-bit. This causes the LSI53C825A 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 LSI53C825A while this bit is set, the LSI53C825A ignores the selection attempt. This is because the parity bit for IDs 15–8 are not driven. See the description of the Enable Parity Checking bit in the <a href="#">SCSI Control Zero (SCNTL0)</a> register for more information.
<b>TTM</b>	<b>Timer Test Mode</b> <span style="float: right;"><b>2</b></span> 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 <a href="#">SCSI Interrupt Enable One (SIEN1)</a> register are asserted, the LSI53C825A generates interrupts at time-out. This bit is intended for internal manufacturing diagnosis and should not be used.
<b>CSF</b>	<b>Clear SCSI FIFO</b> <span style="float: right;"><b>1</b></span> Setting this bit causes the “full flags” for the SCSI FIFO to be cleared. This empties the FIFO. This bit is self-clearing. In addition to the SCSI FIFO pointers, the SIDL, SODL, and SODR full bits in the <a href="#">SCSI Status Zero (SSTAT0)</a> and <a href="#">SCSI Status Two (SSTAT2)</a> are cleared.
<b>STW</b>	<b>SCSI FIFO Test Write</b> <span style="float: right;"><b>0</b></span> 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 <a href="#">SCSI Output</a>

[Data Latch \(SODL\)](#) register cause the entire word contained in SODL to be loaded into the FIFO. These functions are summarized in the following table.

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

## Registers: 0x50–0x51 (0xD0–0xD1)

### 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 LSI53C825A 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; [SCSI Input Data Latch \(SIDL\)](#) contains latched data and the [SCSI Bus Data Lines \(SBDL\)](#) always contains exactly what is currently on the SCSI data bus. Reading this register causes the SCSI parity bit to be checked, and causes a parity error interrupt if the data is not valid. The power-up values are indeterminate.

## Registers: 0x54–0x55 (0xD4–0xD5)

### 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.

## Registers: 0x58–0x59 (0xD8–0xD9)

### 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 the 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 [SCSI Bus Data Lines \(SBDL\)](#) is read, both byte lanes are checked for parity regardless of phase. When in a nondata phase, this causes a parity error interrupt to be generated because upper byte lane parity is invalid.

## Registers: 0x5C–0x5F (0xDC–0xDF)

### Scratch Register B (SCRATCHB)

Read/Write

31	SCRATCHB																												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

#### 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 alter its contents. The LSI53C825A cannot fetch SCRIPTS instructions from this location. When bit 3 in the [Chip Test Two \(CTEST2\)](#) register is set, this register contains the base address for the 4 Kbyte internal RAM. Setting [Chip Test Two \(CTEST2\)](#), bit 3 only causes the base address to appear in the [Scratch Register B \(SCRATCHB\)](#) register; any information that was previously in the register remains intact. Any writes to this register while the bit is set passes through the actual [Scratch Register B \(SCRATCHB\)](#) register. The power-up values are indeterminate.

## Registers: 0x60–0x7F (0xE0–0xFF)

### Scratch Registers C–J (SCRATCHC–SCRATCHJ)

Read/Write

These registers are general purpose scratch registers for user defined functions. The LSI53C825A cannot fetch SCRIPTS instructions from this location. The power-up value of these registers is indeterminate.





# Chapter 5

## SCSI SCRIPTS

### Instruction Set

---

After power-up and initialization of the LSI53C825A, the chip can be operated in the low level register interface mode or in the high level SCSI SCRIPTS mode.

Chapter 5 is divided into the following sections:

- [Section 5.1, “Low Level Register Interface Mode”](#)
  - [Section 5.2, “High Level SCSI SCRIPTS Mode”](#)
  - [Section 5.3, “Block Move Instructions”](#)
  - [Section 5.4, “I/O Instruction”](#)
  - [Section 5.5, “Read/Write Instructions”](#)
  - [Section 5.6, “Transfer Control Instructions”](#)
  - [Section 5.7, “Memory Move Instructions”](#)
  - [Section 5.8, “Load and Store Instructions”](#)
- 

## 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 LSI53C825A requires only a SCRIPTS start address. The start address must be at a Dword (four byte) boundary to align all the following SCRIPTS at a Dword boundary since all SCRIPTS are 8 or 12 bytes long. Instructions are fetched until an interrupt instruction is encountered, or until an unexpected event (such as a hardware error) causes an interrupt to the external processor.

Once an interrupt is generated, the LSI53C825A 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 LSI53C825A 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 LSI53C825A, 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 LSI53C825A 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 LSI53C825A 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

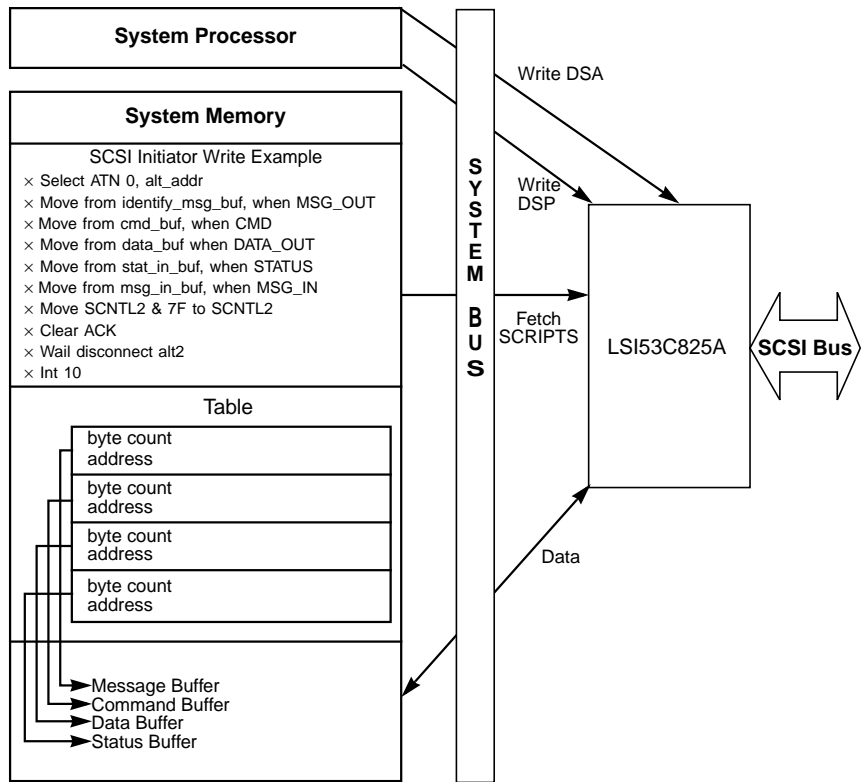
This sample operation describes execution of a SCRIPTS instruction for a Block Move instruction.

- 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 LSI53C825A to fetch its first instruction at the address just loaded. This is from main memory or the internal RAM, depending on the address.
- The LSI53C825A 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 LSI53C825A 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 LSI53C825A requests use of the PCI bus again to transfer the data.
- When the LSI53C825A 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 releases the PCI bus. The LSI53C825A 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 LSI53C825A 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 LSI53C825A 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 LSI53C825A 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 Instructions

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

IT[1:0]                      **Instruction Type - Block Move**                      [31:30]

IA                              **Indirect Addressing**                              29

When this bit is cleared, user data is moved to or from the 32-bit data start address for the Block Move instruction. The value is loaded into the chip's address register and incremented as data is transferred. The address of the data to move is in the second Dword of this instruction.

When 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

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.

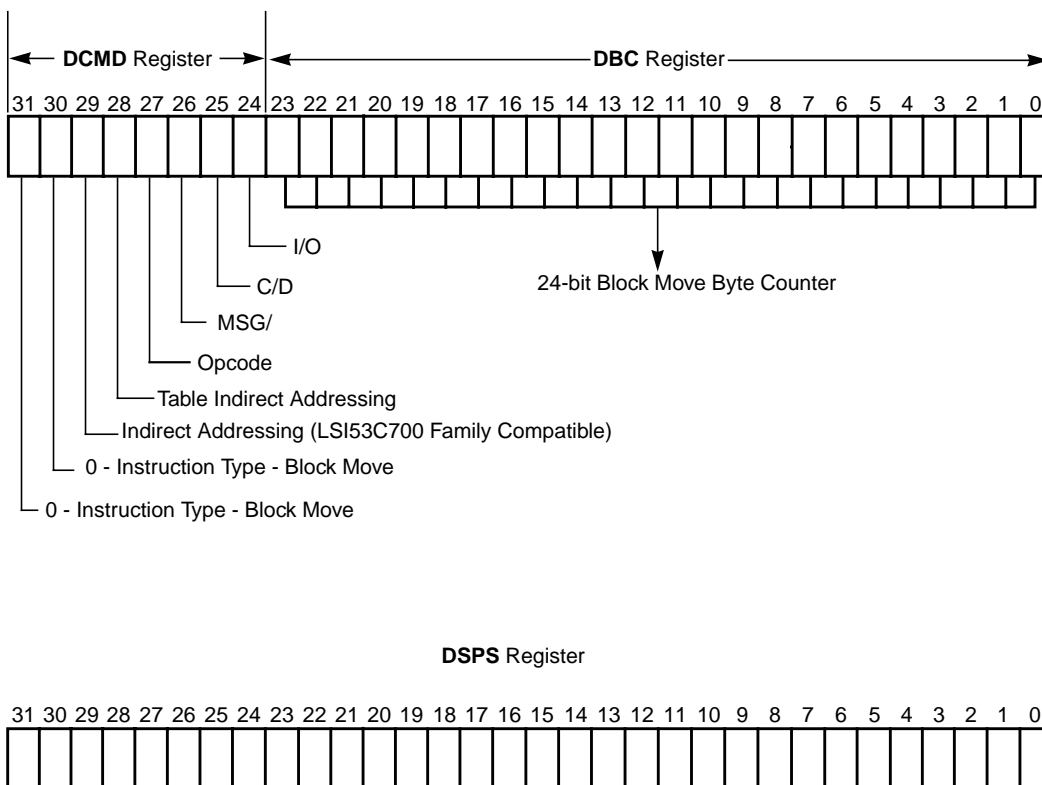
Use the signed integer offset in bits [23:0] of the second four bytes of the instruction, added to the value in the [Data Structure Address \(DSA\)](#) register, to fetch first the byte count and then the data address. The signed value is combined with the data structure base address to generate the 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.

[Figure 5.2](#) illustrates the Block Move Instruction register.

**Figure 5.2 Block Move Instruction Register**



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 LSI53C825A. 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 performed in Target and Initiator mode.

### Target Mode

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

OPC	Instruction Defined
0	MOVE
1	CHMOV

These instructions perform the following steps:

1. The LSI53C825A verifies that it is connected to the SCSI bus as a Target before executing this instruction.
2. The LSI53C825A asserts the SCSI phase signals (MSG/, 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 LSI53C825A 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 ([SCSI Control Two \(SCNTL2\)](#), bit 1) is clear, then the LSI53C825A 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 ([SCSI Control Two \(SCNTL2\)](#), bit 1) is set, the LSI53C825A 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 LSI53C825A 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 LSI53C825A requests the number of bytes specified in the [DMA Byte Counter \(DBC\)](#) register. If the [DMA Byte Counter \(DBC\)](#) register contains 0x000000, an illegal instruction interrupt is generated.
4. The LSI53C825A transfers the number of bytes specified in the [DMA Byte Counter \(DBC\)](#) register starting at the address specified in the [DMA Next Address \(DNAD\)](#) register. If the Opcode bit is set and a data transfer ends on an odd byte boundary, the LSI53C825A 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 LSI53C825A 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 LSI53C825A verifies that it is connected to the SCSI bus as an Initiator before executing this instruction.
2. The LSI53C825A waits for an unserviced phase to occur. An unserviced phase is any phase (with SREQ/ asserted) for which the LSI53C825A has not yet transferred data by responding with a SACK/.
3. The LSI53C825A 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 LSI53C825A 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 LSI53C825A stores the last byte in the [SCSI Wide Residue \(SWIDE\)](#) register during a receive operation, or in the [SCSI Output Control Latch \(SOCL\)](#) 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 LSI53C825A generates a phase mismatch interrupt and the instruction is not executed.
6. During a Message-Out phase, after the LSI53C825A has performed a select with Attention (or SATN/ is manually asserted with a Set ATN instruction), the LSI53C825A deasserts SATN/ during the final SREQ/SACK/ handshake.
7. When the LSI53C825A 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 LSI53C825A operates in Initiator mode, these bits are compared with the latched SCSI phase bits in the [SCSI Status One \(SSTAT1\)](#) register. When the LSI53C825A operates in Target mode, it asserts the phase defined in this field. The following table describes the possible combinations and the corresponding SCSI phase.

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

**TC[23:0]**

### **Transfer Counter**

**[23:0]**

This 24-bit field specifies the number of data bytes to be moved between the LSI53C825A and system memory. The field is stored in the [DMA Byte Counter \(DBC\)](#) register. When the LSI53C825A transfers data to/from memory, the [DMA Byte Counter \(DBC\)](#) 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 [DMA Byte Counter \(DBC\)](#) register is decremented to zero. At this time, the LSI53C825A 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 LSI53C825A transfers data to or from memory, the [DMA Next Address \(DNAD\)](#) 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

<b>IT[1:0]</b>	<b>Instruction Type - I/O Instruction</b> <b>[31:30]</b> The IT bit configuration (01) defines an I/O Instruction Type.
<b>OPC[2:0]</b>	<b>Opcode</b> <b>[29:27]</b> 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 <a href="#">Section 5.5, “Read/Write Instructions.”</a> 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

1. The LSI53C825A 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.
2. If the LSI53C825A wins arbitration, it attempts to reselect the SCSI device whose ID is defined in the destination ID field of the instruction. Once the LSI53C825A 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.
3. If the LSI53C825A 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 LSI53C825A to Initiator mode if it is reselected, or to Target mode if it is selected.

### Disconnect Instruction

The LSI53C825A disconnects from the SCSI bus by deasserting all SCSI signal outputs.

### Wait Select Instruction

1. If the LSI53C825A is selected, it fetches the next instruction from the address pointed to by the [DMA SCRIPTS Pointer \(DSP\)](#) register.
2. If reselected, the LSI53C825A 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 LSI53C825A to Initiator mode when it is reselected.
3. If the CPU sets the SIGP bit in the [SCSI Status Zero \(SSTAT0\)](#) register, the LSI53C825A 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

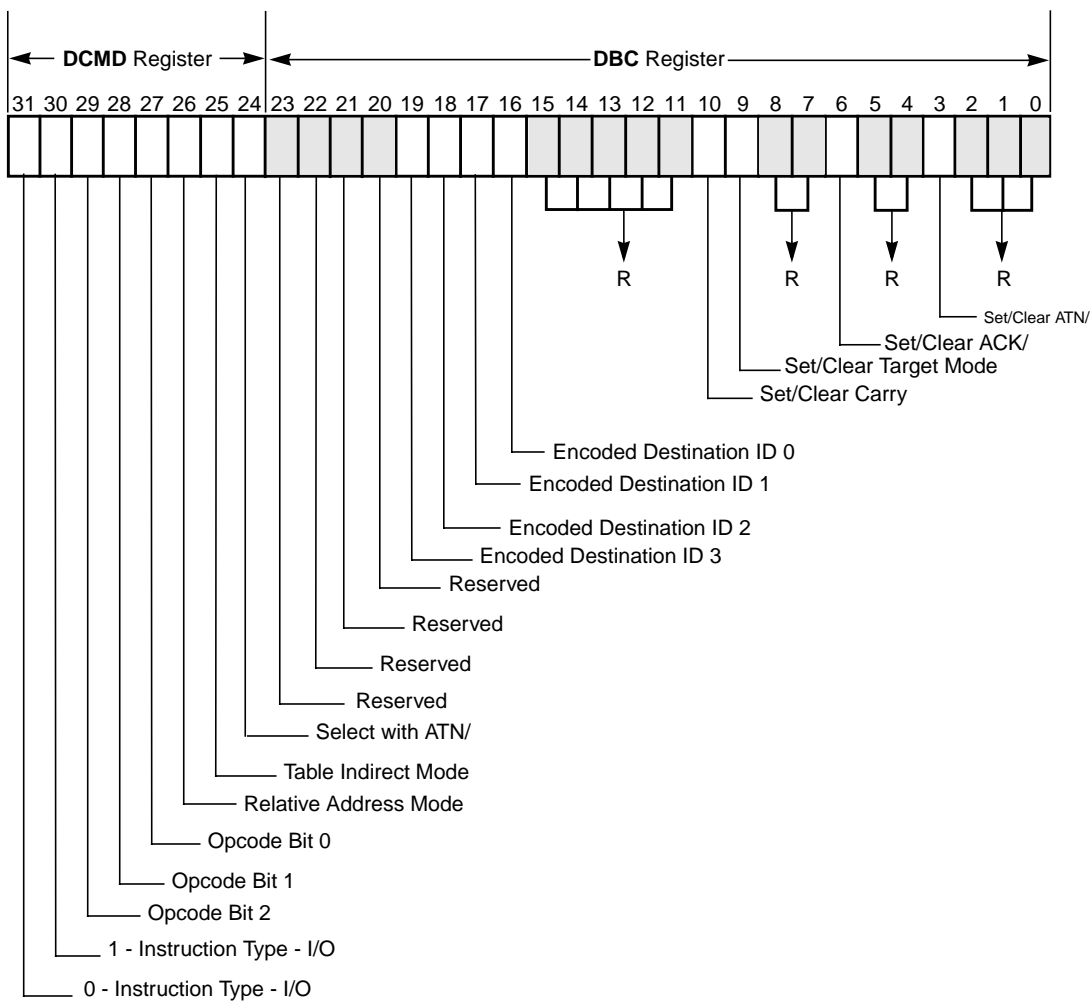
When the SACK/ or SATN/ bits are cleared, the corresponding bits are cleared in the [SCSI Output Control Latch \(SOCL\)](#) register. Do not set SACK/ or SATN/ except for testing purposes. When the target bit is cleared, the corresponding bit in the [SCSI Control Zero \(SCNTL0\)](#) register is cleared. 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.

[Figure 5.3](#) illustrates the I/O Instruction register.

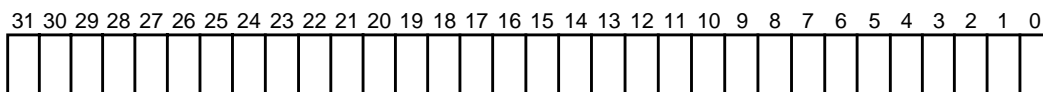


**Figure 5.3 I/O Instruction Register**



Second 32-bit Word of the I/O Instruction

**DSPS Register**



32-bit Jump Address

## 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

1. The LSI53C825A 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.
2. If the LSI53C825A wins arbitration, it attempts to select the SCSI device whose ID is defined in the destination ID field of the instruction. Once the LSI53C825A 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.
3. If the LSI53C825A 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 LSI53C825A to Initiator mode if it is reselected, or to Target mode if it is selected.
4. If the Select with SATN/ field is set, the SATN/ signal is asserted during the selection phase.

### Wait Disconnect Instruction

The LSI53C825A 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 LSI53C825A receives a Disconnect Message or a Command Complete Message.

### **Wait Reselect Instruction**

1. If the LSI53C825A 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 LSI53C825A to Target mode when it is selected.
2. If the LSI53C825A is reselected, it fetches the next instruction from the address pointed to by the [DMA SCRIPTS Pointer \(DSP\)](#) register.
3. If the CPU sets the SIGP bit in the Interrupt Status Zero (ISTAT0) register, the LSI53C825A 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	

<b>Sel</b>	<b>Select with ATN/</b>	<b>24</b>
	This bit specifies whether SATN/ is asserted during the selection phase when the LSI53C825A 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.	
<b>R</b>	<b>Reserved</b>	<b>[23:20]</b>
<b>ENDID[3:0]</b>	<b>Encoded SCSI Destination ID</b>	<b>[19:16]</b>
	This 4-bit field specifies the destination SCSI ID for an I/O instruction.	
<b>R</b>	<b>Reserved</b>	<b>[15:11]</b>
<b>CC</b>	<b>Set/Clear Carry</b>	<b>10</b>
	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.	
<b>TM</b>	<b>Set/Clear Target Mode</b>	<b>9</b>
	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 LSI53C825A as a Target device (this sets bit 0 of the <a href="#">SCSI Control Zero (SCNTL0)</a> register). Clearing this bit with a Clear instruction configures the LSI53C825A as an Initiator device (this clears bit 0 of the <a href="#">SCSI Control Zero (SCNTL0)</a> register).	

<b>R</b>	<b>Reserved</b>	<b>[8:7]</b>
<b>ACK</b>	<b>Set/Clear SACK/</b>	<b>6</b>
<b>R</b>	<b>Reserved</b>	<b>[5:4]</b>
<b>ATN</b>	<b>Set/Clear SATN/</b>	<b>3</b>
<p>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.</p> <p>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 <a href="#">SCSI Output Control Latch (SOCL)</a> register is set or cleared depending on the instruction used.</p> <p>Since SACK/ and SATN/ are Initiator signals, they are not asserted on the SCSI bus unless the LSI53C825A is operating as an Initiator or the SCSI Loopback Enable bit is set in the <a href="#">SCSI Test Two (STEST2)</a> register.</p> <p>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.</p>		
<b>R</b>	<b>Reserved</b>	<b>[2:0]</b>

## 5.4.2 Second Dword

<b>SA</b>	<b>Start Address</b>	<b>[31:0]</b>
<p>This 32-bit field contains the memory address to fetch the next instruction if the selection or reselection fails.</p> <p>If relative or table relative addressing is used, this value is a 24-bit signed offset relative to the current <a href="#">DMA SCRIPTS Pointer (DSP)</a> register value.</p>		

---

## 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.

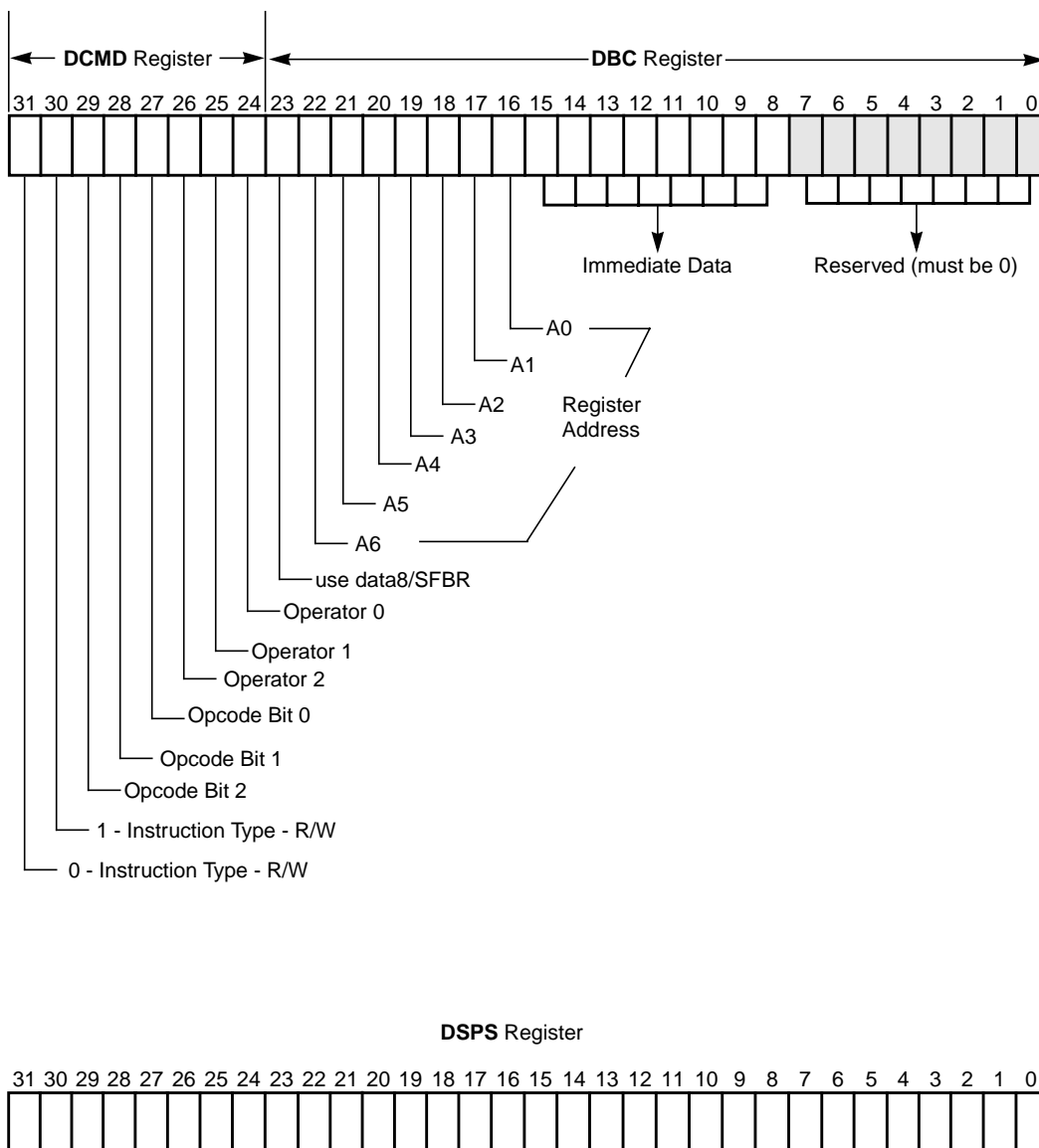
### 5.5.1 First Dword

<b>IT[1:0]</b>	<b>Instruction Type - Read/Write Instruction</b> <b>[31:30]</b> 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.
----------------	---

[Figure 5.4](#) illustrates the Read/Write Instruction register.



**Figure 5.4 Read/Write Instruction Register**



<b>OPC[2:0]</b>	<b>Opcode</b> 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.	<b>[29:27]</b>
<b>O[2:0]</b>	<b>Operator</b> These bits are used in conjunction with the opcode bits to determine which instruction is currently selected. Refer to <a href="#">Table 5.1</a> for field definitions.	<b>[26:24]</b>
<b>D8</b>	<b>Use data8/SFBR</b> When this bit is set, <a href="#">SCSI First Byte Received (SFBR)</a> is used instead of the data8 value during a Read-Modify-Write instruction (see <a href="#">Table 5.1</a> ). This allows the user to add two register values.	<b>23</b>
<b>A[6:0]</b>	<b>Register Address - A[6:0]</b> It is possible to change register values from SCRIPTS in read-modify-write cycles or move to/from <a href="#">SCSI First Byte Received (SFBR)</a> cycles. A[6:0] selects an 8-bit source/destination register within the LSI53C825A.	<b>[22:16]</b>
<b>ImmD</b>	<b>Immediate Data</b> This 8-bit value is used as a second operand in logical and arithmetic functions.	<b>[15:8]</b>
<b>R</b>	<b>Reserved</b>	<b>[7:0]</b>

## 5.5.2 Second Dword

<b>Destination Address</b>	<b>[31:0]</b>
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 [SCSI First Byte Received \(SFBR\)](#) is used instead of data8 in the instruction syntax. To subtract one value from another when using SFBR, first XOR the value to subtract (subtrahend) with 0xFF, and add 1 to the resulting value. This creates the 2's complement of the subtrahend. The two values are then added to obtain the difference.

### 5.5.4 Move To/From SFBR Cycles

All operations are read-modify-writes. However, two registers are involved, one of which is always the [SCSI First Byte Received \(SFBR\)](#). [Table 5.2](#) 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 [SCSI First Byte Received \(SFBR\)](#) from/to any other register.
- Alter the value of a register with AND, OR, ADD, XOR, SHIFT LEFT, or SHIFT RIGHT operators.
- After moving values to the [SCSI First Byte Received \(SFBR\)](#), the compare and jump, call, or similar instructions are used to check the value.
- A Move-to-SFBR followed by a Move-from-SFBR is used to perform a register-to-register move.

**Table 5.2 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 <a href="#">SCSI First Byte Received (SFBR)</a> register. Syntax: "Move data8 to SFBR"	Move data into register. Syntax: "Move data8 to RegA"
001 <sup>1</sup>	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 <a href="#">SCSI First Byte Received (SFBR)</a> register. Syntax: "Move RegA SHL SFBR"	Shift the <a href="#">SCSI First Byte Received (SFBR)</a> register one bit to the left and place the result in the register. Syntax: "Move SFBR SHL RegA"

**Table 5.2 Read/Write Instructions (Cont.)**

Operator	Opcode 111 Read-Modify-Write	Opcode 110 Move to SFBR	Opcode 101 Move from SFBR
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 <b>SCSI First Byte Received (SFBR)</b> 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 <b>SCSI First Byte Received (SFBR)</b> register. Syntax: "Move RegA XOR data8 to SFBR"	XOR data with SFBR and place the result in the register. Syntax: "Move SFBR XOR data8 to RegA"
100	AND data with register and place the result in the same register. Syntax: "Move RegA & data8 to RegA"	AND data with register and place the result in the <b>SCSI First Byte Received (SFBR)</b> 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 <sup>1</sup>	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 <b>SCSI First Byte Received (SFBR)</b> register. Syntax: "Move RegA SHR SFBR"	Shift the <b>SCSI First Byte Received (SFBR)</b> 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 <b>SCSI First Byte Received (SFBR)</b> 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 <b>SCSI First Byte Received (SFBR)</b> 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 **SCSI First Byte Received (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

**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 the following table.

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 LSI53C825A 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 [SCSI First Byte Received \(SFBR\)](#) register with the contents of the [DMA SCRIPTS Pointer Save \(DSPS\)](#) register. The [DMA SCRIPTS Pointer \(DSP\)](#) register now contains the address of the next instruction.

If the comparisons are false, the LSI53C825A fetches the next instruction from the address pointed to by the [DMA SCRIPTS Pointer \(DSP\)](#) register, leaving the instruction pointer unchanged.

### **Call Instruction**

The LSI53C825A 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.

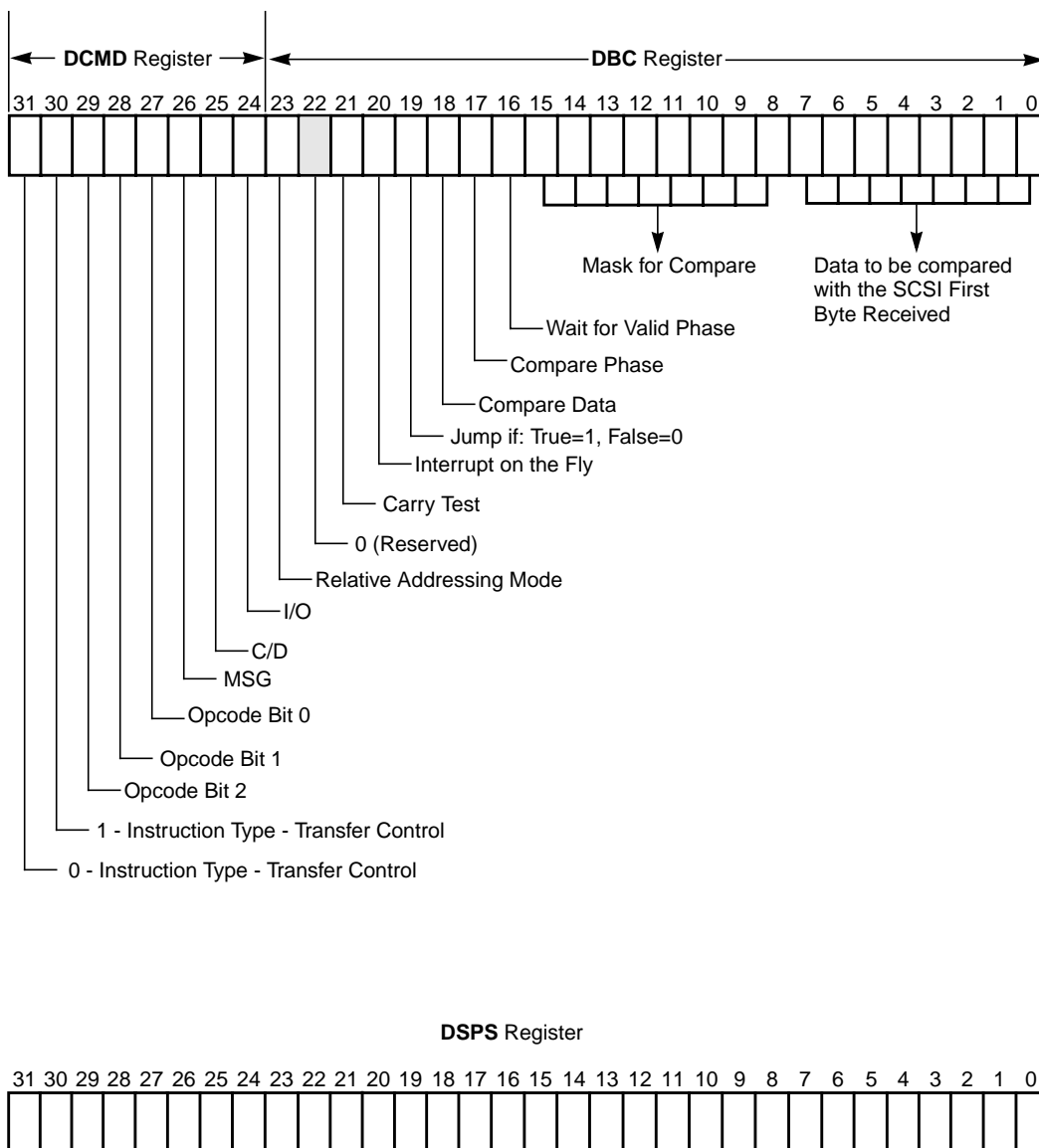
If the comparisons are true, it loads the [DMA SCRIPTS Pointer \(DSP\)](#) register with the contents of the [DMA SCRIPTS Pointer Save \(DSPS\)](#) register and that address value becomes the address of the next instruction.

When the LSI53C825A executes a Call instruction, the instruction pointer contained in the [DMA SCRIPTS Pointer \(DSP\)](#) register is stored in the [Temporary \(TEMP\)](#) register. Since the [Temporary \(TEMP\)](#) register is not a stack and can only hold one Dword, nested call instructions are not allowed.

If the comparisons are false, the LSI53C825A fetches the next instruction from the address pointed to by the [DMA SCRIPTS Pointer \(DSP\)](#) register and the instruction pointer is not modified.

[Figure 5.5](#) illustrates the Transfer Control instruction.

**Figure 5.5 Transfer Control Instruction**



## Return Instruction

The LSI53C825A can do a true/false comparison of the ALU carry bit, or compare the phase and/or data as defined by the Phase Compare, Data Compare, and True/False bit fields.

If the comparisons are true, it loads the [DMA SCRIPTS Pointer \(DSP\)](#) register with the contents of the [DMA SCRIPTS Pointer Save \(DSPS\)](#) register. That address value becomes the address of the next instruction.

When a Return instruction is executed, the value stored in the [Temporary \(TEMP\)](#) register is returned to the [DMA SCRIPTS Pointer \(DSP\)](#) register. The LSI53C825A 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 LSI53C825A fetches the next instruction from the address pointed to by the [DMA SCRIPTS Pointer \(DSP\)](#) register and the instruction pointer is not modified.

## Interrupt Instruction

The LSI53C825A 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 LSI53C825A 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 LSI53C825A halts and the [DMA SCRIPTS Pointer \(DSP\)](#) register must be written to before starting any further operation.



### Interrupt-on-the-Fly Instruction

The LSI53C825A 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 \(ISTAT\)](#), bit 2) is set, the LSI53C825A asserts the Interrupt-on-the-Fly bit.

### SCSIP[2:0]

#### SCSI Phase

[26:24]

This 3-bit field corresponds to the three SCSI bus phase signals that are compared with the phase lines latched when SREQ/ is asserted. Comparisons can be performed to determine the SCSI phase actually being driven on the SCSI bus. The following table describes the possible combinations and their corresponding SCSI phase. These bits are only valid when the LSI53C825A is operating in Initiator mode. Clear these bits when the LSI53C825A is operating in Target mode.

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-Out
1	0	1	Reserved-In
1	1	0	Message-Out
1	1	1	Message-In

### RA

#### Relative Addressing Mode

23

When this bit is set, the 24-bit signed value in the [DMA SCRIPTS Pointer Save \(DPS\)](#) 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 LSI53C825A. 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.

## 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 \(ISTAT\)](#), bit 2) is asserted.

JMP

Jump If True/False

19

This bit determines whether the LSI53C825A 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 LSI53C825A 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 LSI53C825A is operating in Target mode and this bit is set, it tests for an active SCSI SATN/ signal.

<b>WVP</b>	<b>Wait For Valid Phase</b> <span style="float: right;"><b>16</b></span> If the Wait for Valid Phase bit is set, the LSI53C825A waits for a previously unserviced phase before comparing the SCSI phase and data.  If the Wait for Valid Phase bit is cleared, the LSI53C825A compares the SCSI phase and data immediately.
<b>DCM</b>	<b>Data Compare Mask</b> <span style="float: right;"><b>[15:8]</b></span> 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 <a href="#">SCSI First Byte Received (SFBR)</a> 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.
<b>DCV</b>	<b>Data Compare Value</b> <span style="float: right;"><b>[7:0]</b></span> 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

<b>Jump Address</b> <span style="float: right;"><b>[31:0]</b></span> This 32-bit field contains the address of the next instruction to fetch when a jump is taken. Once the LSI53C825A fetches the instruction from the address pointed to by these 32 bits, this address is incremented by 4, loaded into the <a href="#">DMA SCRIPTS Pointer (DSP)</a> 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 LSI53C825A 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

<b>IT[2:0]</b>	<b>Instruction Type - Memory Move</b> <span style="float: right;"><b>[31:29]</b></span> The IT bit configuration (110) defines a Memory Move Instruction Type.
<b>R</b>	<b>Reserved</b> <span style="float: right;"><b>[28:25]</b></span> These bits are reserved and must be zero. If any of these bits are set, an illegal instruction interrupt occurs.
<b>NF</b>	<b>No Flush</b> <span style="float: right;"><b>24</b></span> When this bit is set, the LSI53C825A 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](#).

<b>TC[23:0]</b>	<b>Transfer Counter</b>	<b>[23:0]</b>
	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 LSI53C825A responds to addresses as defined in the [Base Address Zero \(I/O\)](#) or [Base Address 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 [SCSI First Byte Received \(SFBR\)](#) is not writable using the CPU, and therefore not by a Memory Move. However, it can be loaded using SCRIPTS Read/Write operations. To load the SFBR with a byte stored in system memory, first move the byte to an intermediate LSI53C825A register (for example, a SCRATCH register), and then to the [SCSI First Byte Received \(SFBR\)](#).

The same address alignment restrictions apply to register access operations as to normal memory-to-memory transfers.

## 5.7.3 Second Dword

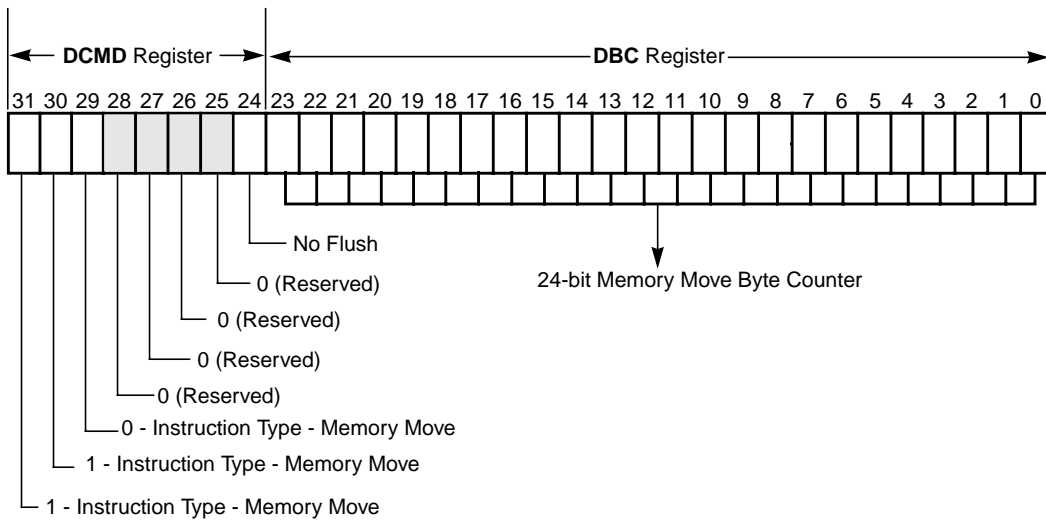
<b>DSPS Register</b>	<b>[31:0]</b>
These bits contain the source address of the Memory Move.	

## 5.7.4 Third Dword

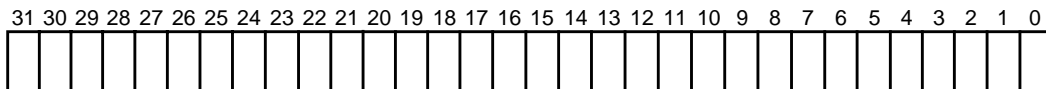
<b>TEMP Register</b>	<b>[31:0]</b>
These bits contain the destination address for the Memory Move.	

[Figure 5.6](#) illustrates the Memory Move instruction.

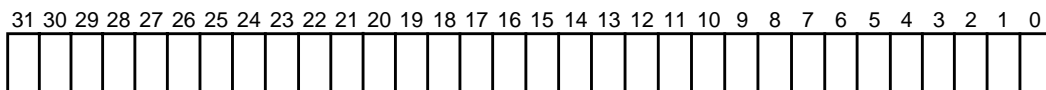
**Figure 5.6 Memory Move Instruction**



**DSPS Register**



**TEMP Register**



---

## 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

<b>IT[2:0]</b>	<b>Instruction Type</b> These bits should be 0b111, indicating the Load and Store instruction.	<b>[31:29]</b>
<b>DSA</b>	<b>DSA Relative</b> When this bit is cleared, the value in the <a href="#">DMA SCRIPTS Pointer Save (DSPS)</a> 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 <a href="#">DMA SCRIPTS Pointer Save (DSPS)</a> to the <a href="#">Data Structure Address (DSA)</a> .	<b>28</b>
<b>R</b>	<b>Reserved</b>	<b>[27:26]</b>
<b>NF</b>	<b>No Flush (Store instruction only)</b> When this bit is set, the LSI53C825A 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.	<b>25</b>
<u>Note:</u> This bit has no effect unless the Prefetch Enable bit in the <a href="#">DMA Control (DCNTL)</a> register is set.		
<b>LS</b>	<b>Load and Store</b> When this bit is set, the instruction is a Load. When cleared, it is a Store.	<b>24</b>
<b>R</b>	<b>Reserved</b>	<b>[23]</b>
<b>RA[6:0]</b>	<b>Register Address</b> A[6:0] selects the register to Load and Store to/from within the LSI53C825A.	<b>[22:16]</b>
<u>Note:</u> It is not possible to Load the <a href="#">SCSI First Byte Received (SFBR)</a> register, although it is possible to store the SFBR contents to another location.		
<b>R</b>	<b>Reserved</b>	<b>[15:3]</b>
<b>BC</b>	<b>Byte Count</b> This value is the number of bytes to Load and Store.	<b>[2:0]</b>

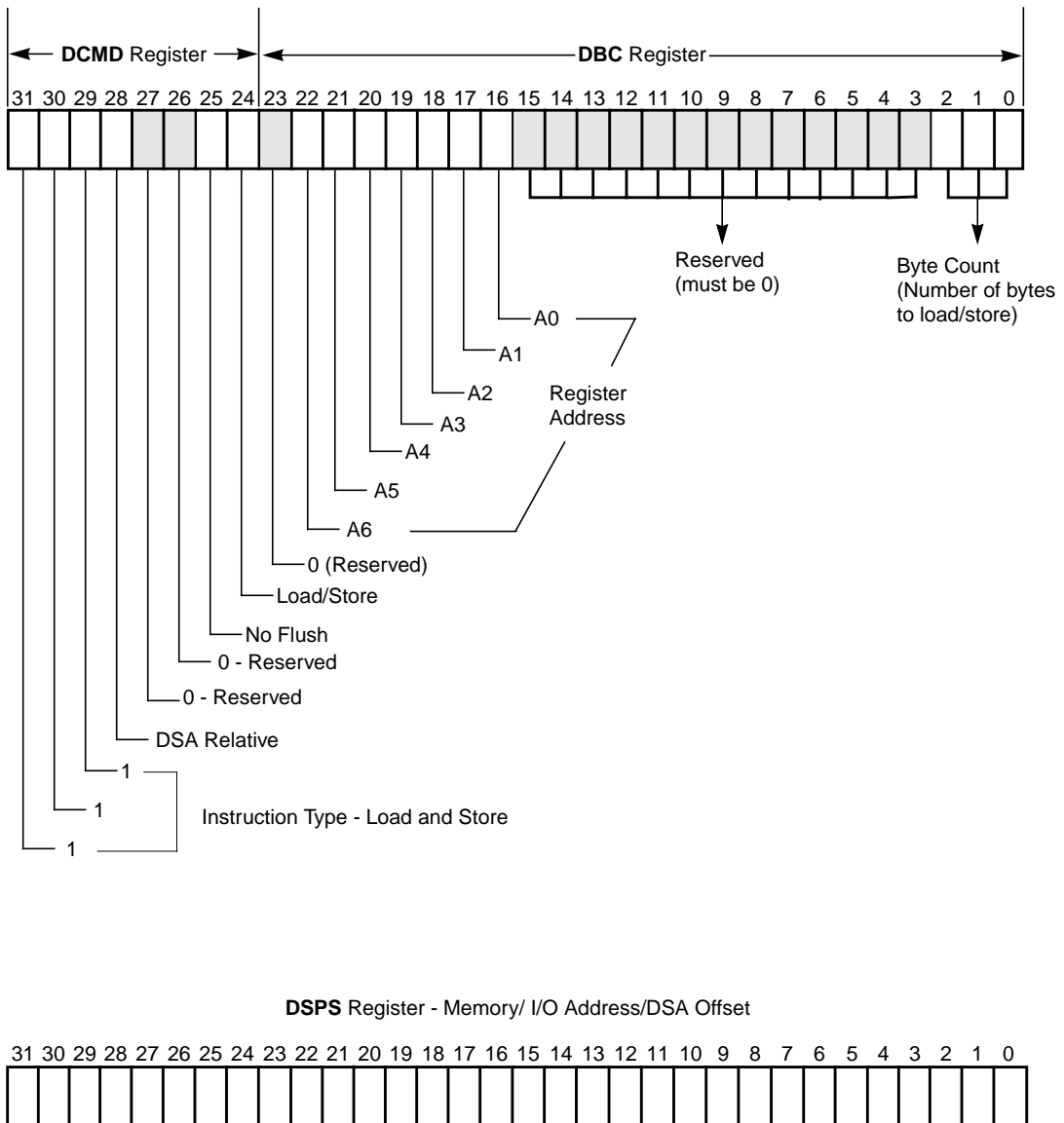
## 5.8.2 Second Dword

### 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.

[Figure 5.7](#) illustrates the Load and Store Instruction format.

**Figure 5.7 Load and Store Instruction Format**





# Chapter 6

## Specifications

---

This chapter specifies the LSI53C825A electrical and mechanical characteristics. It is divided into the following sections:

- [Section 6.1, “DC Characteristics”](#)
- [Section 6.2, “TolerANT Technology Electrical Characteristics”](#)
- [Section 6.3, “AC Characteristics”](#)
- [Section 6.4, “PCI and External Memory Interface Timing Diagrams”](#)
- [Section 6.5, “PCI and External Memory Interface Timing”](#)
- [Section 6.6, “SCSI Timing Diagrams”](#)
- [Section 6.7, “Package Drawings”](#)

### 6.1 DC Characteristics

This section describes the LSI53C825A DC characteristics. [Table 6.1](#) through [Table 6.14](#) give current and voltage specifications.

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 6.1 Absolute Maximum Stress Ratings**

Symbol	Parameter	Min	Max	Unit	Test Conditions
$T_{STG}$	Storage temperature	–55	150	°C	–
$V_{DD}$	Supply voltage	–0.5	7.0	V	–
$V_{IN}$	Input voltage	$V_{SS} - 0.5$	$V_{DD} + 0.5$	V	–
$I_{LP}^1$	Latch-up current	$\pm 150$	–	mA	–
ESD <sup>2</sup>	Electrostatic discharge	–	2 K	V	MIL-STD 883C, Method 3015.7

1.  $-2\text{ V} < V_{PIN} < 8\text{ V}$ .

2. SCSI pins only.

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	4.75	5.25	V	–
$I_{DD}$	Supply current (dynamic)	–	130	mA	–
	Supply current (static)	–	1	mA	–
$T_A$	Operating free air	0	70	°C	–
$\theta_{JA}$	Thermal resistance (junction to ambient air)	–	67	°C/W	–

Note: Conditions that exceed the operating limits may cause the device to function incorrectly.

**Table 6.3 SCSI Signals—SD[15:0]/, SDP[1:0]/, SREQ/, SACK/**

Symbol	Parameter	Min	Max	Unit	Test Conditions
$V_{IH}$	Input high voltage	2.0	$V_{DD} + 0.5$	V	—
$V_{IL}$	Input low voltage	$V_{SS} - 0.5$	0.8	V	—
$V_{OH}^1$	Output high voltage	2.5	3.5	V	2.5 mA
$V_{OL}$	Output low voltage	$V_{SS}$	0.5	V	48 mA
$I_{OZ}$	3-state leakage	–10	10	$\mu A$	—

1. TolerANT active negation enabled.

**Table 6.4 SCSI Signals—SMSG, SI\_O/, SC\_D/, SATN/, SBSY/, SSEL/, SRST/**

Symbol	Parameter	Min	Max	Unit	Test Conditions
$V_{IH}$	Input high voltage	2.0	$V_{DD} + 0.5$	V	—
$V_{IL}$	Input low voltage	$V_{SS} - 0.5$	0.8	V	—
$V_{OL}$	Output low voltage	$V_{SS}$	0.5	V	48 mA
$I_{OZ}$	3-state leakage (SRST/ only)	–10 –500	10 –50	$\mu A$	—

**Table 6.5 Input Signals<sup>1</sup>—CLK, SCLK, GNT/, IDSEL, RST/, TESTIN, DIFFSENS, BIG\_LIT/**

Symbol	Parameter	Min	Max	Unit	Test Conditions
$V_{IH}$	Input high voltage	2.0	$V_{DD} + 0.5$	V	—
$V_{IL}$	Input low voltage	$V_{SS} - 0.5$	0.8	V	—
$I_{IN}$	Input leakage	–10	10	$\mu A$	—

1. CLK, SCLK, and BIG\_LIT/ have 100  $\mu A$  pull-ups, and GNT/ and IDSEL have 25  $\mu A$  pull-ups, that are enabled when TESTIN is low. TESTIN has a 100  $\mu A$  pull-up that is always enabled.

**Table 6.6 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	–	10	pF	–

**Table 6.7 Output Signals<sup>1</sup>—MAC/\_TESTOUT, REQ/**

Symbol	Parameter	Min	Max	Unit	Test Conditions
$V_{OH}$	Output high voltage	2.4	$V_{DD}$	V	–16 mA
$V_{OL}$	Output low voltage	$V_{SS}$	0.4	V	16 mA
$I_{OZ}$	3-state leakage	–10	10	$\mu A$	–

1. REQ/ has a 100  $\mu A$  pull-up that is enabled when TESTIN is low.

**Table 6.8 Output Signals<sup>1</sup>—IRQ/, SDIR[15:0], SDIRP0, SDIRP1, BSYDIR, SELDIR, RSTDIR, TGS, IGS, MAS/[1:0], MCE/, MOE/, MWE/**

Symbol	Parameter	Min	Max	Unit	Test Conditions
$V_{OH}$	Output high voltage	2.4	$V_{DD}$	V	–4 mA <sup>2</sup>
$V_{OL}$	Output low voltage	$V_{SS}$	0.4	V	4 mA <sup>2</sup>
$I_{OZ}$	3-state leakage	–10	10	$\mu A$	–

1. IRQ/, MAS/[1:0], MCE/, MOE/, and MWE/ have a 100  $\mu A$  pull-up that is enabled when TESTIN is low. IRQ/ can be enabled with a register as an open drain with an internal 100  $\mu A$  pull-up.

2. For IRQ/, Test Conditions are 8 mA.

**Table 6.9 Output Signal—SERR/**

Symbol	Parameter	Min	Max	Unit	Test Conditions
$V_{OL}$	Output low voltage	$V_{SS}$	0.4	V	16 mA
$I_{OZ}$	3-state leakage	–10	10	$\mu A$	–



**Table 6.10 Bidirectional Signals<sup>1</sup>—AD[31:0], C\_BE[3:0], FRAME/, IRDY/, TRDY/, DEVSEL/, STOP/, PERR/, PAR/**

Symbol	Parameter	Min	Max	Unit	Test Conditions
V <sub>IH</sub>	Input high voltage	2.0	V <sub>DD</sub> +0.5	V	–
V <sub>IL</sub>	Input low voltage	V <sub>SS</sub> –0.5	0.8	V	–
V <sub>OH</sub>	Output high voltage	2.4	V <sub>DD</sub>	V	16 mA
V <sub>OL</sub>	Output low voltage	V <sub>SS</sub>	0.4	V	16 mA
I <sub>OZ</sub>	3-state leakage	–10	10	μA	–

1. All the signals in this table have 25 μA pull-ups that are enabled when TESTIN is low.

**Table 6.11 Bidirectional Signals<sup>1</sup>—GPIO0\_FETCH/, GPIO1\_MASTER/, GPIO2\_MAS2/, GPIO3, GPIO4**

Symbol	Parameter	Min	Max	Unit	Test Conditions
V <sub>IH</sub>	Input high voltage	2.0	V <sub>DD</sub> +0.5	V	–
V <sub>IL</sub>	Input low voltage	V <sub>SS</sub> –0.5	0.8	V	–
V <sub>OH</sub>	Output high voltage	2.4	V <sub>DD</sub>	V	–16 mA
V <sub>OL</sub>	Output low voltage	V <sub>SS</sub>	0.4	V	16 mA
I <sub>OZ</sub>	3-state leakage	–10	10	μA	–

1. All the signals in this table have 100 μA pull-ups that are enabled when TESTIN is low.

**Table 6.12 Bidirectional Signals<sup>1</sup>—MAD[7:0]**

Symbol	Parameter	Min	Max	Unit	Test Conditions
$V_{IH}$	Input high voltage	2.0	$V_{DD} + 0.5$	V	–
$V_{IH}$	Input high voltage - external memory pull-downs	3.85	$V_{DD} + 0.5$	V	–
$V_{IL}$	Input low voltage	$V_{SS} - 0.5$	0.8	V	–
$V_{IL}$	Input low voltage - external memory pull-downs	$V_{SS} - 0.5$	1.35	V	–
$V_{OH}$	Output high voltage	2.4	$V_{DD}$	V	–4 mA
$V_{OL}$	Output low voltage	$V_{SS}$	0.4	V	4 mA
$I_{OZ}$	3-state leakage	–10	10	$\mu A$	–

1. All the signals in this table have 100  $\mu A$  pull-ups that are enabled when TESTIN is low.

**Table 6.13 Input Signals—TDI, TMS, TCK (LSI53C825AJ only)**

Symbol	Parameter	Min	Max	Unit	Test Conditions
$V_{IH}$	Input high voltage	3.85	$V_{DD} + 0.5$	V	–
$V_{IL}$	Input low voltage	$V_{SS} - 0.5$	1.35	V	–
$I_{IN}$	Input leakage	–800	–200	$\mu A$	–

**Table 6.14 Output Signal—TDO (LSI53C825AJ only)**

Symbol	Parameter	Min	Max	Unit	Test Conditions
$V_{OH}$	Output high voltage	$V_{DD} - 0.5$	$V_{DD}$	V	–4 mA
$V_{OL}$	Output low voltage	$V_{SS}$	0.5	V	4 mA
$I_{OZ}$	3-state leakage	–10	10	$\mu A$	–

---

## 6.2 TolerANT Technology Electrical Characteristics

The LSI53C825A 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 6.15](#) provides electrical characteristics for SE SCSI signals. [Figure 6.1](#) through [Figure 6.5](#) provide reference information for testing SCSI signals.

**Table 6.15 TolerANT Technology Electrical Characteristics**

Symbol	Parameter	Min	Max	Unit	Test Conditions
$V_{OH}^1$	Output high voltage	2.5	3.5	V	$I_{OH} = 2.5 \text{ mA}$
$V_{OL}$	Output low voltage	0.1	0.5	V	$I_{OL} = 48 \text{ mA}$
$V_{IH}$	Input high voltage	2.0	7.0	V	–
$V_{IL}$	Input low voltage	–0.5	0.8	V	Referenced to $V_{SS}$
$V_{IK}$	Input clamp voltage	–0.66	–0.77	V	$V_{DD} = 4.75$ ; $I_I = -20 \text{ mA}$
$V_{TH}$	Threshold, HIGH to LOW	1.1	1.3	V	–
$V_{TL}$	Threshold, LOW to HIGH	1.5	1.7	V	–
$V_{TH}-V_{TL}$	Hysteresis	200	400	mV	–
$I_{OH}^1$	Output high current	2.5	24	mA	$V_{OH} = 2.5 \text{ V}$
$I_{OL}$	Output low current	100	200	mA	$V_{OL} = 0.5 \text{ V}$
$I_{OSH}^1$	Short-circuit output high current	–	625	mA	Output driving low, pin shorted to $V_{DD}$ supply <sup>2</sup>
$I_{OSL}$	Short-circuit output low current	–	95	mA	Output driving high, pin shorted to $V_{SS}$ supply
$I_{LH}$	Input high leakage	–	10	$\mu\text{A}$	$-0.5 < V_{DD} < 5.25$ $V_{PIN} = 2.7 \text{ V}$
$I_{LL}$	Input low leakage	–	–10	$\mu\text{A}$	$-0.5 < V_{DD} < 5.25$ $V_{PIN} = 0.5 \text{ V}$
$R_I$	Input resistance	20	–	$\text{M}\Omega$	SCSI pins <sup>3</sup>
$C_P$	Capacitance per pin	–	10	pF	PQFP
$t_R^1$	Rise time, 10% to 90%	9.7	18.5	ns	<a href="#">Figure 6.1</a>
$t_F$	Fall time, 90% to 10%	5.2	14.7	ns	<a href="#">Figure 6.1</a>
$dV_H/dt$	Slew rate, LOW to HIGH	0.15	0.49	V/ns	<a href="#">Figure 6.1</a>
$dV_L/dt$	Slew rate, HIGH to LOW	0.19	0.67	V/ns	<a href="#">Figure 6.1</a>
ESD	Electrostatic discharge	2	–	kV	MIL-STD-883C; 3015-7
	Latch-up	100	–	mA	–
	Filter delay	20	30	ns	<a href="#">Figure 6.2</a>
	Extended filter delay	40	60	ns	<a href="#">Figure 6.2</a>

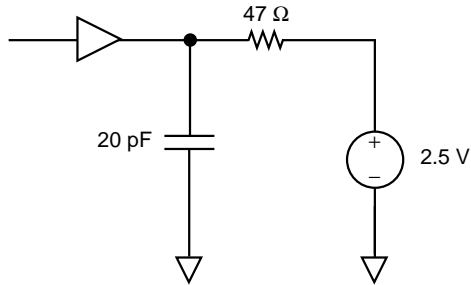
1. Active negation outputs only: Data, Parity, SREQ/, SACK/.

2. Single pin only; irreversible damage may occur if sustained for one second.

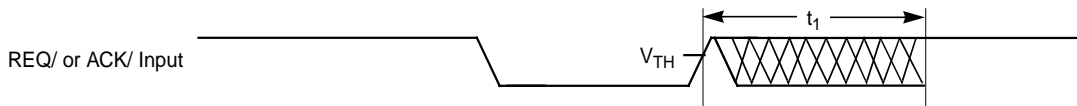
3. SCSI RESET pin has 10 k $\Omega$  pull-up resistor.

Note: These values are guaranteed by periodic characterization; they are not 100% tested on every device.

**Figure 6.1 Rise and Fall Time Test Conditions**

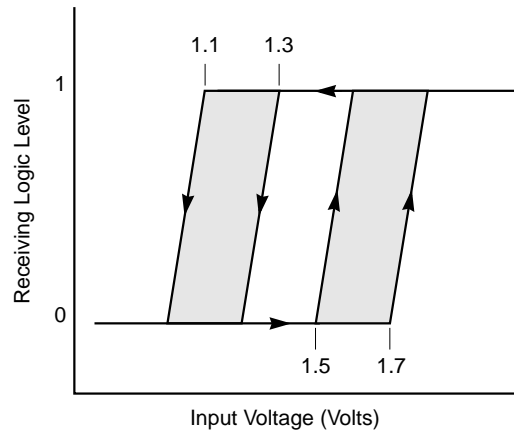


**Figure 6.2 SCSI Input Filtering**

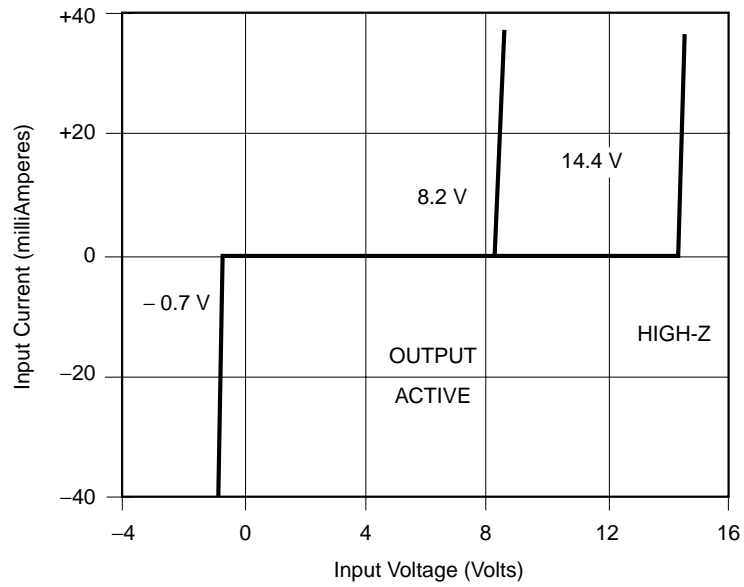


Note:  $t_1$  is the input filtering period.

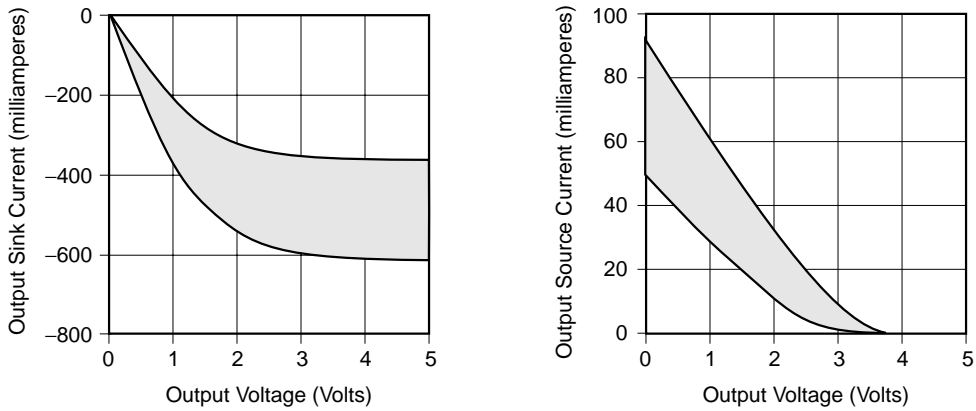
**Figure 6.3 Hysteresis of SCSI Receivers**



**Figure 6.4 Input Current as a Function of Input Voltage**



**Figure 6.5 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 [Section 6.1, “DC Characteristics”](#)). Chip timings are based on simulation at worst case voltage, temperature, and processing. Timing was developed with a load capacitance of 50 pF. [Table 6.16](#) and [Figure 6.6](#) provide External Clock timing data.

**Table 6.16 External Clock**

Symbol	Parameter	Min	Max	Unit
$t_1$	Bus clock cycle time	30	DC	ns
	SCSI clock cycle time (SCLK) <sup>1</sup>	12.5	60	ns
$t_2$	CLK LOW time <sup>2</sup>	12	—	ns
	SCLK LOW time <sup>2</sup>	5	—	ns
$t_3$	CLK HIGH time <sup>2</sup>	12	—	ns
	SCLK HIGH time <sup>2</sup>	5	—	ns
$t_4$	CLK slew rate	1	—	V/ns
	SCLK slew rate	1	—	V/ns

1. This parameter must be met to ensure SCSI timings are within specification.
2. Duty cycle not to exceed 60/40.

**Figure 6.6 External Clock**

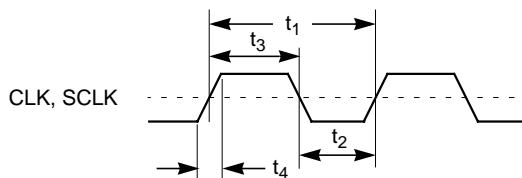
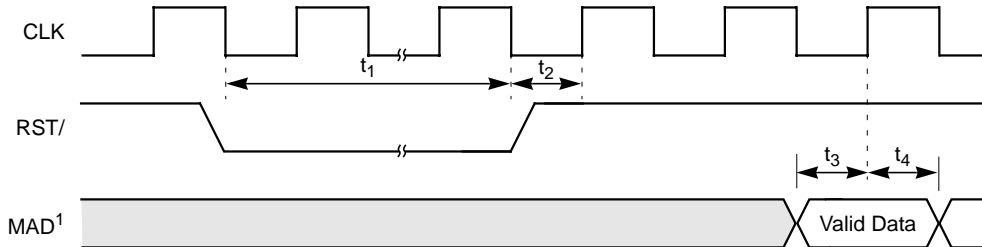


Table 6.17 and Figure 6.7 provide Reset Input timing data.

**Table 6.17 Reset Input**

Symbol	Parameter	Min	Max	Unit
$t_1$	Reset pulse width	10	—	$t_{\text{CLK}}$
$t_2$	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 6.7 Reset Input**



1. When enabled.

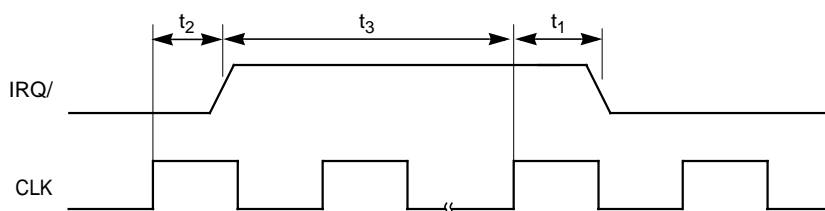


Table 6.18 and Figure 6.8 provide Interrupt Output timing data.

**Table 6.18 Interrupt Output**

Symbol	Parameter	Min	Max	Unit
$t_1$	CLK HIGH to IRQ/ LOW	20	—	ns
$t_2$	CLK HIGH to IRQ/ HIGH	40	—	ns
$t_3$	IRQ/ deassertion time	3	—	CLK

**Figure 6.8 Interrupt Output**



## 6.4 PCI and External Memory Interface Timing Diagrams

Figure 6.9 through Figure 6.30 represent signal activity when the LSI53C825A accesses the PCI bus. The timings for the PCI and external memory buses are listed on [page 6-44](#). This section includes timing diagrams for access to three groups of external memory configurations. The first group applies to systems with memory size of 64 Kbytes and above; one byte read or write cycle, and fast or normal ROMs. The second group applies to systems with memory size of 64 Kbytes and above, one byte read or write cycles, and slow ROMs. The third group applies to systems with memory size of 64 Kbytes or less, one byte read or write cycles, and normal or fast ROM.

**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:

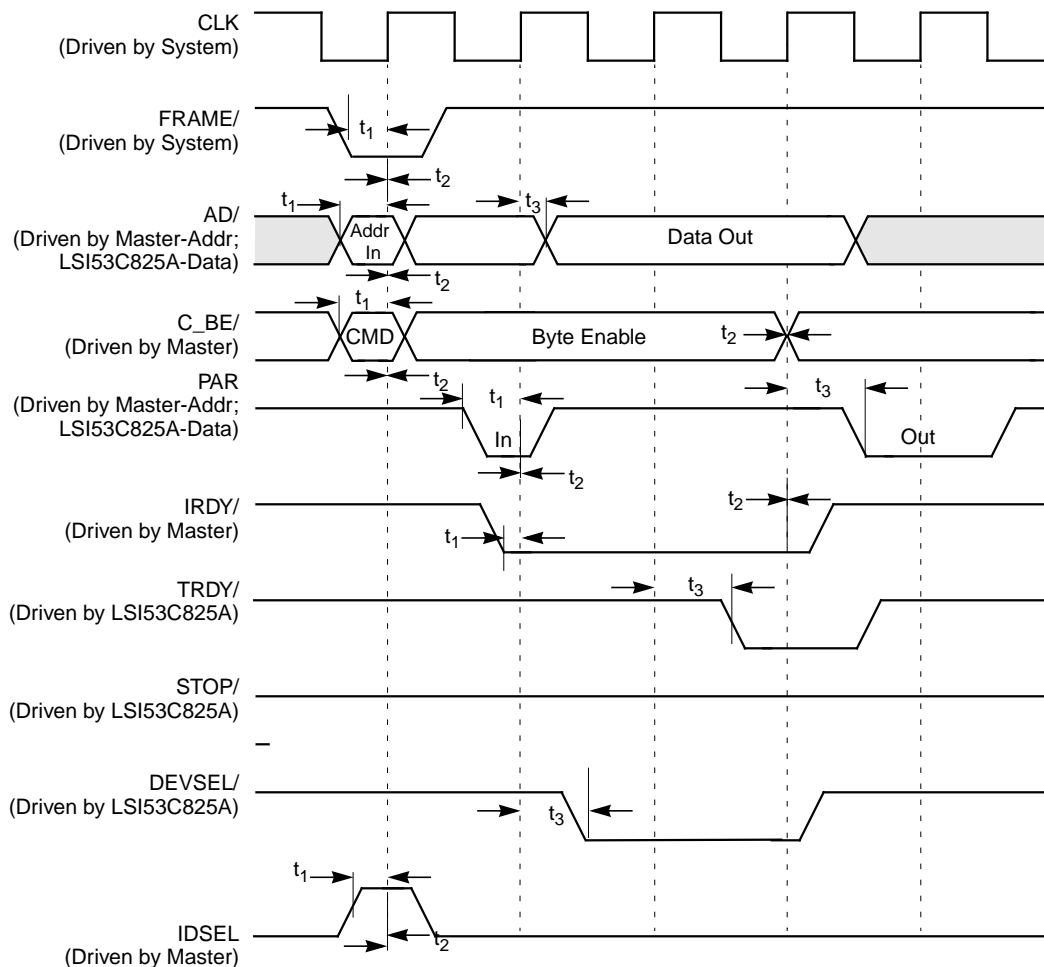
- [Target Timing](#)
  - [PCI Configuration Register Read](#)

- PCI Configuration Register Write
- Operating Register/SCRIPTS RAM Read
- Operating Register/SCRIPTS RAM Write
- External Memory Read
- External Memory Write
- Initiator Timing
  - Nonburst Opcode Fetch
  - Burst Opcode Fetch
  - Back-to-Back Read
  - Back-to-Back Write
  - Burst Read
  - Burst Write
- External Memory Timing
  - Read Cycle, Normal/Fast Memory ( $\geq 64$  Kbytes), Single Byte Access
  - Write Cycle, Normal/Fast Memory ( $\geq 64$  Kbytes), Single Byte Access
  - Read Cycle, Normal/Fast Memory ( $\geq 64$  Kbytes), Multiple Byte Access
  - Write Cycle, Normal/Fast Memory ( $\geq 64$  Kbytes), Multiple Byte Access
  - Read Cycle, Slow Memory ( $\geq 64$  Kbytes)
  - Write Cycle, Slow Memory ( $\geq 64$  Kbytes)
  - Read Cycle, Normal/Fast Memory ( $\geq 64$  Kbytes)
  - Write Cycle, Normal/Fast Memory ( $\geq 64$  Kbytes)
  - Read Cycle, Slow Memory ( $\leq 64$  Kbytes)
  - Write Cycle, Slow Memory ( $\leq 64$  Kbytes)

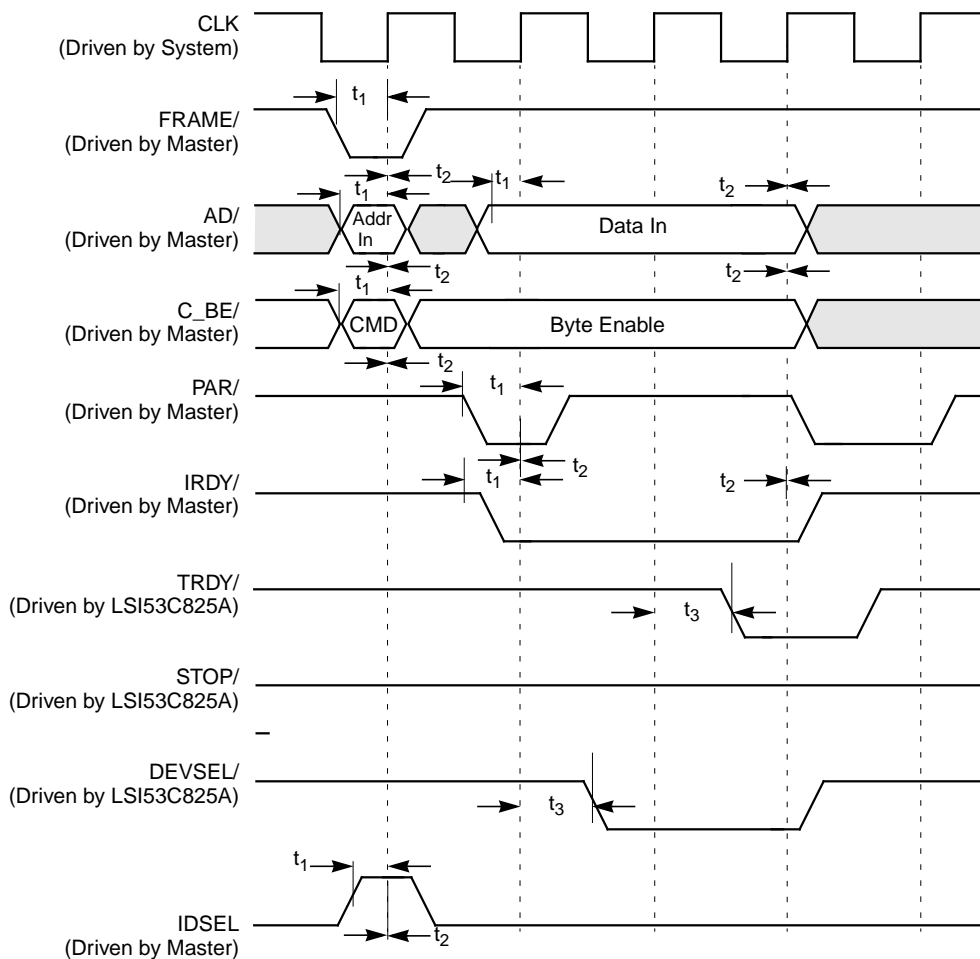
## 6.4.1 Target Timing

Figure 6.9 through Figure 6.14 describe Target timing.

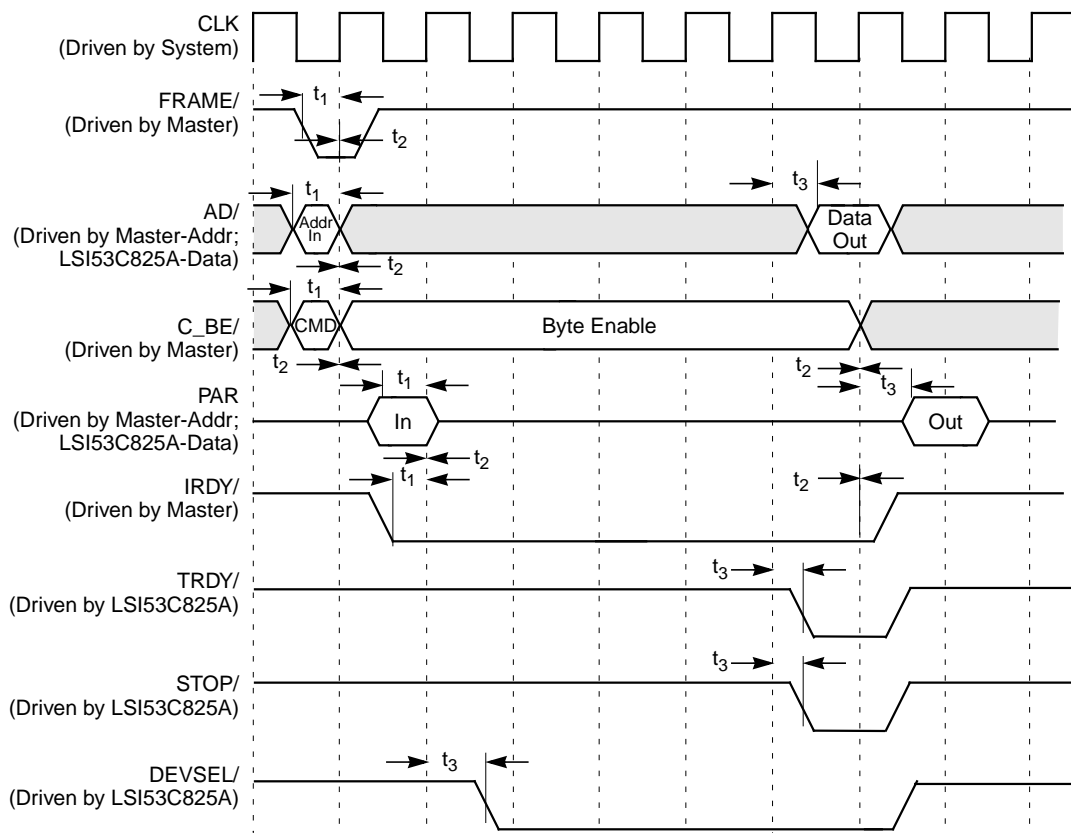
**Figure 6.9 PCI Configuration Register Read**



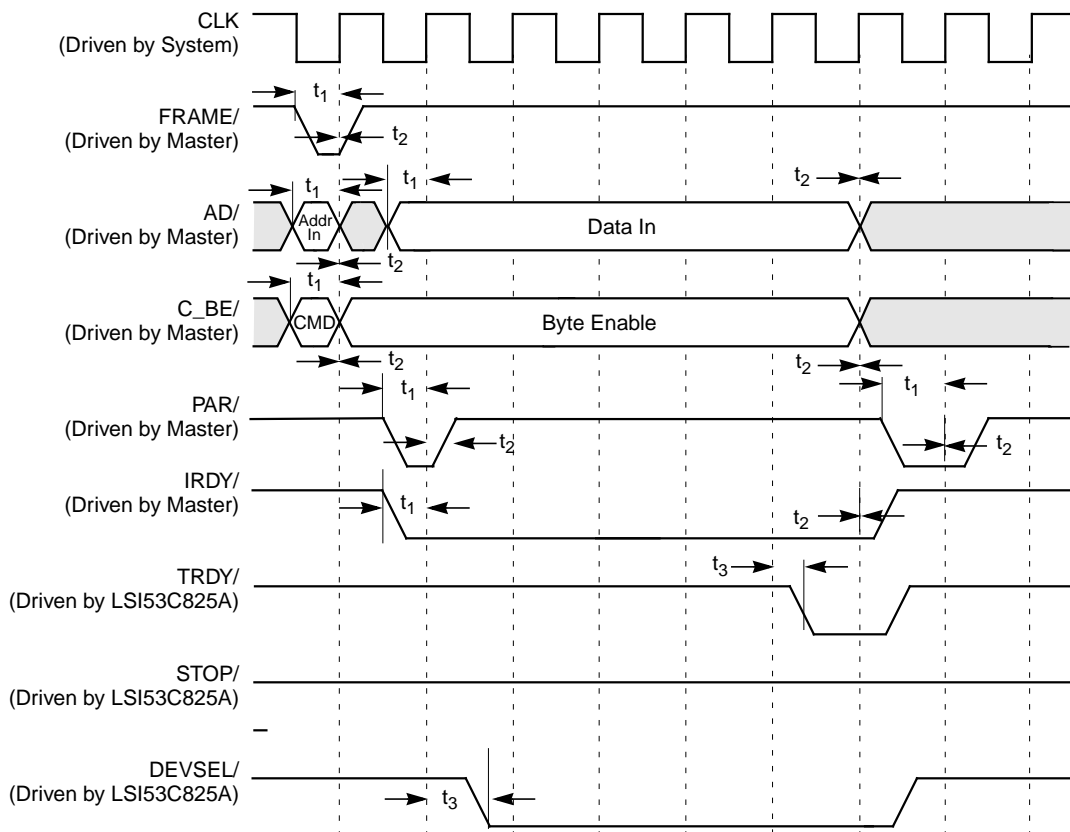
**Figure 6.10 PCI Configuration Register Write**



**Figure 6.11 Operating Register/SCRIPTS RAM Read**

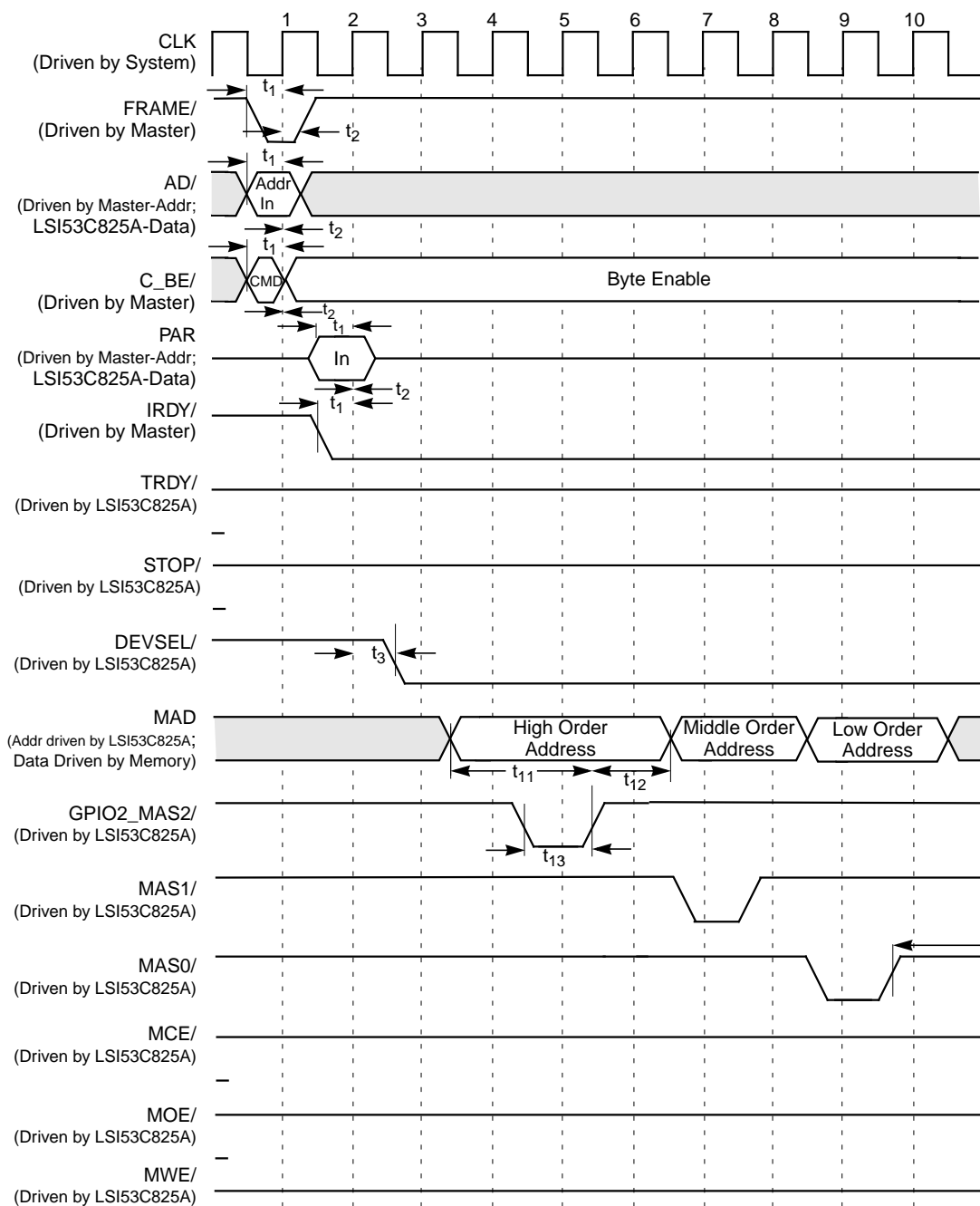


**Figure 6.12 Operating Register/SCRIPTS RAM Write**



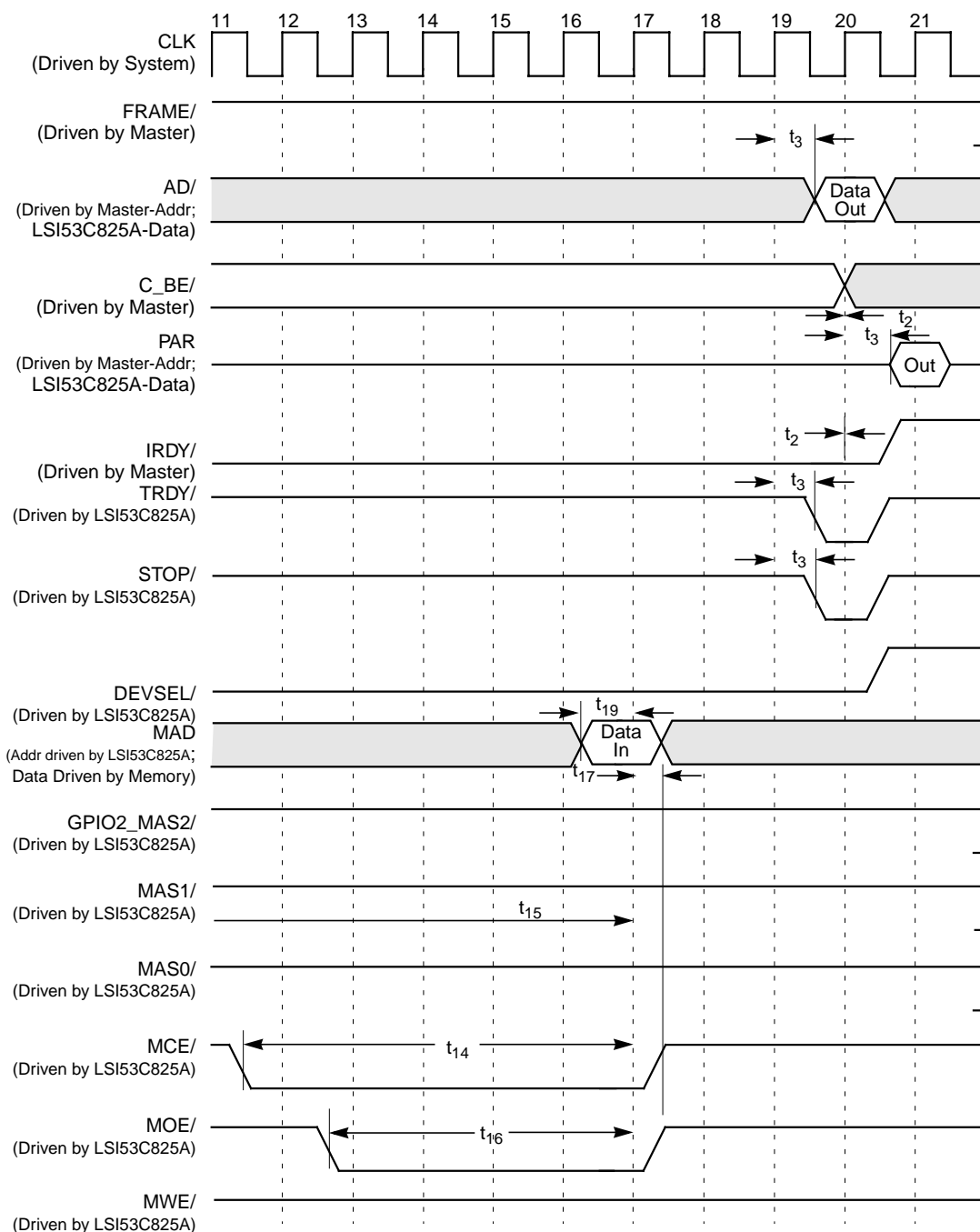
This page intentionally left blank.

**Figure 6.13 External Memory Read**

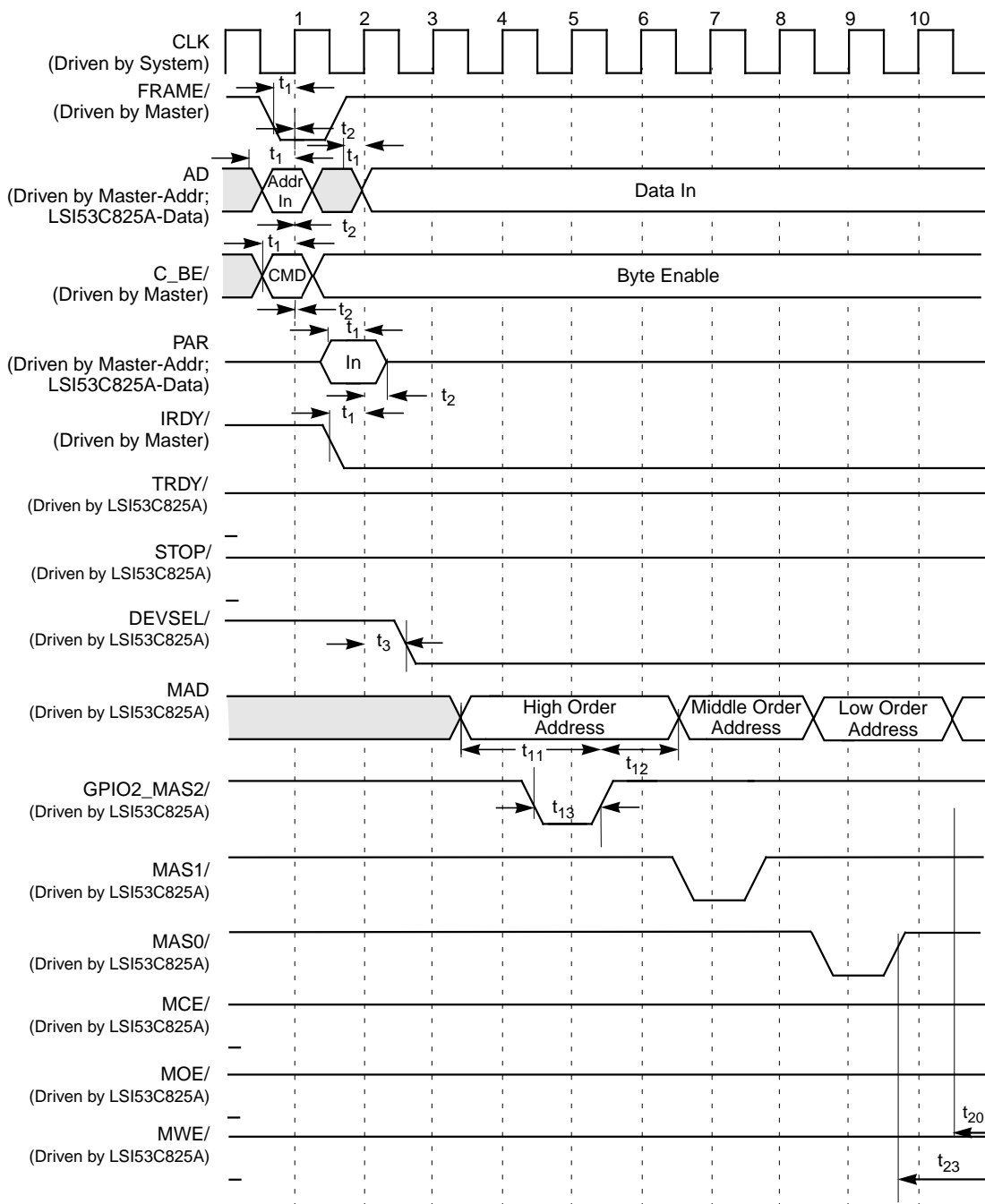




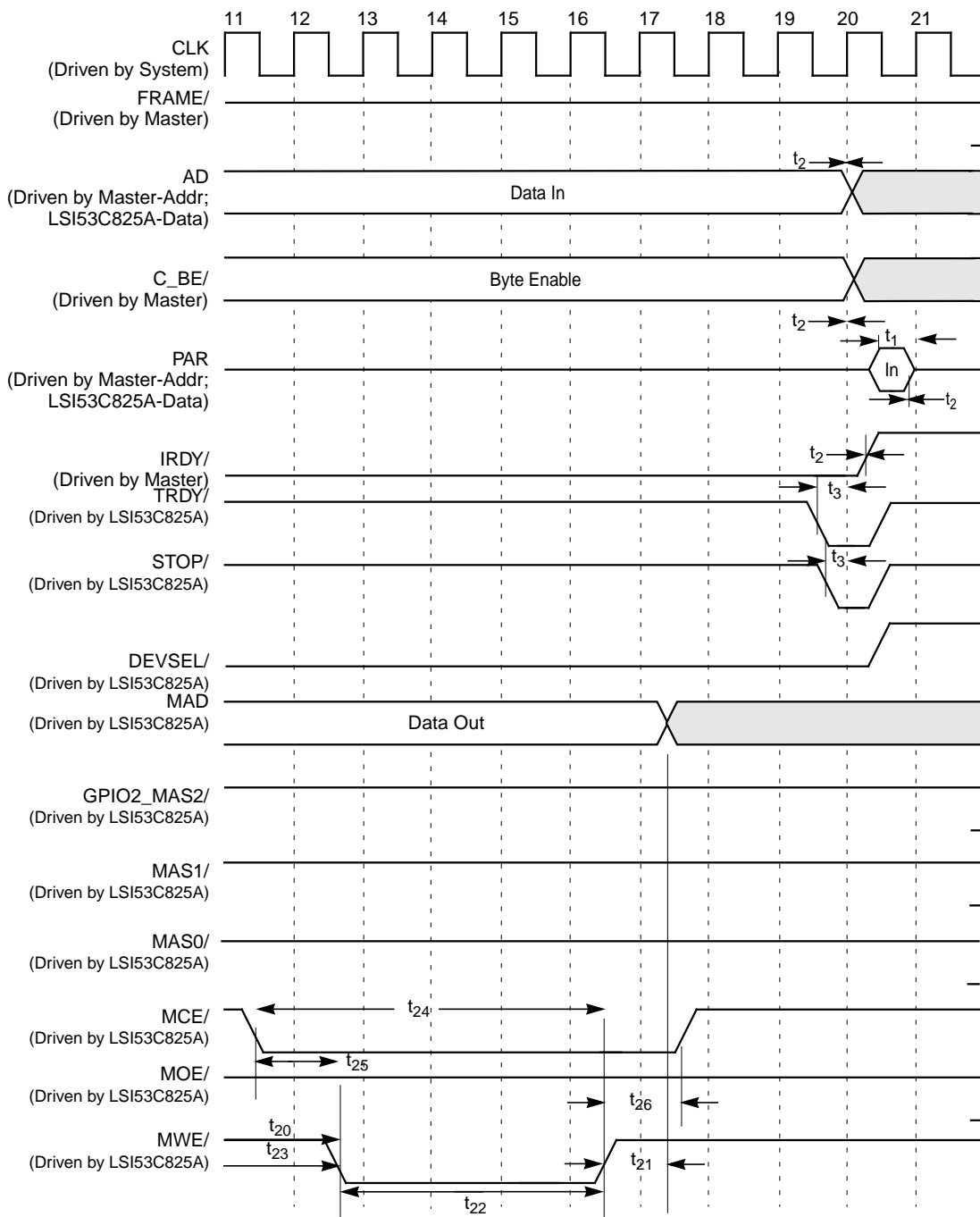
**Figure 6.13 External Memory Read (Cont.)**



**Figure 6.14 External Memory Write**



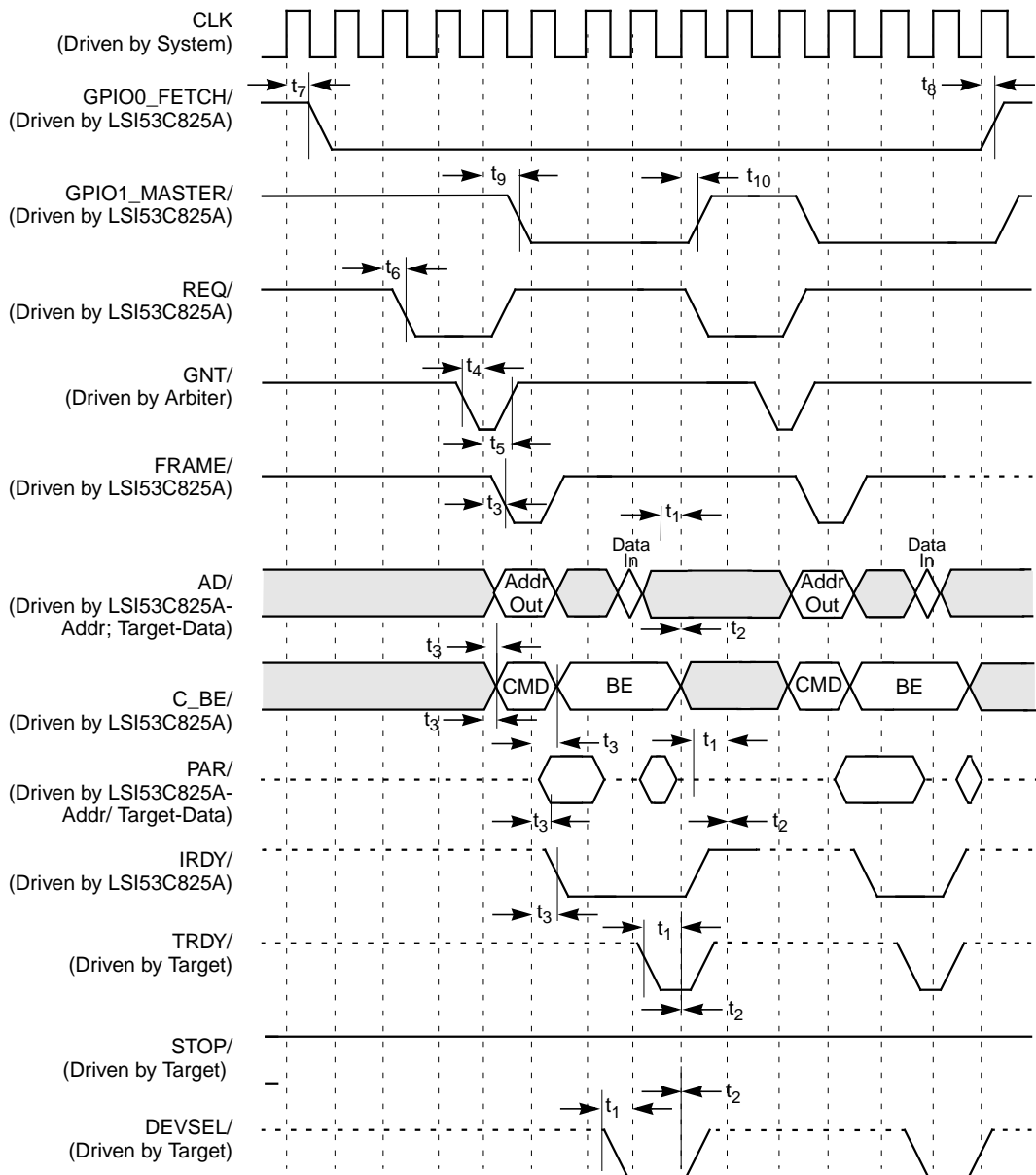
**Figure 6.14 External Memory Write (Cont.)**



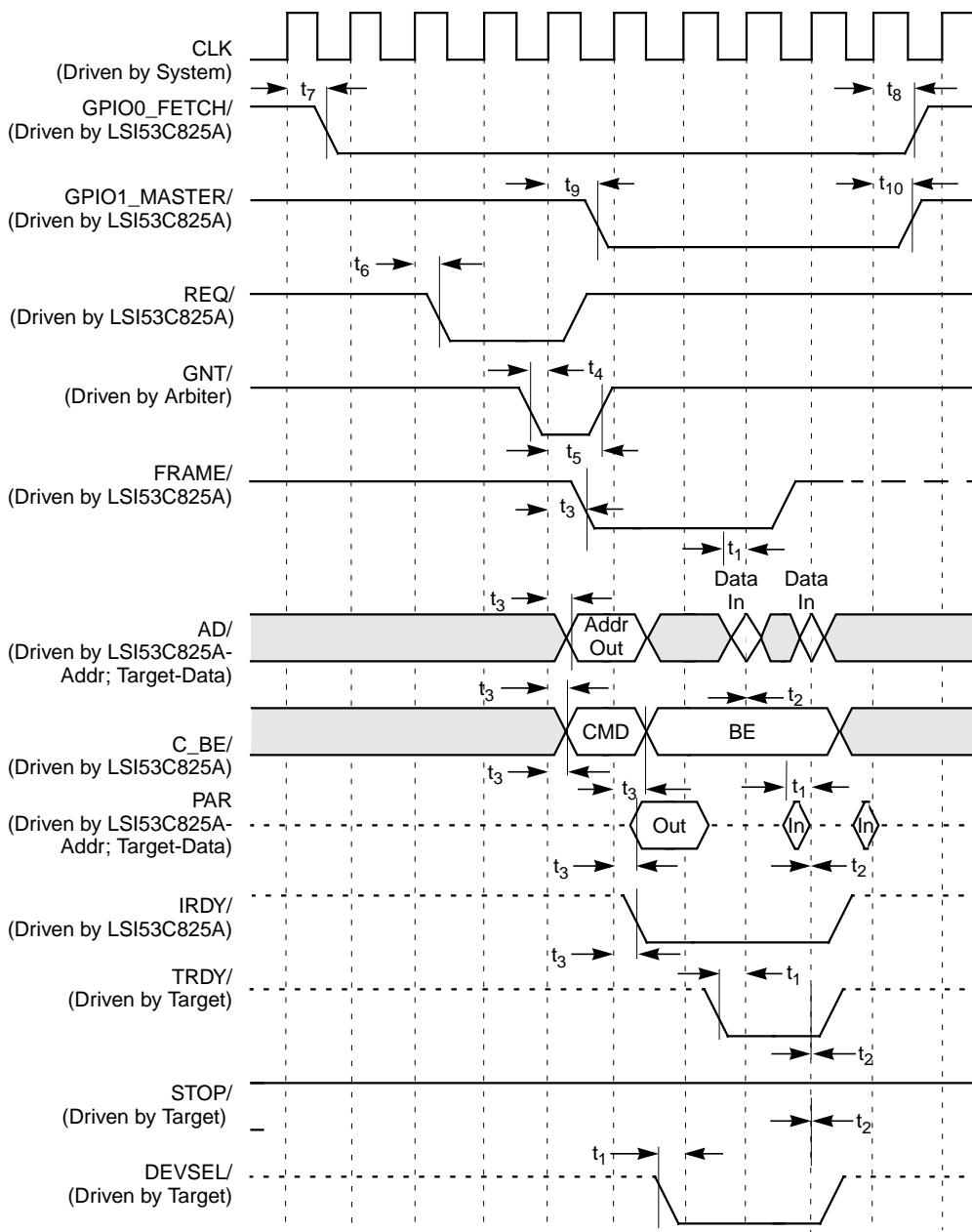
## 6.4.2 Initiator Timing

Figure 6.15 through Figure 6.20 describe LSI53C825A Initiator timing.

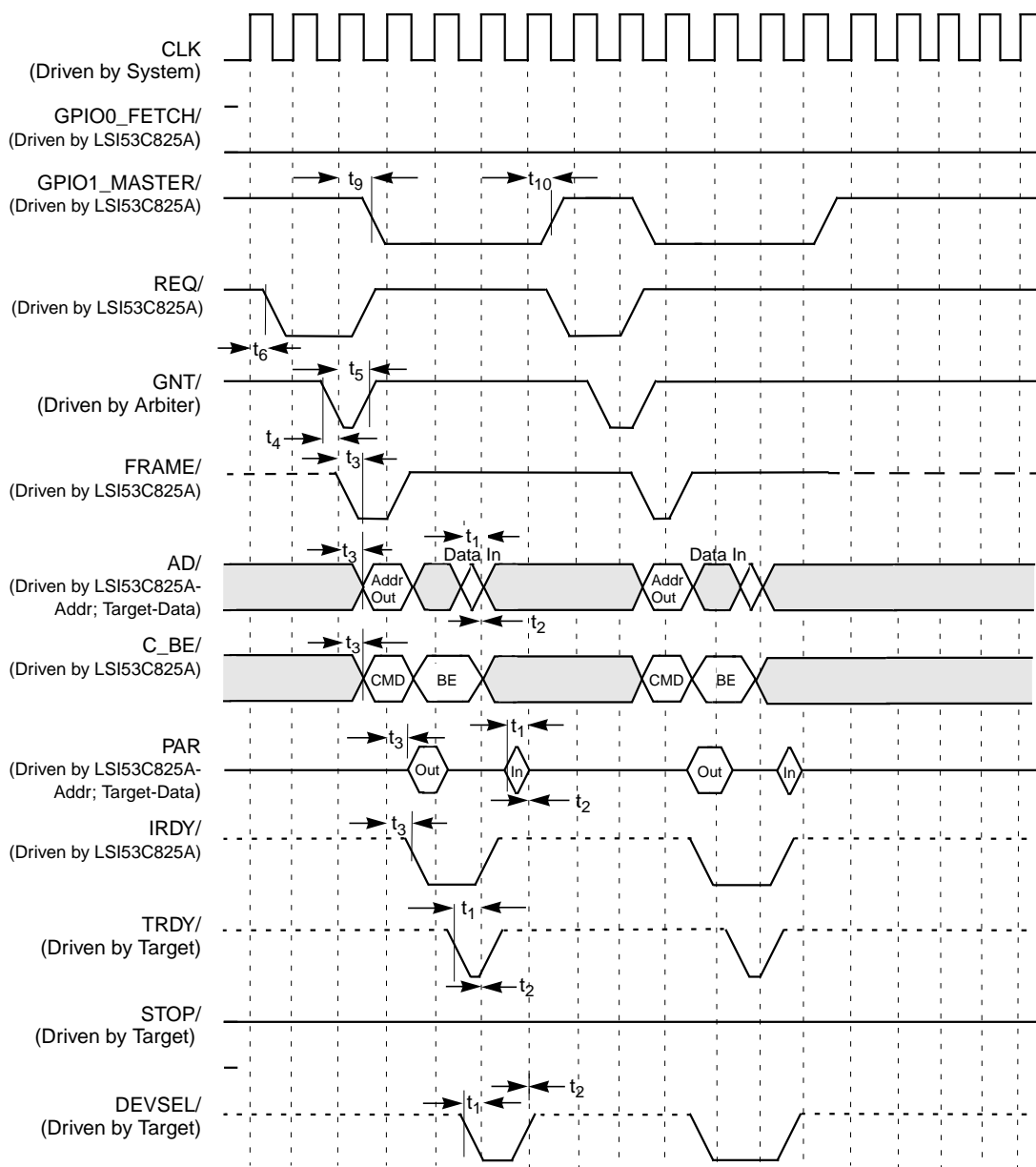
**Figure 6.15 Nonburst Opcode Fetch**



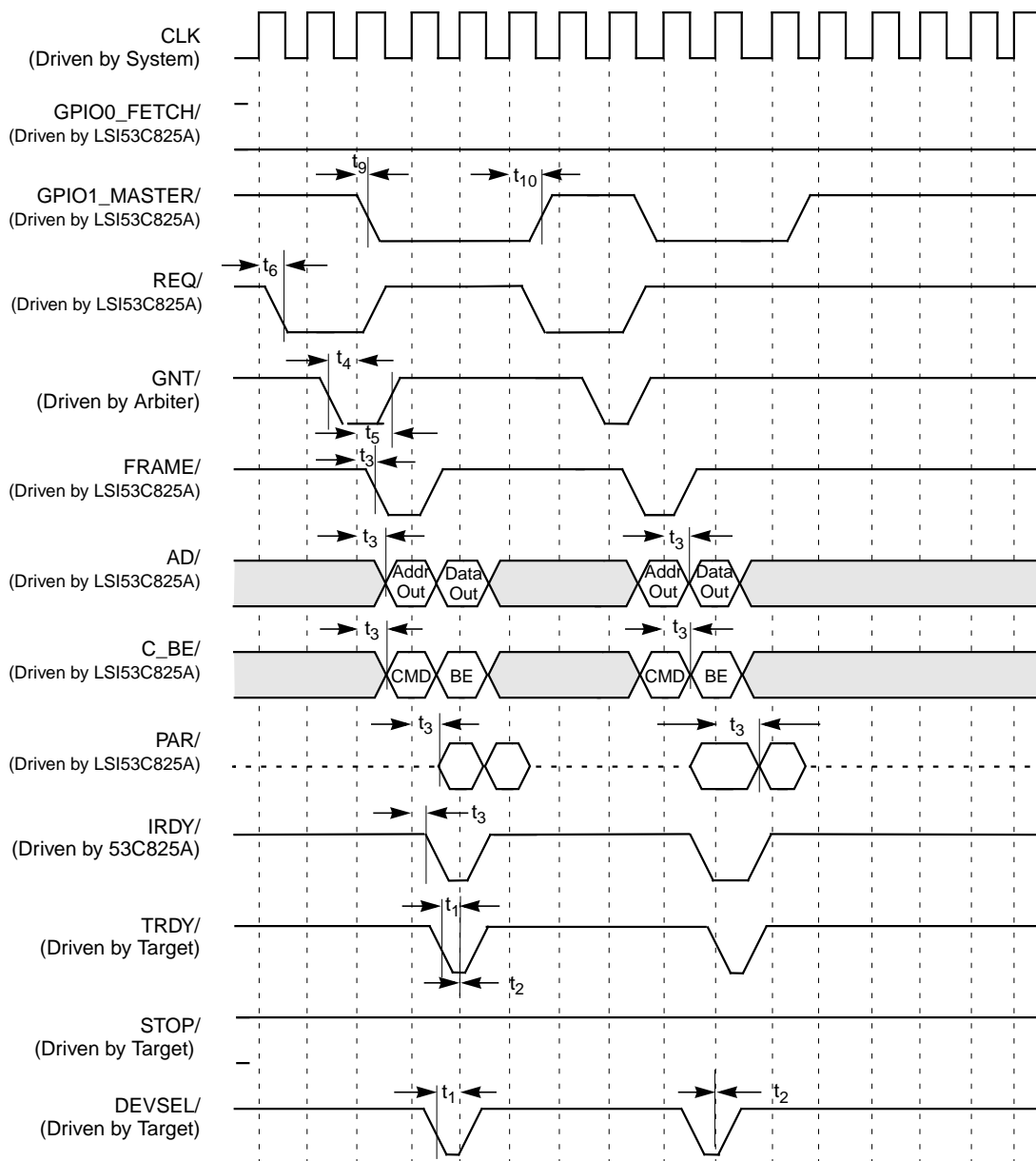
**Figure 6.16 Burst Opcode Fetch**



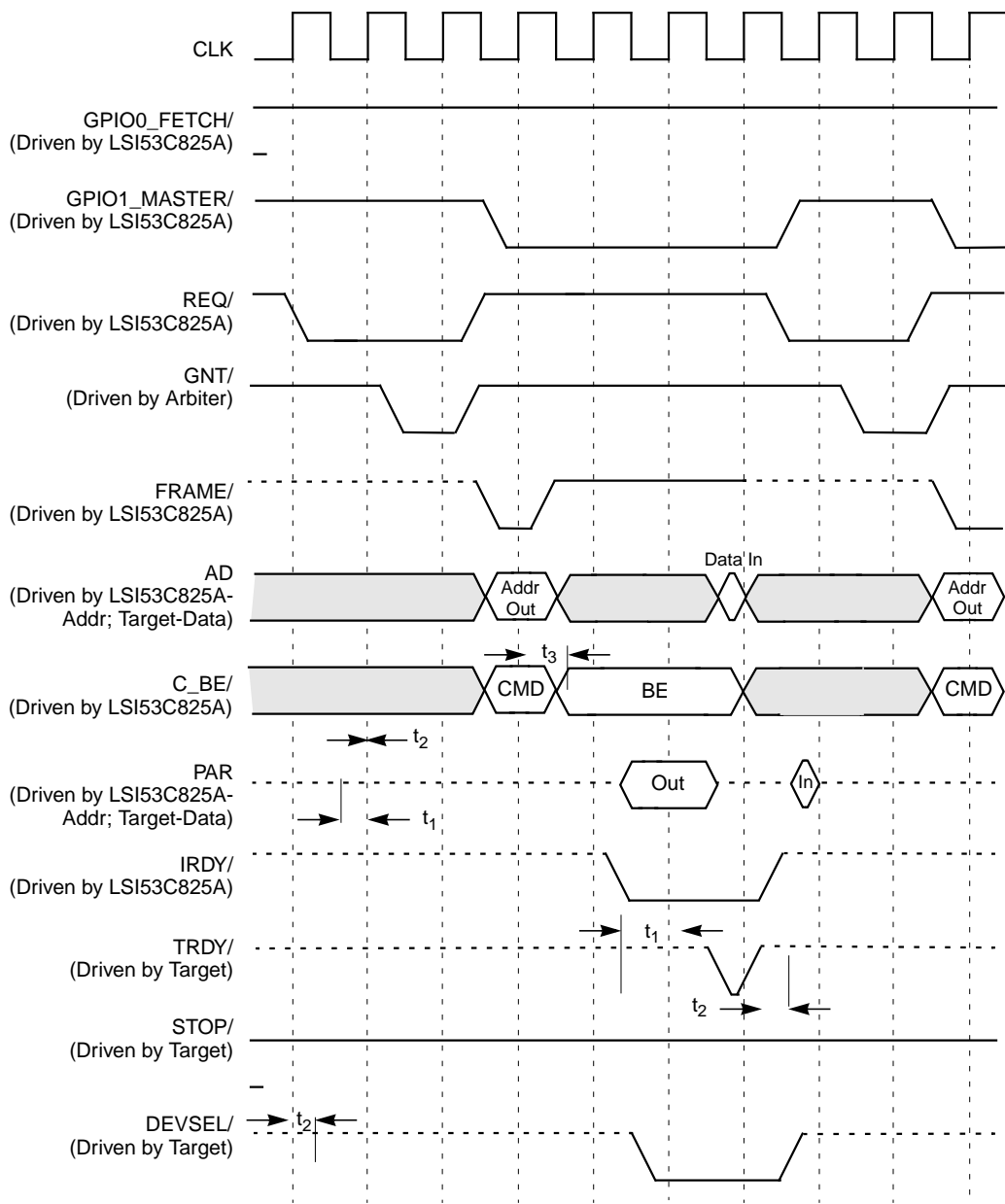
**Figure 6.17 Back-to-Back Read**



**Figure 6.18 Back-to-Back Write**

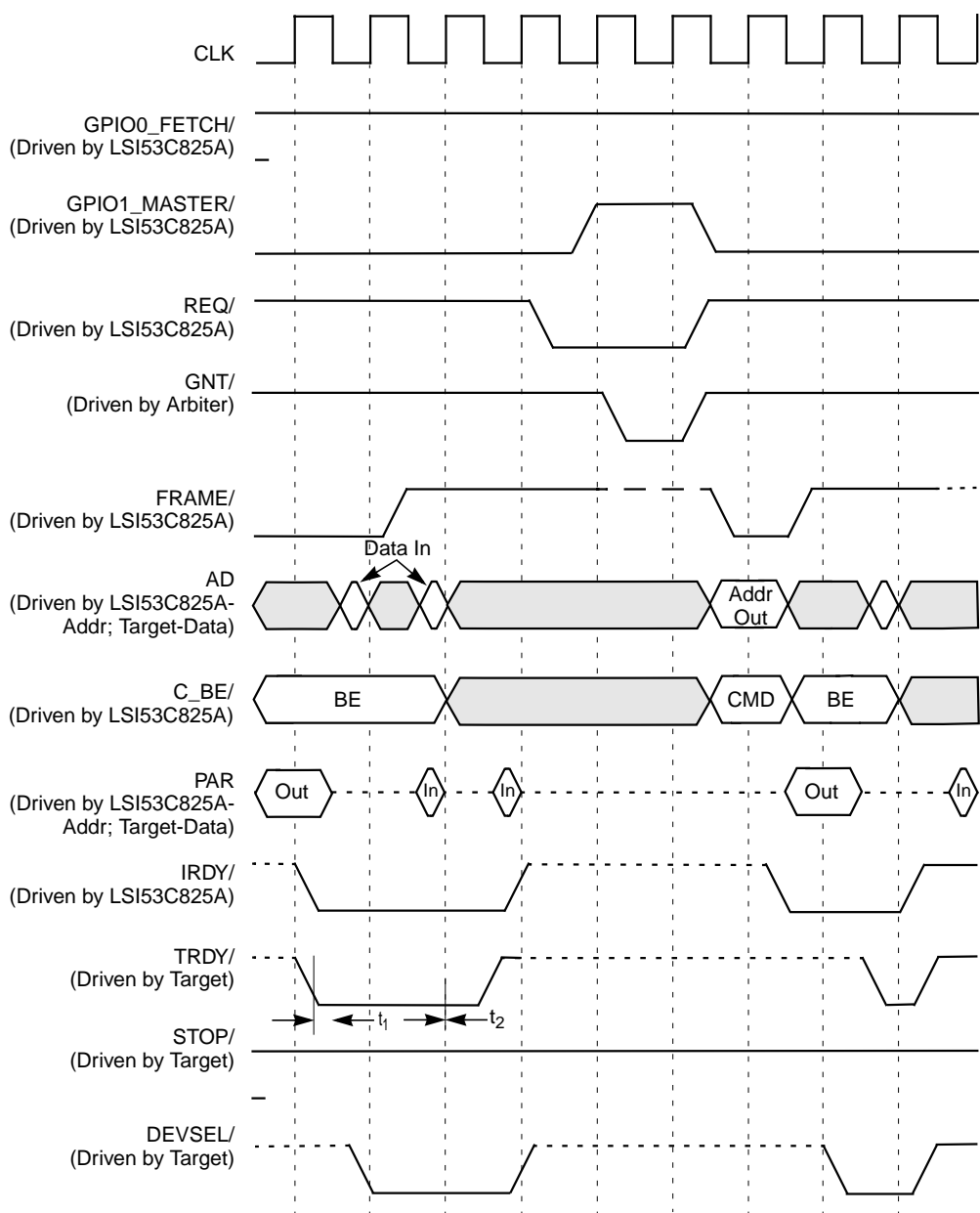


**Figure 6.19 Burst Read**

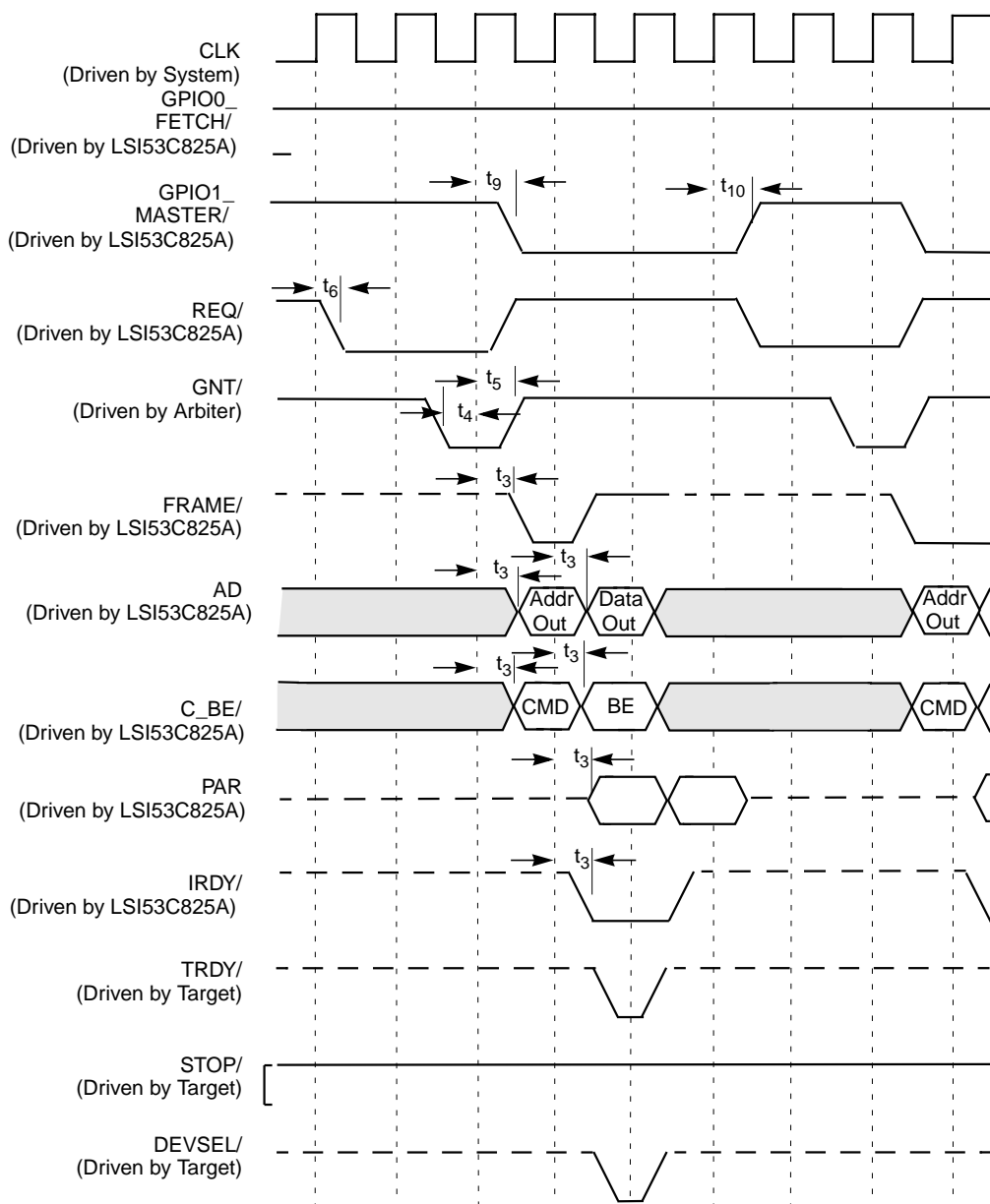




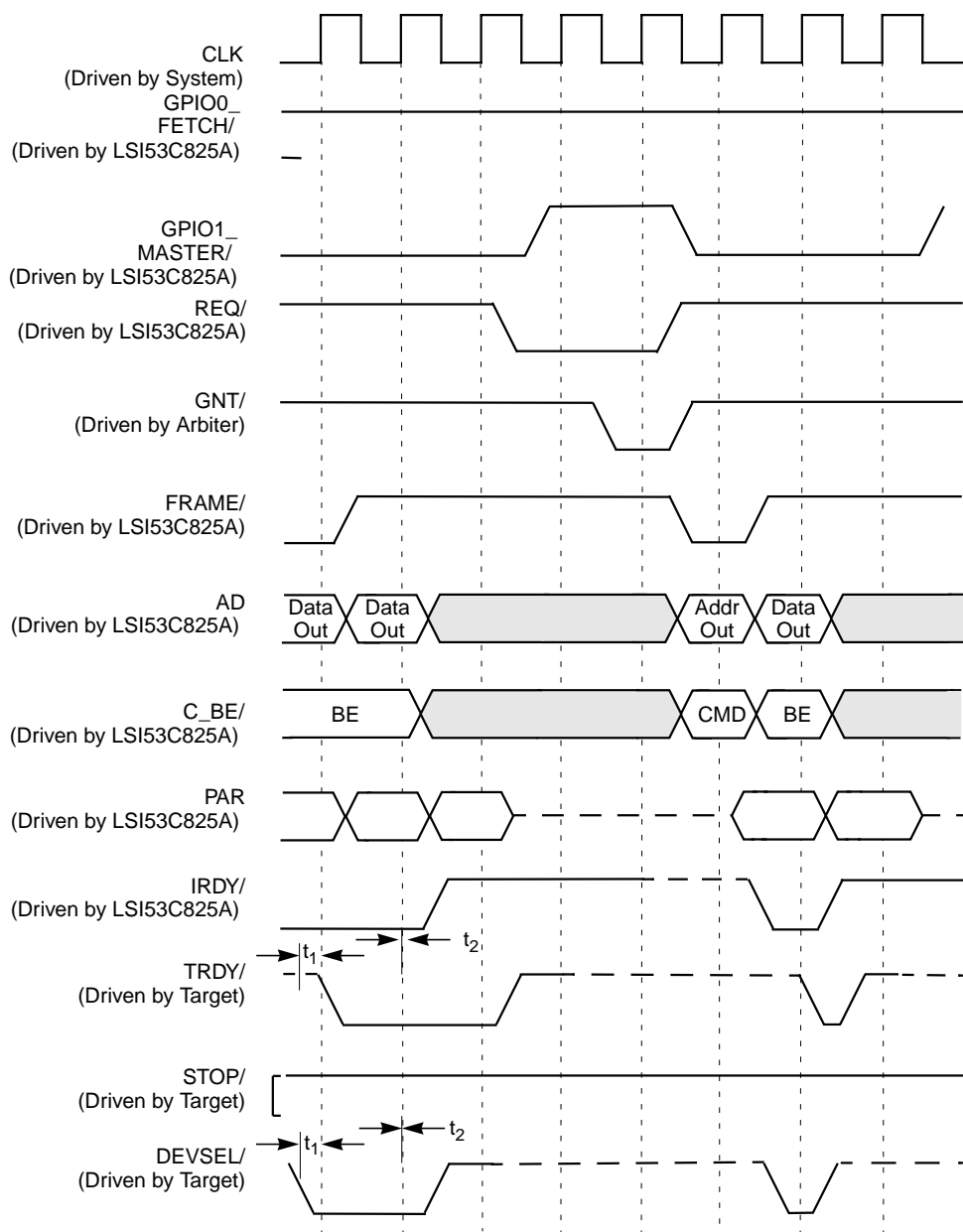
**Figure 6.19 Burst Read (Cont.)**



**Figure 6.20 Burst Write**



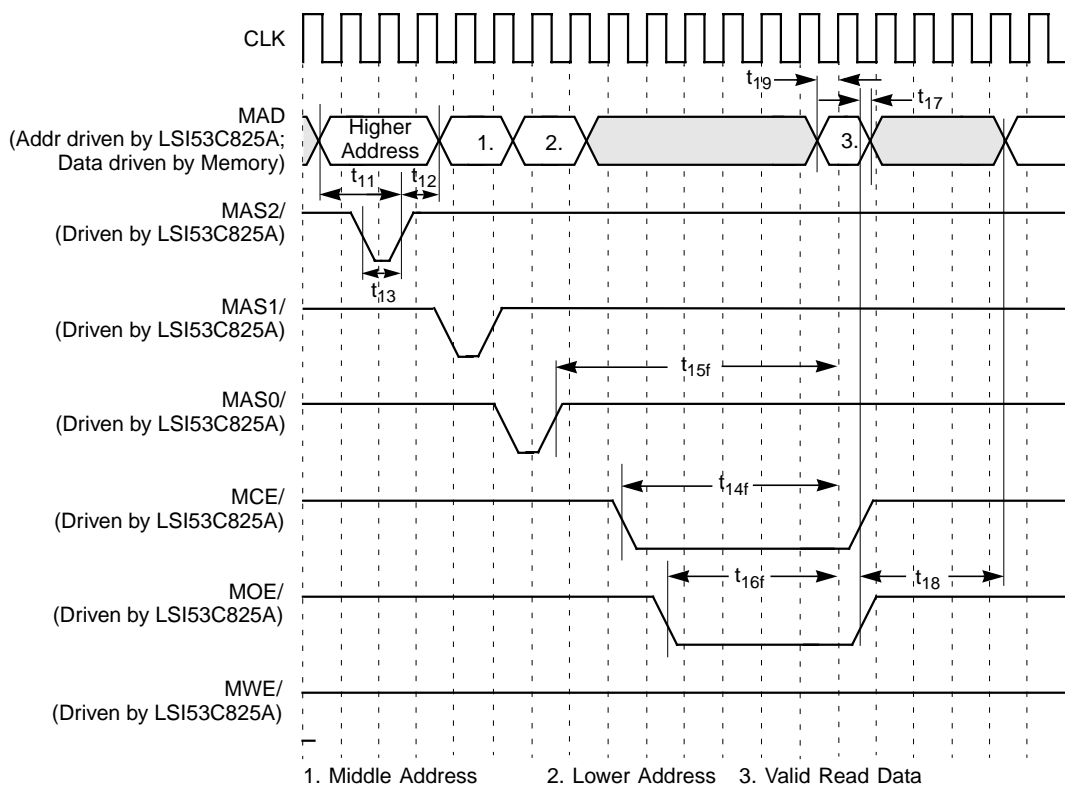
**Figure 6.20 Burst Write (Cont.)**



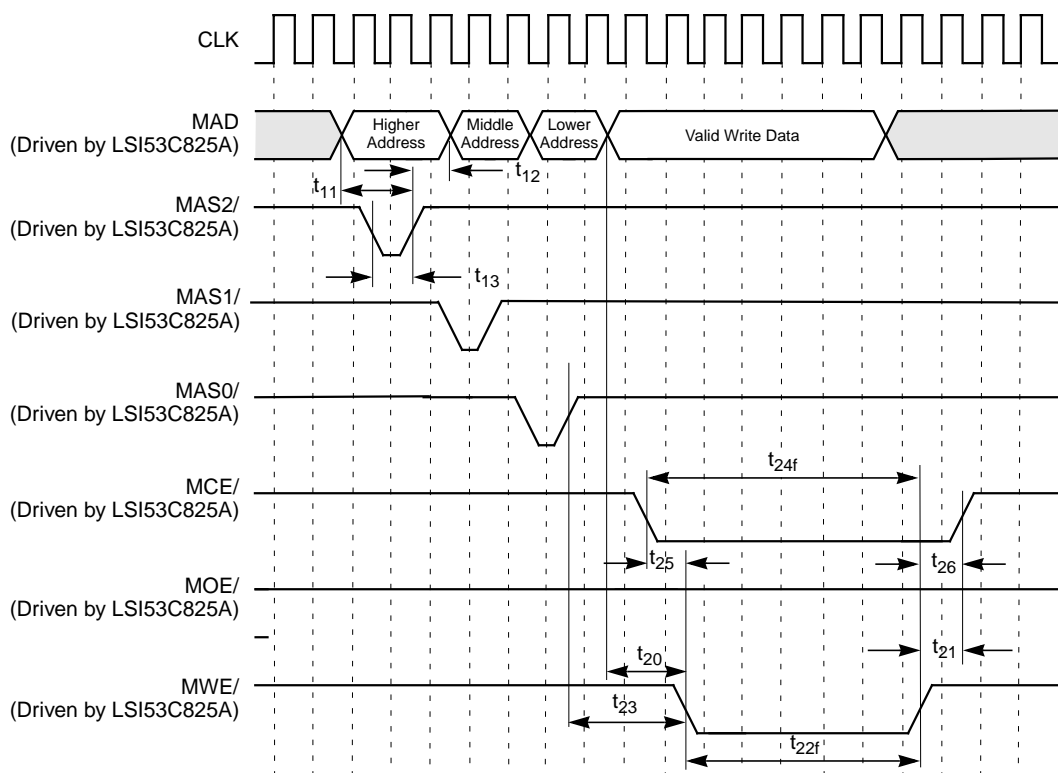
## 6.4.3 External Memory Timing

Figure 6.21 through Figure 6.30 describe LSI53C825A External Memory timing.

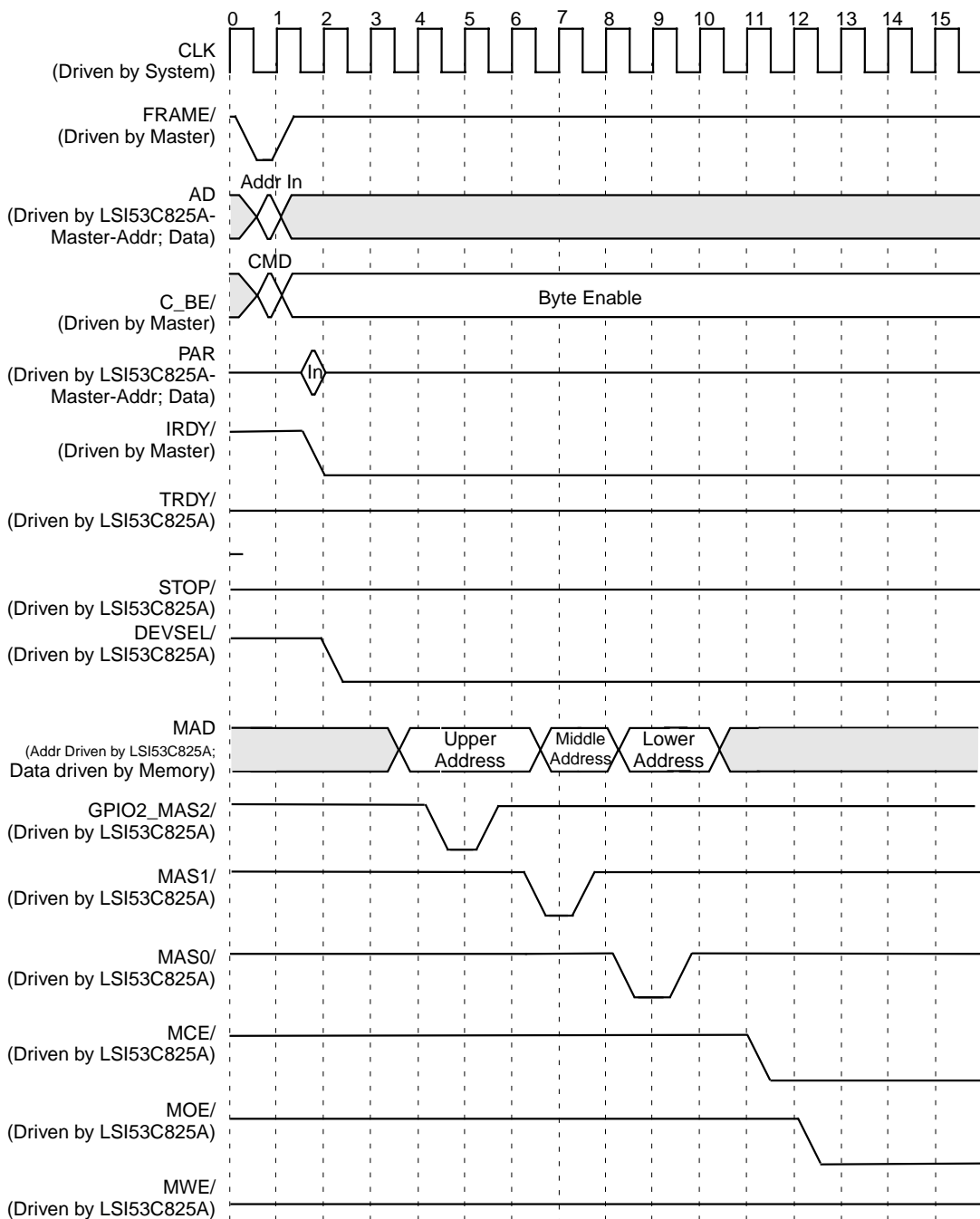
**Figure 6.21 Read Cycle, Normal/Fast Memory ( $\geq 64$  Kbytes), Single Byte Access**



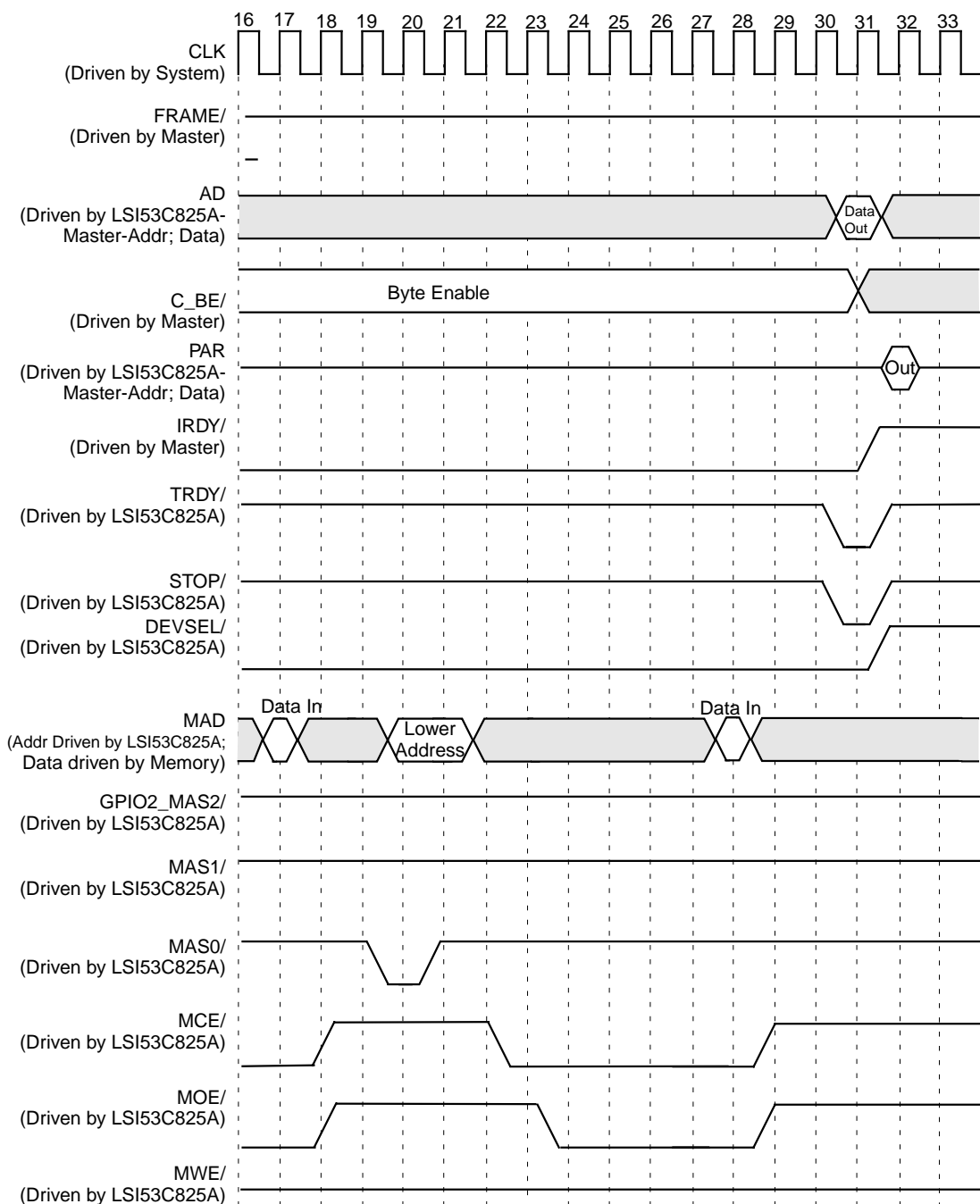
**Figure 6.22 Write Cycle, Normal/Fast Memory ( $\geq 64$  Kbytes), Single Byte Access**



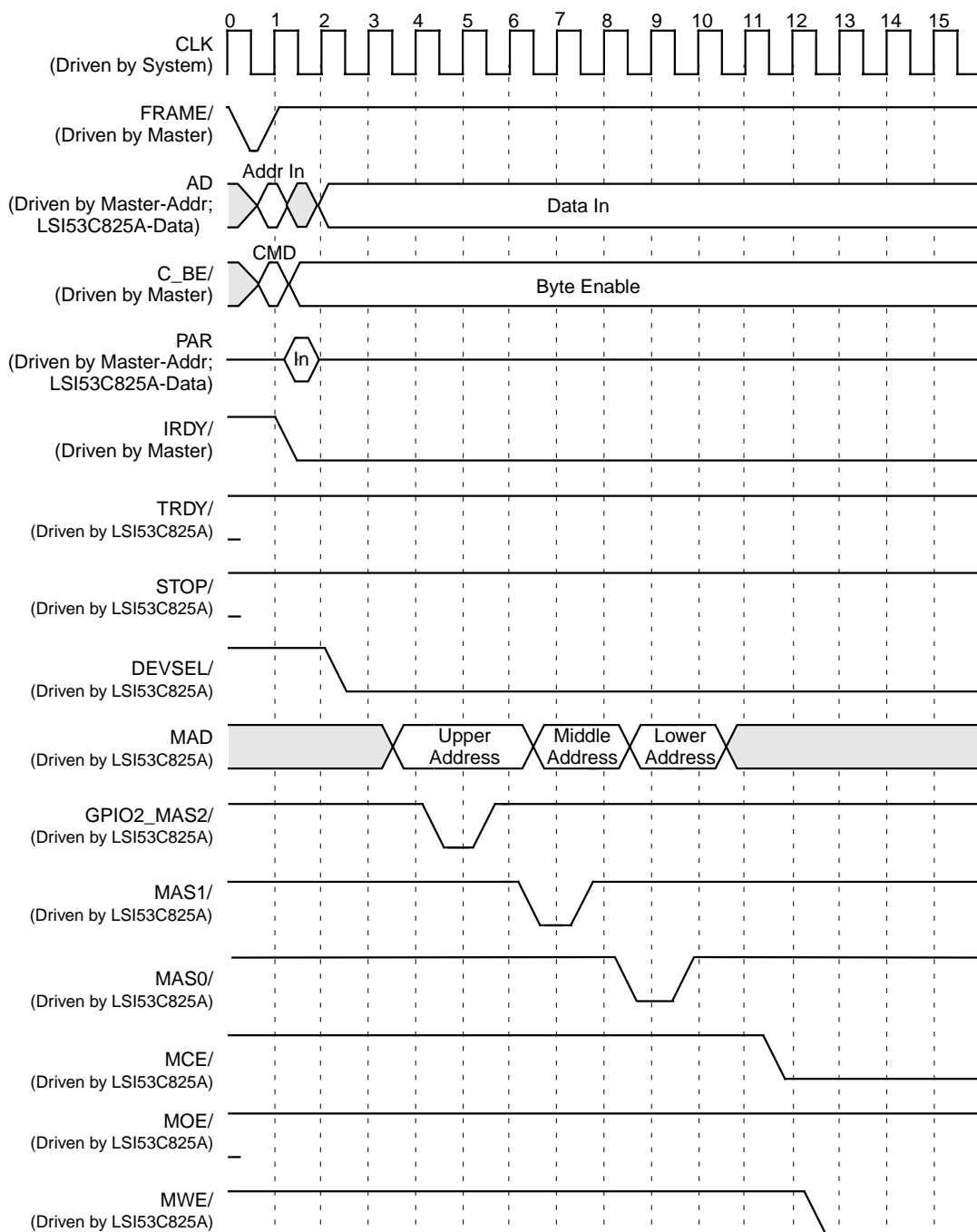
**Figure 6.23 Read Cycle, Normal/Fast Memory ( $\geq 64$  Kbytes), Multiple Byte Access**



**Figure 6.23 Read Cycle, Normal/Fast Memory ( $\geq 64$  Kbytes), Multiple Byte Access (Cont.)**

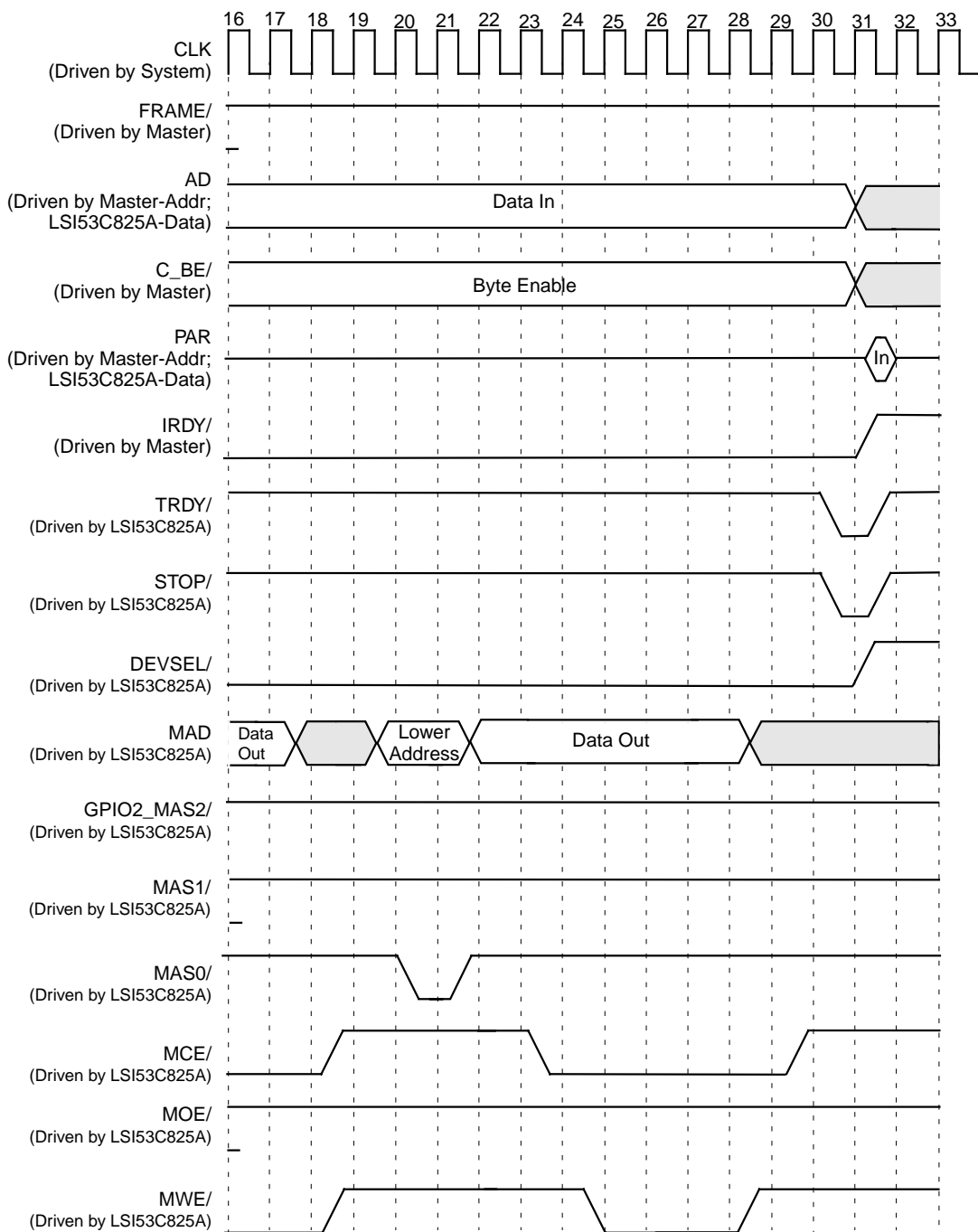


**Figure 6.24 Write Cycle, Normal/Fast Memory ( $\geq 64$  Kbytes), Multiple Byte Access**

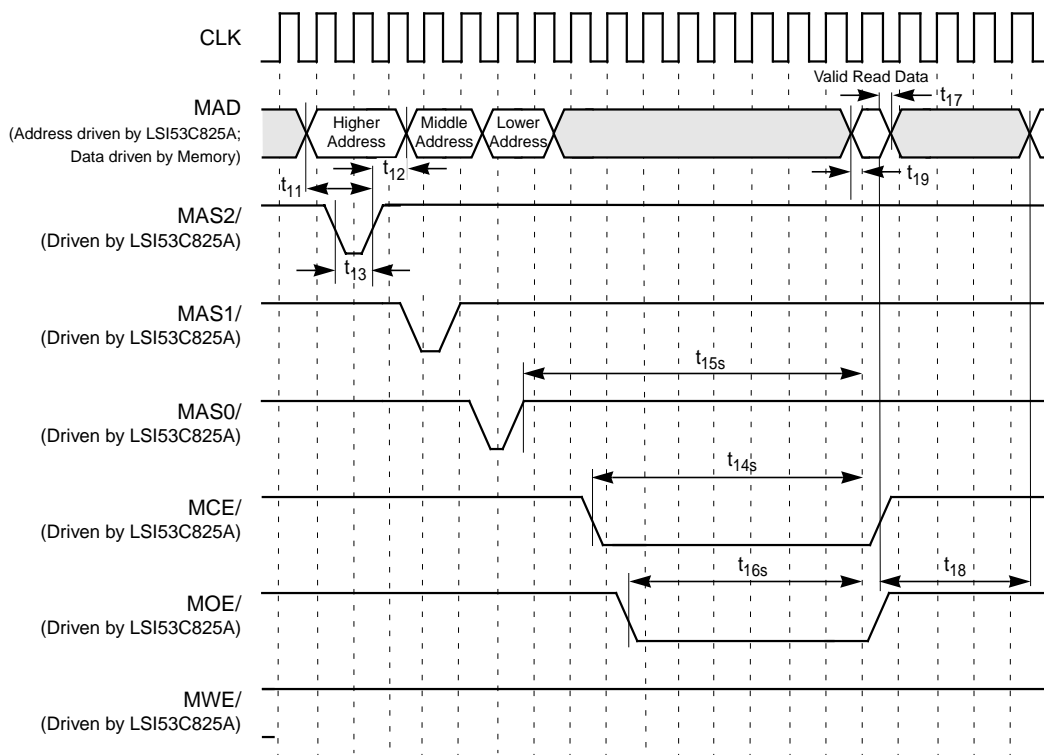




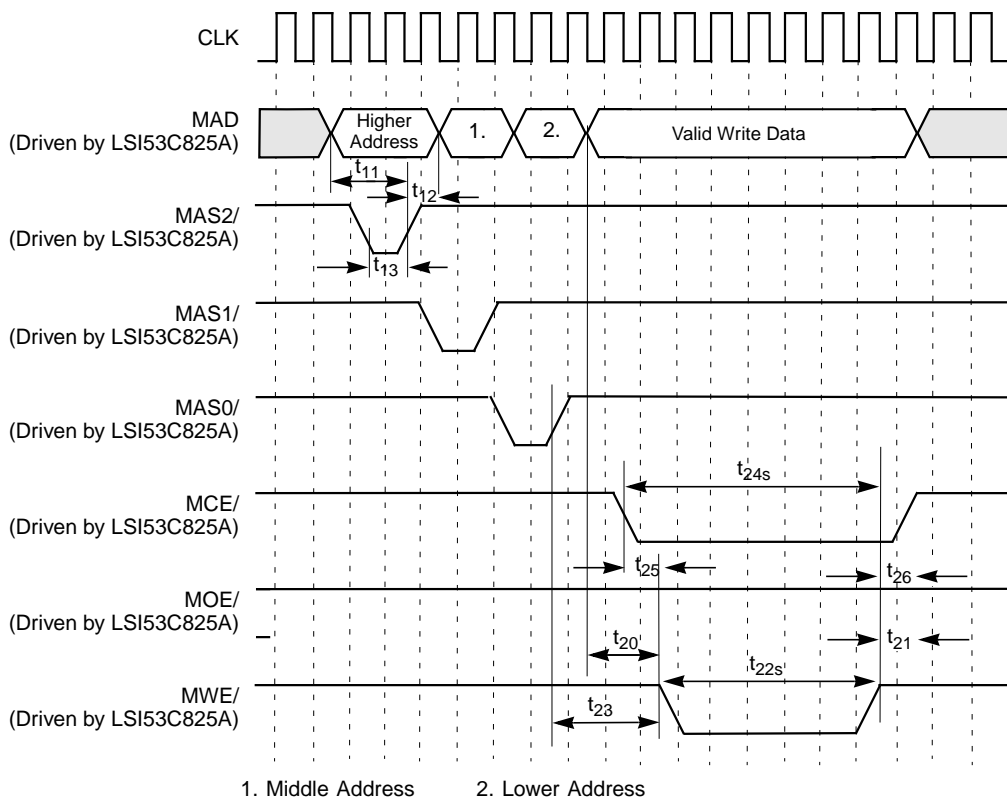
**Figure 6.24 Write Cycle, Normal/Fast Memory ( $\geq 64$  Kbytes), Multiple Byte Access (Cont.)**



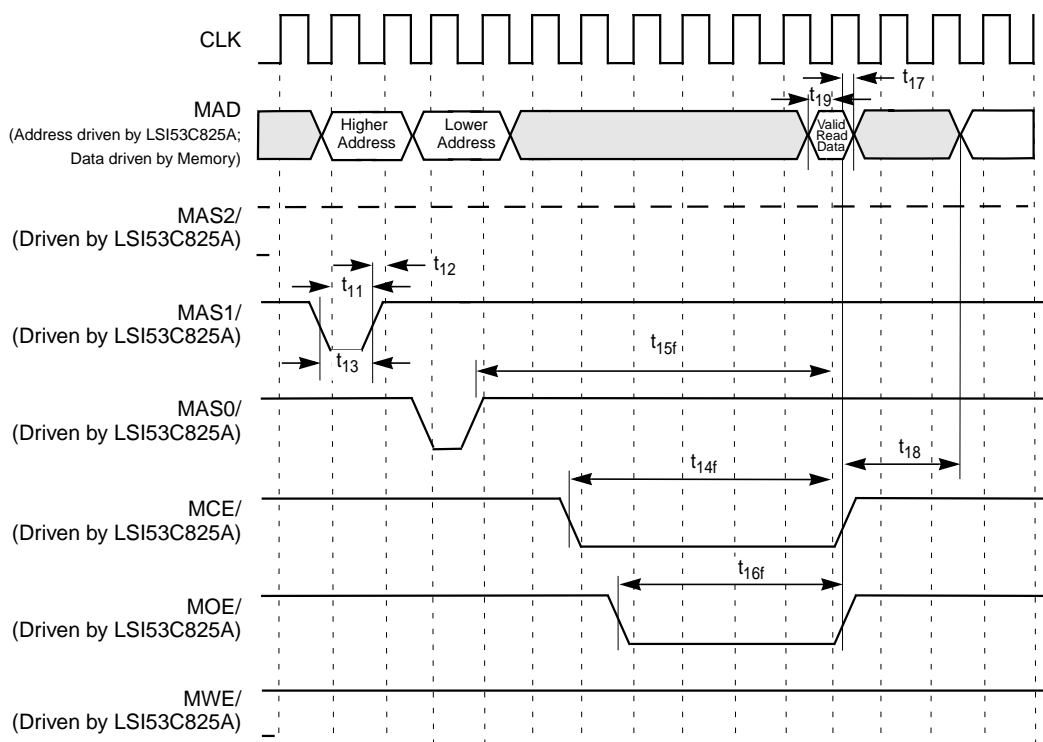
**Figure 6.25 Read Cycle, Slow Memory ( $\geq 64$  Kbytes)**



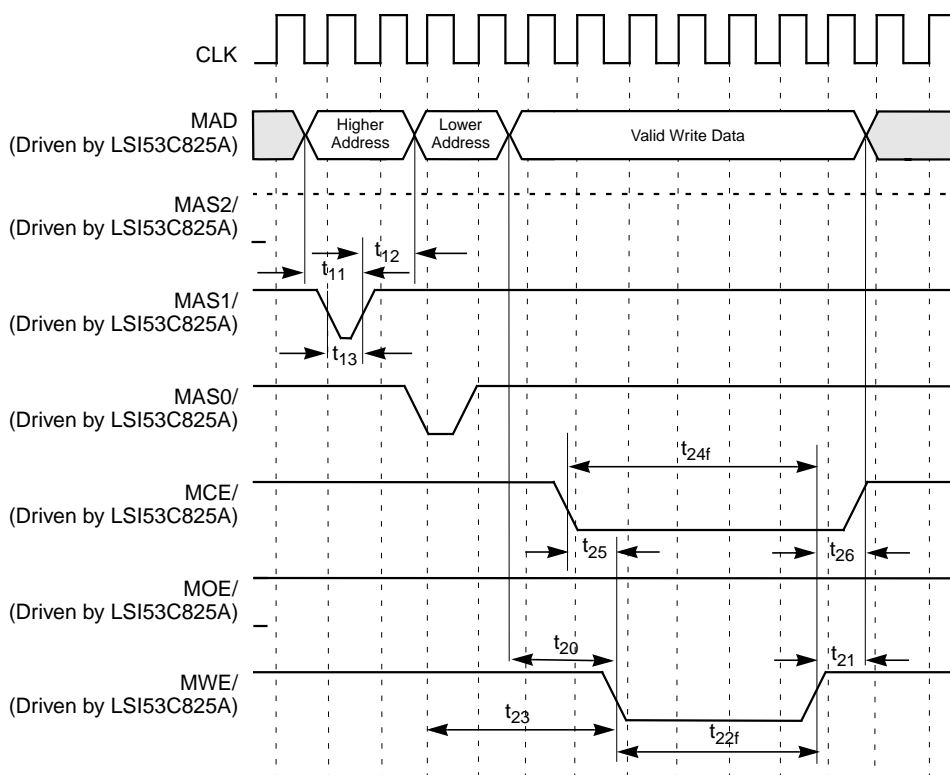
**Figure 6.26 Write Cycle, Slow Memory ( $\geq 64$  Kbytes)**



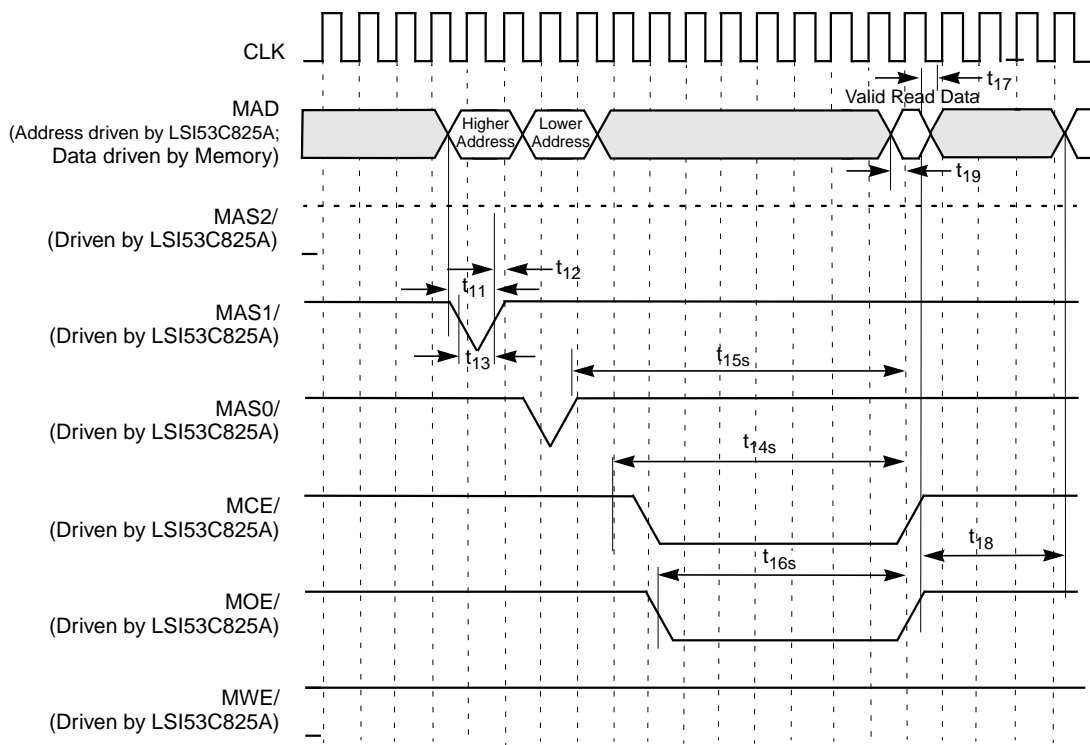
**Figure 6.27 Read Cycle, Normal/Fast Memory ( $\geq 64$  Kbytes)**



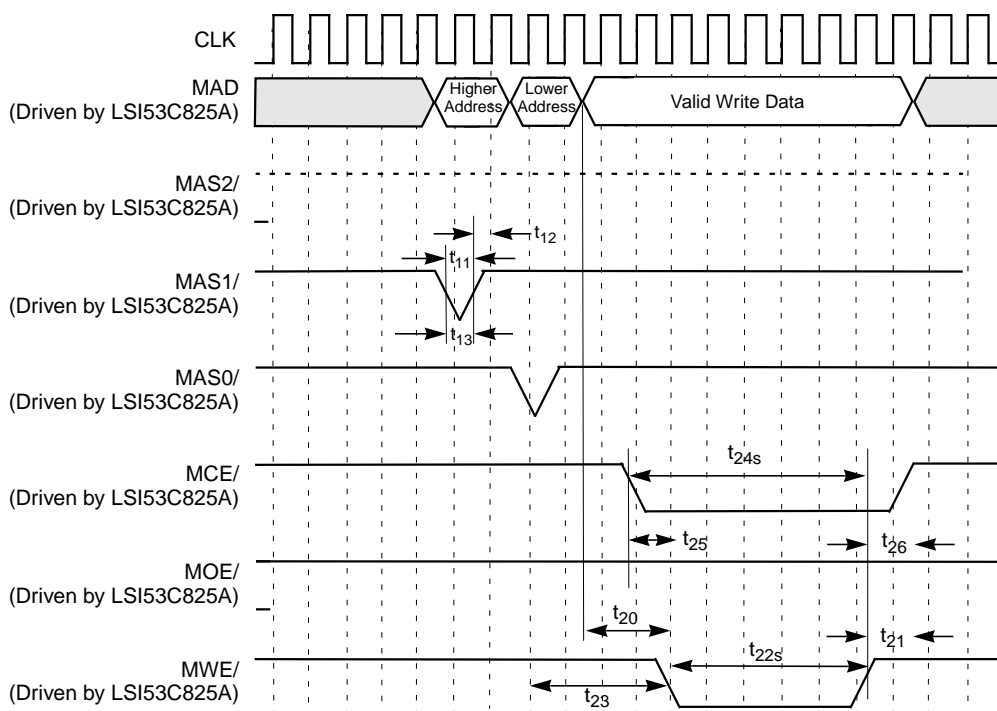
**Figure 6.28 Write Cycle, Normal/Fast Memory ( $\geq 64$  Kbytes)**



**Figure 6.29 Read Cycle, Slow Memory ( $\leq 64$  Kbytes)**



**Figure 6.30 Write Cycle, Slow Memory ( $\leq 64$  Kbytes)**



---

## 6.5 PCI and External Memory Interface Timing

Table 6.19 lists the LSI53C825A PCI and External Memory Interface timing data.

**Table 6.19 LSI53C825A PCI and External Memory Interface Timing**

Symbol	Parameter	Min	Max	Unit
t <sub>1</sub>	Shared signal input setup time	7	–	ns
t <sub>2</sub>	Shared signal input hold time	0	–	ns
t <sub>3</sub>	CLK to shared signal output valid	–	11	ns
t <sub>4</sub>	Side signal input setup time	10	–	ns
t <sub>5</sub>	Side signal input hold time	0	–	ns
t <sub>6</sub>	CLK to side signal output valid	–	12	ns
t <sub>7</sub>	CLK high to FETCH/ low	–	20	ns
t <sub>8</sub>	CLK high to FETCH/ high	–	20	ns
t <sub>9</sub>	CLK high to MASTER/ low	–	20	ns
t <sub>10</sub>	CLK high to MASTER/ high	–	20	ns
t <sub>11</sub>	Address setup to MAS/ high	25	–	ns
t <sub>12</sub>	Address hold from MAS/ high	15	–	ns
t <sub>13</sub>	MAS/ pulse width	25	–	ns
t <sub>14f</sub>	MCE/ low to data clocked in (fast memory)	160	–	ns
t <sub>14s</sub>	MCE/ low to data clocked in (slow memory)	220	–	ns
t <sub>15f</sub>	Address valid to data clocked in (fast memory)	205	–	ns
t <sub>15s</sub>	Address valid to data clocked in (slow memory)	265	–	ns
t <sub>16f</sub>	MOE/ low to data clocked in (fast memory)	100	–	ns
t <sub>16s</sub>	MOE/ low to data clocked in (slow memory)	160	–	ns
t <sub>17</sub>	Data hold from address, MOE/, MCE/ change	0	–	ns
t <sub>18</sub>	Next address out from MOE/, MCE/ high	50	–	ns
t <sub>19</sub>	Data setup to CLK high	5	–	ns
t <sub>20</sub>	Data setup to MWE/ low	30	–	ns
t <sub>21</sub>	Data hold from MWE/ high	20	–	ns
t <sub>25</sub>	MCE/ low to MWE/ low	25	–	ns
t <sub>26</sub>	MWE/ high to MCE/ high	25	–	ns



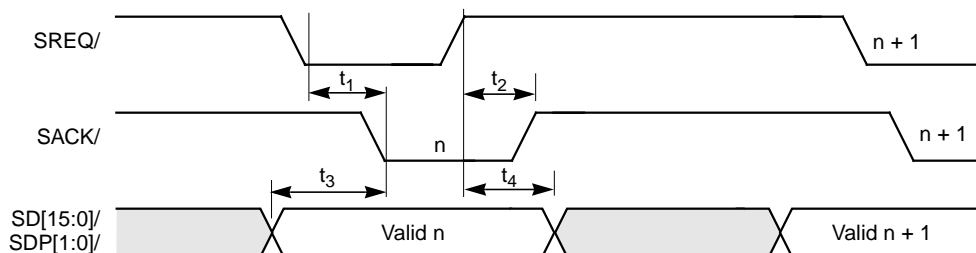
## 6.6 SCSI Timing Diagrams

Table 6.20 through Table 6.27 and Figure 6.31 through Figure 6.35 describe the LSI53C825A SCSI timing.

**Table 6.20 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	20	—	ns

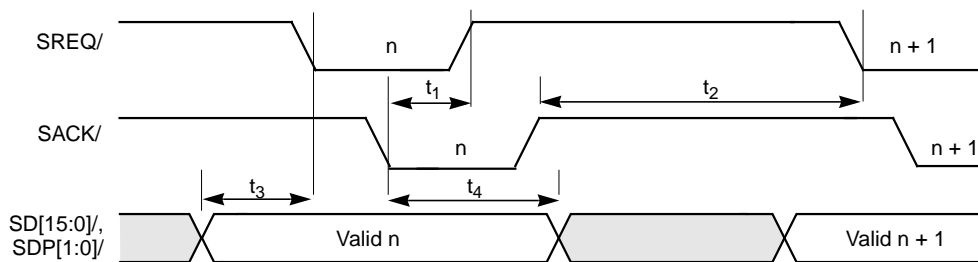
**Figure 6.31 Initiator Asynchronous Send**



**Table 6.21 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

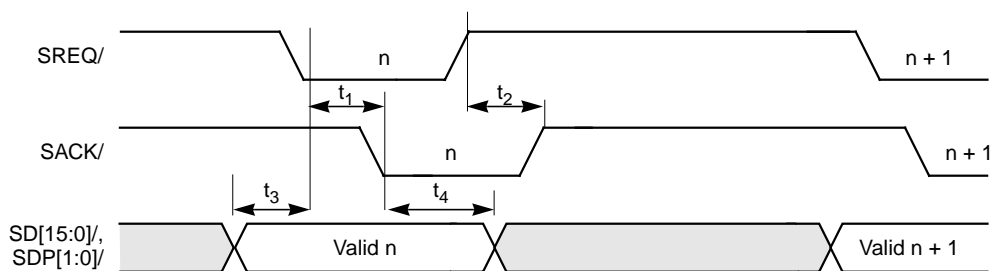
**Figure 6.32 Initiator Asynchronous Receive**



**Table 6.22 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	20	–	ns

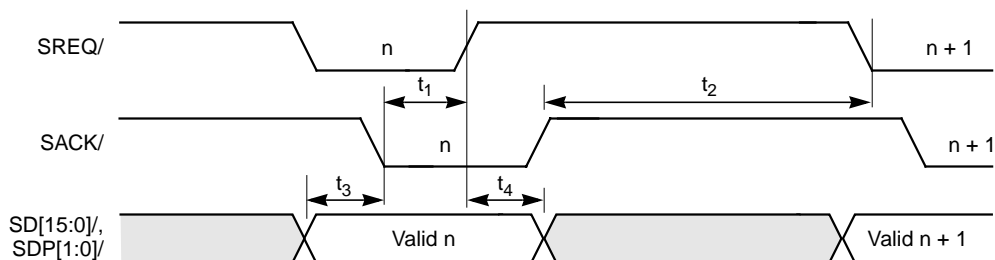
**Figure 6.33 Target Asynchronous Send**



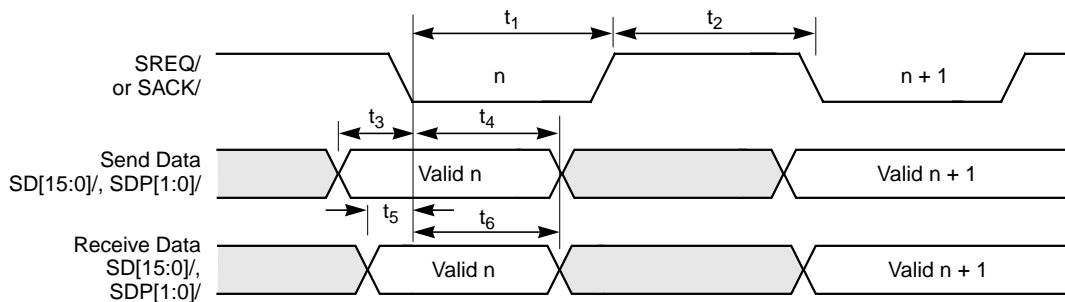
**Table 6.23 Target Asynchronous Receive**

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 6.34 Target Asynchronous Receive**



**Figure 6.35 Initiator and Target Synchronous Transfers**



**Table 6.24 SCSI-1 Transfers (SE 5.0 Mbytes)**

Symbol	Parameter	Min	Max	Unit
$t_1$	Send SREQ/ or SACK/ assertion pulse width	90	—	ns
$t_2$	Send SREQ/ or SACK/ deassertion pulse width	90	—	ns
$t_1$	Receive SREQ/ or SACK/ assertion pulse width	90	—	ns
$t_2$	Receive SREQ/ or SACK/ deassertion pulse width	90	—	ns
$t_3$	Send data setup to SREQ/ or SACK/ asserted	55	—	ns
$t_4$	Send data hold from SREQ/ or SACK/ asserted	100	—	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 6.25 SCSI-1 Transfers (Differential, 4.17 Mbytes/s)**

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 6.26 SCSI-2 Fast Transfers (10.0 Mbytes/s (8-Bit Transfers) or 20.0 Mbytes/s (16-Bit Transfers), 40 MHz Clock)**

Symbol	Parameter	Min	Max	Unit
t <sub>1</sub>	Send SREQ/ or SACK/ assertion pulse width	35	–	ns
t <sub>2</sub>	Send SREQ/ or SACK/ deassertion pulse width	35	–	ns
t <sub>1</sub>	Receive SREQ/ or SACK/ assertion pulse width	20	–	ns
t <sub>2</sub>	Receive SREQ/ or SACK/ deassertion pulse width	20	–	ns
t <sub>3</sub>	Send data setup to SREQ/ or SACK/ asserted	33	–	ns
t <sub>4</sub>	Send data hold from SREQ/ or SACK/ asserted	45	–	ns
t <sub>5</sub>	Receive data setup to SREQ/ or SACK/ asserted	0	–	ns
t <sub>6</sub>	Receive data hold from SREQ/ or SACK/ asserted	10	–	ns

**Table 6.27 SCSI-2 Fast Transfers 10.0 Mbytes (8-Bit Transfers) or 20.0 Mbytes/s (16-Bit Transfers) 50 MHz Clock**

Symbol	Parameter	Min	Max	Unit
t <sub>1</sub>	Send SREQ/ or SACK/ assertion pulse width	35	–	ns
t <sub>2</sub>	Send SREQ/ or SACK/ deassertion pulse width	35	–	ns
t <sub>1</sub>	Receive SREQ/ or SACK/ assertion pulse width	20	–	ns
t <sub>2</sub>	Receive SREQ/ or SACK/ deassertion pulse width	20	–	ns
t <sub>3</sub>	Send data setup to SREQ/ or SACK/ asserted	33	–	ns
t <sub>4</sub>	Send data hold from SREQ/ or SACK/ asserted	40 <sup>1</sup>	–	ns
t <sub>5</sub>	Receive data setup to SREQ/ or SACK/ asserted	0	–	ns
t <sub>6</sub>	Receive data hold from SREQ/ or SACK/ asserted	10	–	ns

1. Analysis of system configuration is recommended due to reduced driver skew margin in differential systems.

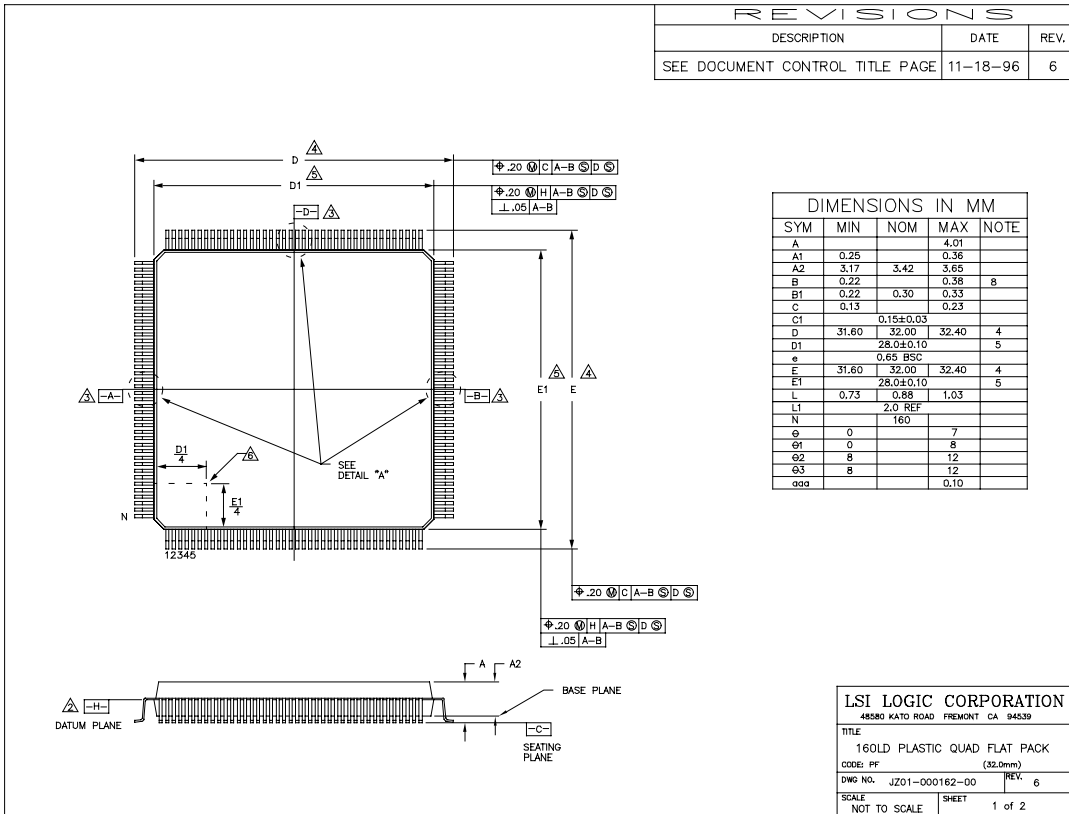
Notes: Transfer period bits (bits [7:5] in the [SCSI Transfer \(SXFER\)](#) register) are set to zero and the Extra Clock Cycle of Data Setup bit (bit 7 in [SCSI Control One \(SCNTL1\)](#)) is set. For Fast SCSI, set the TolerANT Enable bit (bit 7 in [SCSI Test Three \(STEST3\)](#)).

This page intentionally left blank.

# 6.7 Package Drawings

Figure 6.36 is the mechanical drawing for the LSI53C825A.

Figure 6.36 LSI53C825A 160 Pin PQFP (PF) 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 PF.



**DETAIL B**

SEATING PLANE  
A1  
C-10  
C-

WITH PLATING  
B  
C  
B1  
C1  
BASE METAL

△ $\phi .000$  C A-B S D △

DATUM PLANE -H-  
Ø1  
Ø2  
0.40 MIN.  
0.13 R MIN.  
0.13/0.25R  
L  
L1  
θ  
GAGE LINE  
0.25  
DETAIL "B"

NOTES:

- ALL DIMENSIONS AND TOLERANCES CONFORM TO ANSI Y14.5M-1982.
- DATUM PLANE -H- IS LOCATED AT THE MOLD PARTING LINE AND IS COINCIDENT WITH THE BOTTOM OF THE LEADS WHERE THE LEAD EXITS THE PLASTIC BODY.
- DATUMS A-B AND -D- TO BE DETERMINED AT DATUM PLANE -H-.
- TO BE DETERMINED AT SEATING PLANE -C-.
- DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS .25mm/.010" PER SIDE. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE -H-.
- DETAILS OF PIN 1 IDENTIFIER ARE OPTIONAL BUT MUST BE LOCATED WITHIN THE ZONE INDICATED.
- CONTROLLING DIMENSION: MILLIMETER.
- DIMENSION B DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE .08mm/.003" TOTAL IN EXCESS OF THE B DIMENSION AT MAXIMUM MATERIAL CONDITION. DAMBAR CANNOT BE LOCATED ON THE LOWER RADII OR THE FOOT. MINIMUM SPACING BETWEEN ADJACENT LEADS TO BE 0.10mm.
- MARKING AREA MUST BE FREE FROM PACKAGE SURFACE PROTRUSION, OR INTRUSION.
- PLATING THICKNESS INCLUDED; PLATING THICKNESS TO BE 0.005mm MINIMUM, 0.020mm MAXIMUM.

EVEN LEAD SIDES

ODD LEAD SIDES

DETAIL "A"

<b>LSI LOGIC CORPORATION</b>			
48580 KATO ROAD FREMONT CA 94539			
TITLE			
160LD PLASTIC QUAD FLAT PACK			
CODE: PF	(32.0mm)		
DWG NO. JZ01-000162-00	REV. 6		
SCALE	NOT TO SCALE	SHEET	2 of 2

6-53



# Appendix A

## Register Summary

Table A.1 lists the LSI53C825A configuration registers by register name.

**Table A.1 Configuration Registers**

Register Name	Address	Read/Write	Page
Base Address One (Memory)	0x14	Read/Write	4-9
Base Address Zero (I/O)	0x10	Read/Write	4-9
Bridge Support Extensions (PMCSR_BSE)	0x46	Read Only	4-17
Cache Line Size	0x0C	Read/Write	4-8
Capability ID	0x40	Read Only	4-14
Capability Pointer	0x34	Read Only	4-12
Class Code	0x09	Read Only	4-7
Command	0x04	Read/Write	4-3
Data	0x47	Read Only	4-17
Device ID	0x02	Read Only	4-3
Expansion ROM Base Address	0x30	Read/Write	4-11
Header Type	0x0E	Read Only	4-9
Interrupt Line	0x3C	Read/Write	4-13
Interrupt Pin	0x3D	Read Only	4-13
Latency Timer	0x0D	Read/Write	4-8
Max_Lat	0x3F	Read Only	4-14
Min_Gnt	0x3E	Read Only	4-13
Next Item Pointer	0x41	Read Only	4-15

**Table A.1 Configuration Registers (Cont.)**

Register Name	Address	Read/Write	Page
Power Management Capabilities	0x42	Read Only	4-15
Power Management Control/Status	0x44	Read Write	4-16
RAM Base Address Two (Memory) SCRIPTS RAM	0x18	Read/Write	4-10
Revision ID	0x08	Read Only	4-7
Status	0x06	Read/Write	4-5
Subsystem ID (SSID)	0x2E	Read Only	4-11
Subsystem Vendor ID (SSVID)	0x2C	Read Only	4-10
Vendor ID	0x00	Read Only	4-3

Table A.2 lists the LSI53C825A register map by register name.

**Table A.2 LSI53C825A Register Map**

Register Name	Address	Read/Write	Page
Adder Sum Output (ADDER)	0x3C–0x3F (0xBC–0xBF)	Read Only	4-71
Chip Test Five (CTEST5)	0x22 (0xA2)	Read/Write	4-59
Chip Test Four (CTEST4)	0x21 (0xA1)	Read/Write	4-57
Chip Test One (CTEST1)	0x19 (0x99)	Read Only	4-52
Chip Test Six (CTEST6)	0x23 (0xA3)	Read/Write	4-61
Chip Test Three (CTEST3)	0x1B (0x9B)	Read/Write	4-54
Chip Test Two (CTEST2)	0x1A (0x9A)	Read/Write	4-53
Chip Test Zero (CTEST0)	0x18 (0x98)	Read/Write	4-52
Data Structure Address (DSA)	0x10–0x13 (0x90–0x93)	Read/Write	4-48
DMA Byte Counter (DBC)	0x24–0x26 (0xA4–0xA6)	Read/Write	4-62
DMA Command (DCMD)	0x27 (0xA7)	Read/Write	4-63
DMA Control (DCNTL)	0x3B (0xBB)	Read/Write	4-69
DMA FIFO (DFIFO)	0x20 (0xA0)	Read/Write	4-56

**Table A.2 LSI53C825A Register Map (Cont.)**

Register Name	Address	Read/Write	Page
DMA Interrupt Enable (DIEN)	0x39 (0xB9)	Read/Write	4-68
DMA Mode (DMODE)	0x3B (0xBB)	Read/Write	4-65
DMA Next Address (DNAD)	0x28–0x2B (0xA8–0xAB)	Read/Write	4-63
DMA SCRIPTS Pointer (DSP)	0x2C–0x2F (0xAC–0xAF)	Read/Write	4-64
DMA SCRIPTS Pointer Save (DSPS)	0x30–0x33 (0xB0–0xB3)	Read/Write	4-64
DMA Status (DSTAT)	0x0C (0x8C)	Read Only	4-40
General Purpose (GPREG)	0x07 (0x87)	Read/Write	4-35
General Purpose Pin Control (GPCNTL)	0x47 (0xC7)	Read/Write	4-81
Interrupt Status (ISTAT)	0x14 (0x94)	Read/Write	4-48
Memory Access Control (MACNTL)	0x46 (0xC6)	Read/Write	4-80
Response ID One (RESPID1)	0x4B (0xCB)	Read/Write	4-86
Response ID Zero (RESPID0)	0x4A (0xCA)	Read/Write	4-86
Scratch Byte Register (SBR)	0x3A (0xBA)	Read/Write	4-69
Scratch Register A (SCRATCHA)	0x34–0x37 (0xB4–0xB7)	Read/Write	4-65
Scratch Register B (SCRATCHB)	0x5C–0x5F (0xDC–0xDF)	Read/Write	4-95
Scratch Registers C–J (SCRATCHC–SCRATCHJ)	0x60–0x7F (0xE0–0xFF)	Read/Write	4-95
SCSI Bus Control Lines (SBCL)	0x0B (0x8B)	Read Only	4-39
SCSI Bus Data Lines (SBDL)	0x58–0x59 (0xD8–0xD9)	Read Only	4-94
SCSI Chip ID (SCID)	0x04 (0x84)	Read/Write	4-30
SCSI Control One (SCNTL1)	0x01 (0x81)	Read/Write	4-23
SCSI Control Three (SCNTL3)	0x03 (0x83)	Read/Write	4-28
SCSI Control Two (SCNTL2)	0x02 (0x82)	Read/Write	4-26
SCSI Control Zero (SCNTL0)	0x00 (0x80)	Read/Write	4-20
SCSI Destination ID (SDID)	0x06 (0x86)	Read/Write	4-35
SCSI First Byte Received (SFBR)	0x08 (0x88)	Read/Write	4-37

**Table A.2 LSI53C825A Register Map (Cont.)**

Register Name	Address	Read/Write	Page
SCSI Input Data Latch (SIDL)	0x50–0x51 (0xD0–0xD1)	Read Only	4-93
SCSI Interrupt Enable One (SIEN1)	0x41 (0xC1)	Read/Write	4-73
SCSI Interrupt Enable Zero (SIEN0)	0x40 (0xC0)	Read/Write	4-71
SCSI Interrupt Status One (SIST1)	0x43 (0xC3)	Read Only	4-77
SCSI Interrupt Status Zero (SIST0)	0x42 (0xC2)	Read Only	4-74
SCSI Longitudinal Parity (SLPAR)	0x44 (0xC4)	Read/Write	4-78
SCSI Output Control Latch (SOCL)	0x09 (0x89)	Read/Write	4-38
SCSI Output Data Latch (SODL)	0x54–0x55 (0xD4–0xD5)	Read/Write	4-94
SCSI Selector ID (SSID)	0x0A (0x09)	Read Only	4-39
SCSI Status One (SSTAT1)	0x0E (0x8E)	Read Only	4-44
SCSI Status Two (SSTAT2)	0x0F (0x8F)	Read Only	4-46
SCSI Status Zero (SSTAT0)	0x0D (0x8D)	Read Only	4-43
SCSI Test One (STEST1)	0x4D (0xCD)	Read/Write	4-88
SCSI Test Three (STEST3)	0x4F (0xCF)	Read/Write	4-91
SCSI Test Two (STEST2)	0x4E (0xCE)	Read/Write	4-89
SCSI Test Zero (STEST0)	0x4C (0xCC)	Read Only	4-87
SCSI Timer One (STIME1)	0x49 (0xC9)	Read/Write	4-84
SCSI Timer Zero (STIME0)	0x48 (0xC8)	Read/Write	4-82
SCSI Transfer (SXFER)	0x05 (0x85)	Read/Write	4-31
SCSI Wide Residue (SWIDE)	0x45 (0xC5)	Read/Write	4-79
Temporary (TEMP)	0x1C–0x1F (0x9C–0x9F)	Read/Write	4-56

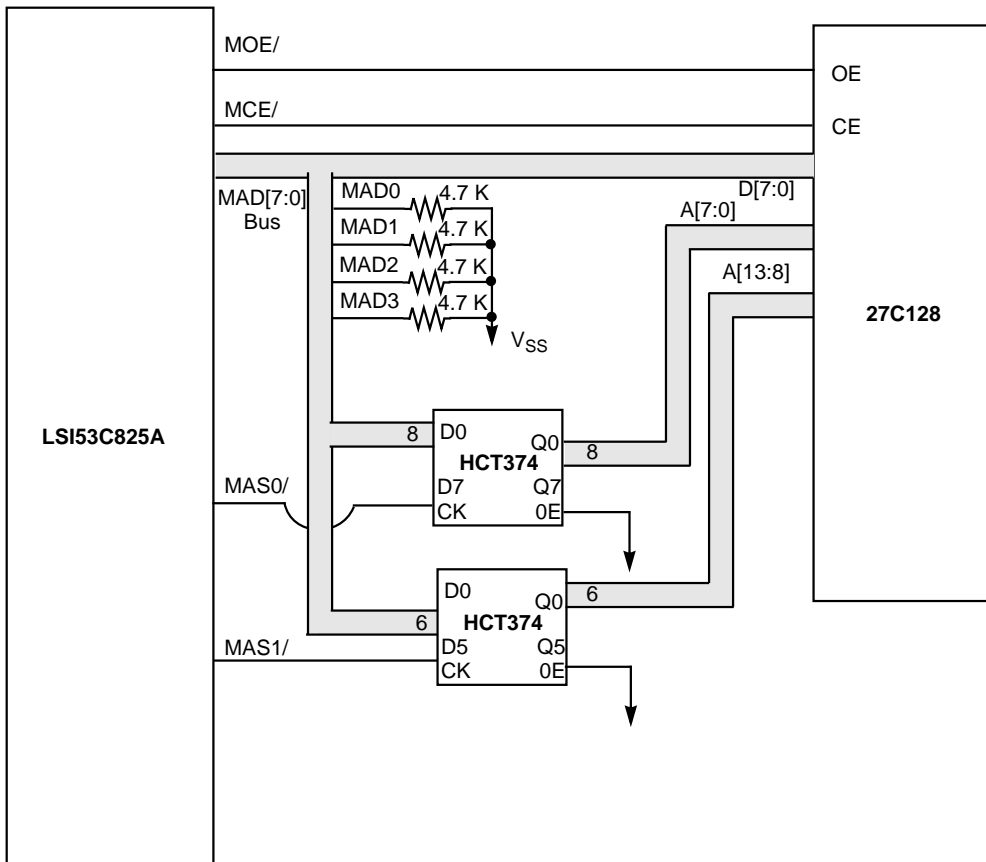
# Appendix B

## External Memory

### Interface Diagram

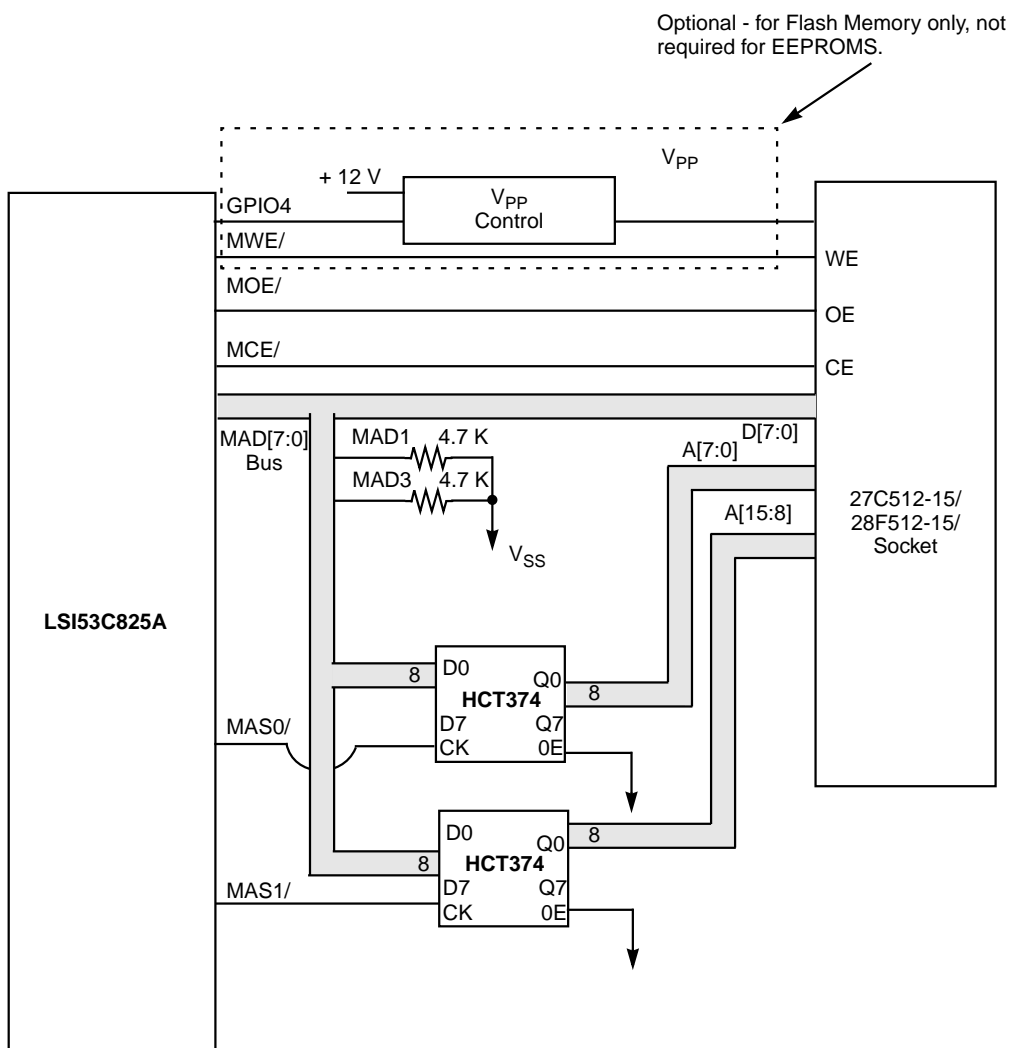
### Examples

Figure B.1 64 Kbyte Interface with 200 ns Memory



Note: MAD bus sense logic enabled for 16 Kbytes of slow memory (200 ns device @ 33 MHz).

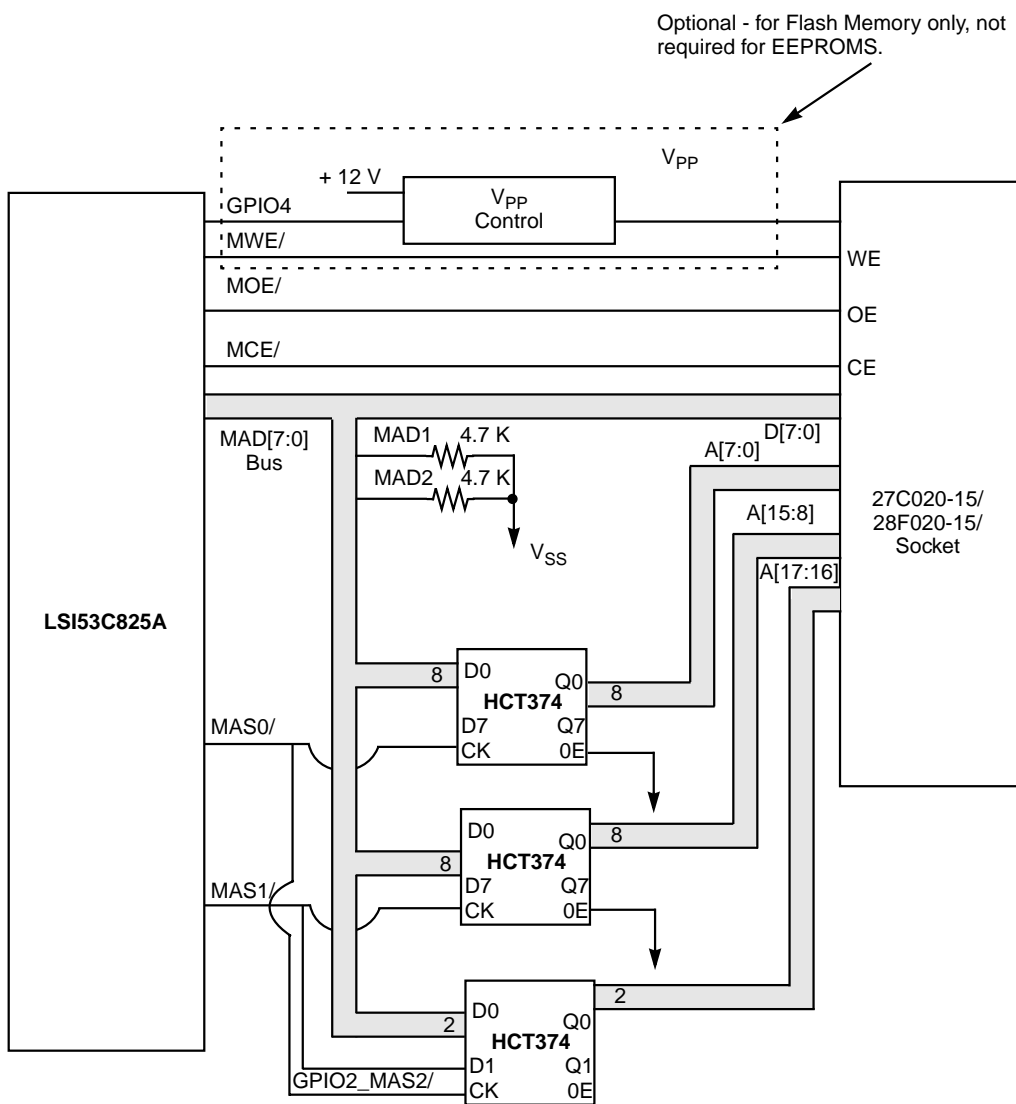
**Figure B.2 64 Kbyte Interface with 150 ns Memory**



Note: MAD bus sense logic enabled for 64 Kbytes of fast memory (150 ns device @ 33 MHz).

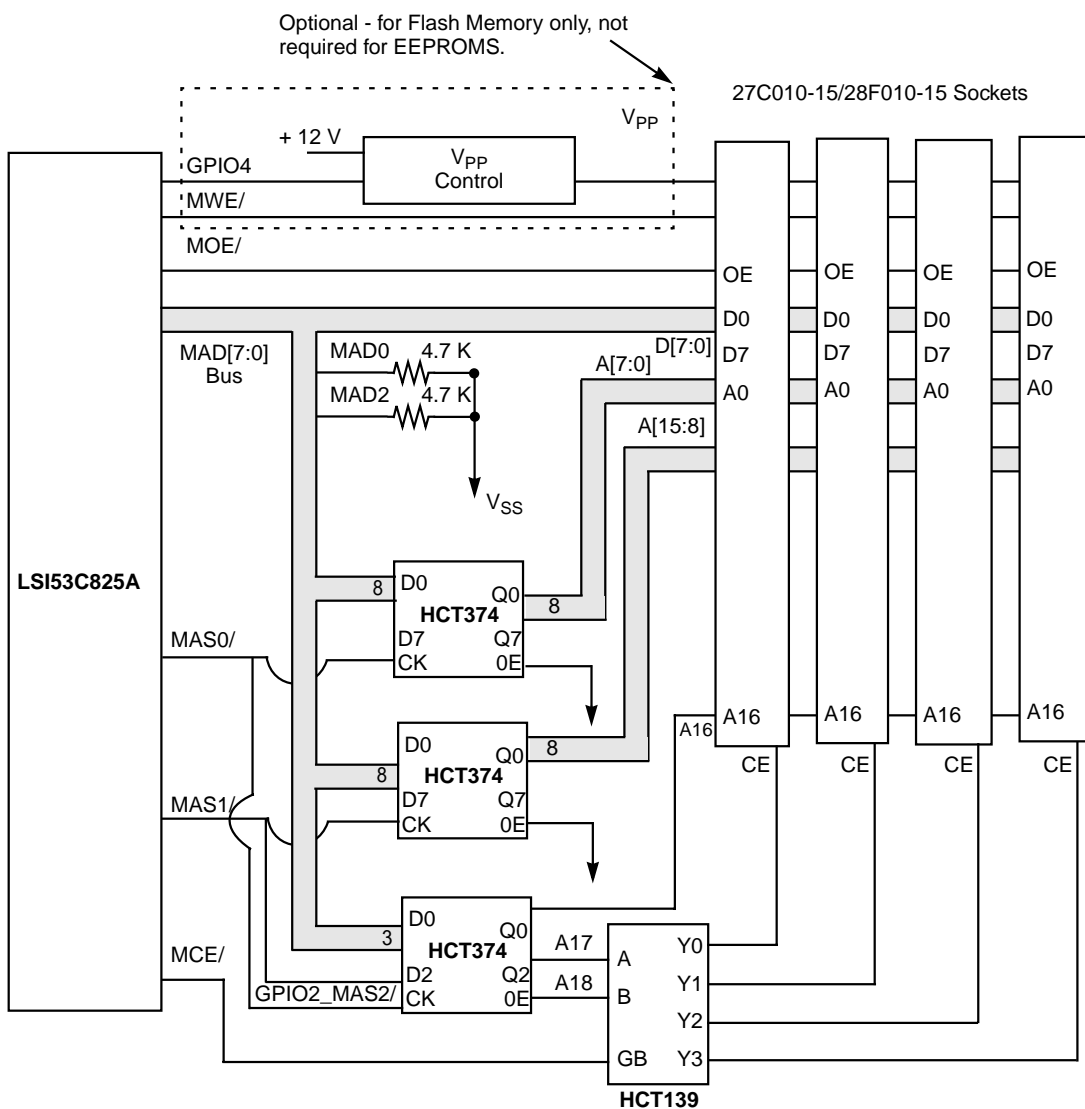


**Figure B.3 256 Kbyte Interface with 150 ns Memory**



Note: MAD bus sense logic enabled for 256 Kbytes of fast memory (150 ns device @ 33 MHz). Address latch for the two most significant bits can be HCT74, HCT174, HCT175, etc.

**Figure B.4 512 Kbyte Interface with 150 ns Memory**



Note: MAD bus sense logic enabled for 512 Kbytes of slow memory (150 ns device, additional time required for HCT139 @ 33 MHz).

# Index

---

## Symbols

(AAP) 4-22  
(ABRT) 4-41, 4-48  
(ACK) 4-38, 4-39  
(ADB) 4-23  
(ADCK) 4-59  
(ADDER) 4-70  
(AESP) 4-24  
(AIP) 4-44  
(ARB[1:0]) 4-20  
(ART) 4-86  
(ATN) 4-38, 4-39  
(AWS) 4-89  
(BARO[31:0]) 4-9, 4-10  
(BARZ[31:0]) 4-9  
(BBCK) 4-59  
(BDIS) 4-57  
(BF) 4-41, 4-67  
(BL[1:0]) 4-64  
(BL2) 4-60  
(BO[7:0]) 4-56  
(BO[9:8]) 4-60  
(BOF) 4-66  
(BSE[7:0]) 4-17  
(BSY) 4-38, 4-39  
(C\_D) 4-38, 4-40, 4-45  
(CCF[2:0]) 4-29  
(CHM) 4-26  
(CID[7:0]) 4-14  
(CIO) 4-53  
(CLF) 4-55  
(CLS[7:0]) 4-8  
(CLSE) 4-68  
(CM) 4-53  
(CMP) 4-71, 4-74  
(COM) 4-70  
(CON) 4-24, 4-50  
(CP[7:0]) 4-12  
(CSF) 4-91  
(CTEST0) 4-52  
(CTEST1) 4-52  
(CTEST2) 4-53  
(CTEST3) 4-54  
(CTEST4) 4-57  
(CTEST5) 4-59  
(CTEST6) 4-60  
(DACK) 4-54  
(DATA[7:0]) 4-17  
(DBC) 4-61  
(DCMD) 4-62  
(DCNTL) 4-68  
(DDIR) 4-53, 4-60  
(DF[7:0]) 4-60  
(DFE) 4-40  
(DFIFO) 4-56  
(DFS) 4-59  
(DHP) 4-24  
(DIEN) 4-67  
(DIF) 4-88  
(DIFF) 4-47  
(DIOM) 4-65  
(DIP) 4-51  
(DMODE) 4-64  
(DNAD) 4-62  
(DREQ) 4-54  
(DSA) 4-48  
(DSI) 4-91  
(DSP) 4-63  
(DSPS) 4-63  
(DSTAT) 4-40  
(ENC[3:0]) 4-30, 4-35  
(ENID[3:0]) 4-39  
(EPC) 4-22  
(ERBA[31:0]) 4-11  
(ERL) 4-66  
(ERMP) 4-66  
(EWS) 4-29  
(EXC) 4-23  
(EXT) 4-89  
(FBL[2:0]) 4-58  
(FE) 4-80  
(FF[3:0]) 4-44  
(FF4) 4-47  
(FFL[3:0]) 4-52  
(FLF) 4-54  
(FM) 4-55  
(FMT) 4-52  
(GEN) 4-73, 4-76  
(GEN[3:0]) 4-84  
(GENSF) 4-83  
(GPCNTL) 4-80  
(GPIO[1:0]) 4-80  
(GPIO[4:0]) 4-35  
(GPIO[4:2]) 4-80  
(GPREG) 4-35  
(HSC) 4-90  
(HT[7:0]) 4-9  
(HTH) 4-73, 4-76  
(HTH[3:0]) 4-81  
(HTHBA) 4-83  
(HTHSF) 4-84  
(I/O) 4-38, 4-40, 4-45

(IARB) 4-25  
 (IID) 4-41, 4-67  
 (IL[7:0]) 4-13  
 (ILF) 4-43  
 (ILF1) 4-46  
 (INTF) 4-50  
 (IP[7:0]) 4-13  
 (IRQD) 4-69  
 (IRQM) 4-69  
 (ISTAT) 4-48  
 (LDSC) 4-47  
 (LOA) 4-44  
 (LOW) 4-89  
 (LT[7:0]) 4-8  
 (M/A) 4-71, 4-74  
 (MACNTL) 4-79  
 (MAN) 4-66  
 (MASR) 4-59  
 (MDPE) 4-41, 4-67  
 (ME) 4-80  
 (MG[7:0]) 4-13  
 (ML[7:0]) 4-14  
 (MO[4:0]) 4-34  
 (MPEE) 4-58  
 (MSG) 4-38, 4-40, 4-45  
 (NIP[7:0]) 4-15  
 (OLF) 4-43  
 (OLF1) 4-46  
 (ORF) 4-43  
 (ORF1) 4-46  
 (PAR) 4-72, 4-75  
 (PFEN) 4-68  
 (PFF) 4-68  
 (R) 4-80  
 (REQ) 4-38, 4-39  
 (RESPID0) 4-85  
 (ROF) 4-88  
 (RRE) 4-30  
 (RSL) 4-71, 4-74  
 (RST) 4-24, 4-44, 4-72, 4-75  
 (S16) 4-91  
 (SBCL) 4-39  
 (SBDL) 4-93  
 (SBR) 4-68  
 (SCE) 4-88  
 (SCF[2:0]) 4-28  
 (SCID) 4-30  
 (SCLK) 4-87  
 (SCNTL0) 4-20  
 (SCNTL1) 4-23  
 (SCNTL2) 4-26  
 (SCNTL3) 4-28  
 (SCRATCHA) 4-64  
 (SCRATCHB) 4-94  
 (SCRATCHC–SCRATCHR) 4-94  
 (SDID) 4-35  
 (SDP0) 4-44  
 (SDP0L) 4-45  
 (SDP1) 4-47  
 (SDU) 4-26  
 (SEL) 4-38, 4-39, 4-71, 4-74  
 (SEM) 4-49  
 (SFBR) 4-37  
 (SGE) 4-71, 4-74  
 (SID[15:0]) 4-11  
 (SIEN0) 4-70

(SIEN1) 4-72  
 (SIGP) 4-49, 4-53  
 (SIP) 4-50  
 (SIR) 4-41, 4-67  
 (SISO) 4-87  
 (SIST0) 4-73  
 (SIST1) 4-76  
 (SLB) 4-88  
 (SLPAR) 4-77  
 (SLPHBEN) 4-27  
 (SLPMD) 4-27  
 (SLT) 4-86  
 (SOCL) 4-38  
 (SODL) 4-93  
 (SOM) 4-87  
 (SOZ) 4-86  
 (SPL1) 4-47  
 (SRE) 4-30  
 (SRST) 4-49  
 (SRTCH) 4-53  
 (SRTM) 4-58  
 (SSAID[3:0]) 4-86  
 (SSI) 4-41, 4-67  
 (SSID) 4-39  
 (SSM) 4-69  
 (SST) 4-25  
 (SSTAT0) 4-43  
 (SSTAT1) 4-44  
 (SSTAT2) 4-46  
 (START) 4-21  
 (STD) 4-69  
 (STEST0) 4-86  
 (STEST1) 4-87  
 (STEST2) 4-88  
 (STEST3) 4-90  
 (STIME0) 4-81  
 (STIME1) 4-83  
 (STO) 4-73, 4-76  
 (STR) 4-90  
 (STW) 4-91  
 (SWIDE) 4-78  
 (SXFER) 4-31  
 (SZM) 4-89  
 (TE) 4-90  
 (TEMP) 4-55  
 (TEOP) 4-54  
 (TP[2:0]) 4-31  
 (TRG) 4-23  
 (TTM) 4-91  
 (TYP[3:0]) 4-79  
 (UDC) 4-72, 4-75  
 (V[3:0]) 4-54  
 (VAL) 4-39  
 (VUE0) 4-27  
 (VUE1) 4-27  
 (WATN) 4-22  
 (WOA) 4-44  
 (WRIE) 4-55  
 (WSR) 4-28  
 (WSS) 4-27  
 (ZMOD) 4-57  
 (ZSD) 4-57

## Numerics

16-bit system (S16) 4-91  
 3-state 3-4

## A

- A[6:0] [5-26](#)
- abort operation (ABRT) [4-48](#)
- aborted (ABRT) [4-41](#), [4-67](#)
- absolute maximum stress ratings [6-2](#)
- AC characteristics [6-11](#)
- active termination [2-29](#)
- adder sum output (ADDER) [4-70](#)
- address and data signals [3-7](#)
- always wide SCSI (AWS) [4-89](#)
- arbitration
  - arbitration signals [3-9](#)
  - in progress (AIP) [4-44](#)
  - mode bits 1 and 0 (ARB[1:0]) [4-20](#)
  - priority encoder test (ART) [4-86](#)
- assert
  - even SCSI parity (force bad parity) (AESP) [4-24](#)
  - SATN/ on parity error (AAP) [4-22](#)
- SCSI
  - ACK/ signal (ACK) [4-38](#), [4-39](#)
  - ATN/ signal (ATN) [4-38](#), [4-39](#)
  - BSY/ signal (BSY) [4-38](#), [4-39](#)
  - C\_D/ signal (C\_D) [4-38](#), [4-40](#)
  - data bus (ADB) [4-23](#)
  - I\_O/ signal (I/O) [4-38](#), [4-40](#)
  - MSG/ signal (MSG) [4-38](#), [4-39](#)
  - REQ/ signal (REQ) [4-38](#), [4-39](#)
  - RST/ signal (RST) [4-24](#)
  - SEL/ signal (SEL) [4-38](#), [4-39](#)

## B

- base address register
  - one (BARO[31:0]) [4-9](#)
  - one (BART[31:0]) [4-10](#)
  - zero - I/O (BARZ[31:0]) [4-9](#)
- bidirectional [3-4](#)
- big and little endian support [2-19](#)
- block move instructions [5-6](#)
- bridge support extensions (BSE[7:0]) [4-17](#)
- burst
  - disable (BDIS) [4-57](#)
  - length (BL[1:0]) [4-64](#)
  - length bit 2 (BL2) [4-60](#)
  - opcode fetch enable (BOF) [4-66](#)
- bus
  - fault (BF) [4-41](#), [4-67](#)
- byte
  - count [5-41](#)
  - empty in DMA FIFO (FMT) [4-52](#)
  - full in DMA FIFO (FFL[3:0]) [4-52](#)
  - offset counter (BO[7:0]) [4-56](#)

## C

- cache line size
  - (CLS[7:0]) [4-8](#)
  - enable (CLSE) [4-68](#)
- call instruction [5-30](#)
- cap\_ID (CID[7:0]) [4-14](#)
- capabilities pointer (CPI[7:0]) [4-12](#)
- carry test [5-34](#)
- chained block moves [2-42](#)
  - SODL register [2-43](#)

- SWIDE register [2-43](#)
- wide SCSI receive bit [2-42](#)
- wide SCSI send bit [2-42](#)
- chained mode (CHM) [4-26](#)
- chip
  - revision level (V[3:0]) [4-54](#)
  - test five (CTEST5) [4-59](#)
  - test four (CTEST4) [4-57](#)
  - test one (CTEST1) [4-52](#)
  - test six (CTEST6) [4-60](#)
  - test three (CTEST3) [4-54](#)
  - test two (CTEST2) [4-53](#)
  - test zero (CTEST0) [4-52](#)
  - type (TYP[3:0]) [4-79](#)
- clear DMA FIFO (CLF) [4-55](#)
- clear instruction [5-16](#), [5-19](#)
- clear SCSI FIFO (CSF) [4-91](#)
- clock
  - address incrementor (ADCK) [4-59](#)
  - byte counter (BBCK) [4-59](#)
  - conversion factor (CCF[2:0]) [4-29](#)
- compare
  - data [5-35](#)
  - phase [5-35](#)
- configured
  - as I/O (CIO) [4-53](#)
  - as memory (CM) [4-53](#)
- connected (CON) [4-24](#), [4-50](#)

## D

- data
  - (DATA[7:0]) [4-17](#)
  - acknowledge status (DACK) [4-54](#)
  - compare mask [5-36](#)
  - compare value [5-36](#)
  - request status (DREQ) [4-54](#)
  - structure address (DSA) [4-48](#)
  - transfer direction (DDIR) [4-53](#)
- data path [2-24](#)
- destination
  - address [5-26](#)
  - I/O-memory enable (DIOM) [4-65](#)
- differential mode
  - DIFFSENS [3-12](#)
  - direction control pins [3-10](#)
  - operation [2-28](#)
- diffsens mismatch (DIFF) [4-47](#)
- DIFFSENS SCSI signal [6-3](#)
- direct [5-21](#)
- disable
  - halt on parity error or ATN (target only) (DHP) [4-24](#)
  - single initiator response (DSI) [4-91](#)
- disconnect instruction [5-15](#)
- DMA
  - byte counter (DBC) [4-61](#)
  - command (DCMD) [4-62](#)
  - control (DCNTL) [4-68](#)
  - direction (DDIR) [4-60](#)
  - FIFO
    - (DF[7:0]) [4-60](#)
    - (DFIFO) [4-56](#)
    - byte offset counter, bits [9:8] (BO[9:8]) [4-60](#)
    - empty (DFE) [4-40](#)
    - size (DFS) [4-59](#)
  - interrupt
    - enable (DIEN) [4-67](#)

- DMA (Cont.)
  - interrupt pending (DIP) [4-51](#)
  - mode (DMODE) [4-64](#)
  - next address (DNAD) [4-62](#)
  - SCRIPTS
    - pointer (DSP) [4-63](#)
    - pointer save (DSPS) [4-63](#)
    - status (DSTAT) [4-40](#)
- DMA core [2-11](#)
- DSA
  - relative [5-41](#)
- DSPS register [5-38](#)

## E

- enable
  - parity
    - checking (EPC) [4-22](#)
  - read
    - line (ERL) [4-66](#)
    - multiple (ERMP) [4-66](#)
  - response to
    - reselection (RRE) [4-30](#)
    - selection (SRE) [4-30](#)
  - wide SCSI (EWS) [4-29](#)
- encoded
  - chip SCSI ID (ENC[3:0]) [4-30](#)
  - destination SCSI ID
    - (ENC[3:0]) [4-35](#)
    - (ENID[3:0]) [4-39](#)
  - SCSI destination ID [5-22](#)
- error reporting signals [3-9](#)
- expansion ROM base address [4-11](#)
- extend SREQ/SACK filtering (EXT) [4-89](#)
- external
  - clock [6-11](#)
- external memory interface [2-14](#)
  - configuration [2-16](#)
  - flash ROM updates [2-14](#)
  - memory sizes supported [2-16](#)
  - multiple byte accesses [6-13](#)
  - slow memory [2-16](#)
  - system requirements [2-14](#)
- extra clock cycle of data setup (EXC) [4-23](#)

## F

- fetch
  - enable (FE) [4-80](#)
  - pin mode (FM) [4-55](#)
- FIFO
  - byte control (FBL[2:0]) [4-58](#)
  - flags (FF[3:0]) [4-44](#)
  - flags, bit 4 (FF4) [4-47](#)
- first dword [5-6](#), [5-14](#), [5-24](#), [5-29](#), [5-41](#)
- flush DMA FIFO (FLF) [4-54](#)
- full arbitration, selection/reselection [4-21](#)
- function complete
  - (CMP) [4-71](#), [4-74](#)

## G

- general purpose
  - (GPREG) [4-35](#)
  - I/O (GPIO[4:0]) [4-35](#)

- pin control (GPCNTL) [4-80](#)
- timer expired (GEN) [4-73](#), [4-76](#)
- timer period (GEN[3:0]) [4-84](#)
- timer scale factor (GENSF) [4-83](#)
- GPIO enable (GPIO[1:0]) [4-80](#)
- GPIO enable (GPIO[4:2]) [4-80](#)

## H

- halt SCSI clock (HSC) [4-90](#)
- handshake-to-handshake
  - timer bus activity enable (HTHBA) [4-83](#)
  - timer expired (HTH) [4-73](#), [4-76](#)
  - timer period (HTH[3:0]) [4-81](#)
  - timer scale factor (HTHSF) [4-84](#)
- header type (HT[7:0]) [4-9](#)
- high impedance mode (SZM) [4-89](#)
- high impedance mode (ZMOD) [4-57](#)
- HVD or SE/LVD (DIF) [4-88](#)

## I

- I/O
  - instructions [5-14](#)
- illegal instruction detected (IID) [4-41](#), [4-67](#)
- immediate
  - arbitration (IARB) [4-25](#)
  - data [5-26](#)
- indirect addressing [5-6](#)
- initiator
  - mode [5-18](#)
  - phase mismatch [4-74](#)
- input [3-4](#)
- instruction
  - type [5-41](#)
    - block move [5-6](#)
    - I/O instruction [5-14](#)
    - memory move [5-37](#)
    - read/write instruction [5-24](#)
    - transfer control instruction [5-29](#)
- instructions
  - block move [5-6](#)
- INTA, INTB disable (IRQD) [4-69](#)
- interface control signals [3-8](#)
- interrupt
  - instruction [5-32](#)
  - line [4-13](#)
  - on the fly [5-35](#)
  - on-the-fly (INTF) [4-50](#)
  - output [6-13](#)
  - pin (IP[7:0]) [4-13](#)
  - status (ISTAT) [4-48](#)
- interrupt-on-the-fly instruction [5-33](#)
- interrupts [2-35](#)
  - fatal vs. nonfatal interrupts [2-37](#)
  - halting [2-40](#)
  - masking [2-38](#)
  - sample interrupt service routine [2-41](#)
  - stacked interrupts [2-39](#)
- IRQ mode (IRQM) [4-69](#)

## J

- jump
  - address [5-36](#)

jump (Cont.)  
  call a relative address 5-34  
  call an absolute address 5-34  
  if true/false 5-35  
  instruction 5-29

## L

last disconnect (LDSC) 4-47  
latched SCSI parity  
  (SDP0L) 4-45  
  for SD[15:8] (SPL1) 4-47  
latency  
  timer (LT[7:0]) 4-8  
load/store 5-41  
lost arbitration (LOA) 4-44  
LSI53C700 family compatibility (COM) 4-70

## M

MAD bus programming  
  MAD[3:1] 3-17  
manual start mode (MAN) 4-66  
master  
  control for set or reset pulses (MASR) 4-59  
  data parity error (MDPE) 4-41, 4-67  
  enable (ME) 4-80  
  parity error enable (MPEE) 4-58  
max SCSI synchronous offset (MO[4:0]) 4-34  
max\_lat (ML[7:0]) 4-14  
maximum stress ratings 6-2  
memory  
  I/O address/DSA offset 5-42  
  move instructions 5-36  
memory access control (MACNTL) 4-79  
min\_gnt (MG[7:0]) 4-13  
move to/from SFBR cycles 5-27

## N

next\_item\_ptr (NIP[7:0]) 4-15  
no flush 5-37  
  store instruction only 5-41

## O

opcode 5-9, 5-14, 5-26, 5-29  
operating conditions 6-2  
operating registers  
  general information 4-18  
operator 5-26

## P

parity  
  error  
    (PAR) 4-75  
PCI  
  and external memory interface timing diagrams 6-13  
PCI commands 2-2  
PCI configuration registers 4-1  
  base address one (memory) 4-9  
  base address zero (I/O) 4-9  
  class code 4-7  
  command 4-3

  device ID 4-3  
  expansion ROM base address 4-11  
  header type 4-9  
  interrupt line 4-13  
  interrupt pin 4-13  
  latency timer 4-8  
  max\_lat 4-14  
  min\_gnt 4-13  
  revision ID 4-7  
  status 4-5  
  vendor ID 4-3  
PCI configuration space 2-1  
PCI I/O space 2-2  
PCI memory space 2-2  
prefetch  
  enable (PFEN) 4-68  
  flush (PFF) 4-68

## R

read  
  modify-write cycles 5-26  
  write instructions 5-24  
  write system memory from SCRIPTS 5-38  
read/write  
  instructions 5-24, 5-27  
  system memory from SCRIPTS 5-38  
register  
  address 5-41  
  address - A[6:0] 5-26  
register addresses  
  PCI configuration registers  
    0x02 4-3  
    0x04 4-3  
    0x06 4-5  
    0x08 4-7  
    0x09 4-7  
    0x0D 4-8  
    0x0E 4-9  
    0x10 4-9  
    0x3D 4-13  
    0x3E 4-13  
    0x3F 4-14  
relative 5-21  
relative addressing mode 5-19, 5-33  
reselect  
  instruction 5-15  
reselected (RSL) 4-71, 4-74  
Reserved 4-80  
reset  
  input 6-12  
  SCSI offset (ROF) 4-88  
response ID zero (RESPID0) 4-85  
return instruction 5-32

## S

scratch  
  byte register (SBR) 4-68  
  register A (SCRATCHA) 4-64  
  register B (SCRATCHB) 4-94  
  registers C–R (SCRATCHC–SCRATCHR) 4-94  
scratcha/b operation (SRTCH) 4-53  
SCRIPTS  
  interrupt instruction received (SIR) 4-41, 4-67  
SCRIPTS processor 2-11  
  performance 2-11

## SCSI

- ATN condition - target mode (M/A) 4-71
  - bus control lines (SBCL) 4-39
  - bus data lines (SBDL) 4-93
  - C\_D/ signal (C\_D) 4-45
  - chip ID (SCID) 4-30
  - clock (SCLK) 4-87
  - control enable (SCE) 4-88
  - control one (SCNTL1) 4-23
  - control three (SCNTL3) 4-28
  - control two (SCNTL2) 4-26
  - control zero (SCNTL0) 4-20
  - core 2-10
  - data high impedance (ZSD) 4-57
  - destination ID (SDID) 4-35
  - differential mode 2-28
  - disconnect unexpected (SDU) 4-26
  - encoded destination ID 5-22
  - FIFO test read (STR) 4-90
  - FIFO test write (STW) 4-91
  - first byte received (SFBR) 4-37
  - gross error (SGE) 4-71, 4-74
  - I\_O/ signal (I/O) 4-45
  - input data latch (SIDL) 4-92
  - instructions
    - I/O 5-14
    - read/write 5-24
  - interrupt enable one (SIEN1) 4-72
  - interrupt enable zero (SIEN0) 4-70
  - interrupt pending (SIP) 4-50
  - interrupt status one (SIST1) 4-76
  - interrupt status zero (SIST0) 4-73
  - isolation mode (SISO) 4-87
  - longitudinal parity (SLPAR) 4-77
  - loopback mode (SLB) 4-88
  - low level mode (LOW) 4-89
  - MSG/ signal (MSG) 4-45
  - output control latch (SOCL) 4-38
  - output data latch (SODL) 4-93
  - parity error (PAR) 4-72
  - phase 5-12, 5-33
  - phase mismatch - initiator mode 4-71
  - reset condition (RST) 4-72
  - RST/ received (RST) 4-75
  - RST/ signal (RST) 4-44
  - SDP0/ parity signal (SDP0) 4-44
  - SDP1/ parity signal (SDP1) 4-47
  - selected as ID (SSAID[3:0]) 4-86
  - selector ID (SSID) 4-39
  - signals 3-10
  - status one (SSTAT1) 4-44
  - status two (SSTAT2) 4-46
  - status zero (SSTAT0) 4-43
  - synchronous offset maximum (SOM) 4-87
  - synchronous offset zero (SOZ) 4-86
  - synchronous transfer period (TP[2:0]) 4-31
  - termination 2-29
  - test one (STEST1) 4-87
  - test three (STEST3) 4-90
  - test two (STEST2) 4-88
  - test zero (STEST0) 4-86
  - timer one (STIME1) 4-83
  - timer zero (STIME0) 4-81
  - TolerANT technology 1-3
  - transfer (SXFER) 4-31
  - true end of process 4-54
  - valid (VAL) 4-39
  - wide residue (SWIDE) 4-78
- SCSI bus interface 2-27 to 2-33
  - SCSI core 2-10
  - SCSI instructions
    - block move 5-6
  - SCSI SCRIPTS operation 5-2
    - sample instruction 5-3
  - SCSI-1 transfers (differential 4.17 Mbytes) 6-49
  - SCSI-1 transfers (single-ended 5.0 Mbytes) 6-49
  - SCSI-2 fast transfers 10.0 Mbytes (8-bit transfers) or 20.0 Mbytes (16-bit transfers) 40 MHz clock 6-50
  - SCSI-2 fast transfers 10.0 Mbytes (8-bit transfers) or 20.0 Mbytes (16-bit transfers) 50 MHz clock 6-50
  - second dword 5-13, 5-23, 5-26, 5-36, 5-38, 5-42
  - select
    - instruction 5-18
    - with ATN/ 5-22
    - with SATN/ on a start sequence (WATN) 4-22
  - selected (SEL) 4-71, 4-74
  - selection or reselection time-out (STO) 4-73, 4-76
  - selection response logic test (SLT) 4-86
  - semaphore (SEM) 4-49
  - set instruction 5-16, 5-19
  - set/clear
    - carry 5-22
    - SACK/ 5-23
  - shadow register test mode (SRTM) 4-58
  - SIDL
    - least significant byte full (ILF) 4-43
    - most significant byte full (ILF1) 4-46
  - signal process (SIGP) 4-49, 4-53
  - signals
    - address and data signals 3-7
    - arbitration signals 3-9
    - error reporting signals 3-9
    - interface control signals 3-8
    - SCSI signals 3-10
  - simple arbitration 4-20
  - single
    - step interrupt (SSI) 4-41, 4-67
    - step mode (SSM) 4-69
  - single-ended operation 2-27
  - SLPAR high byte enable (SLPHBEN) 4-27
  - SLPAR mode (SLPMD) 4-27
  - SODL
    - least significant byte full (OLF) 4-43
    - most significant byte full (OLF1) 4-46
  - SODR
    - least significant byte full (ORF) 4-43
    - most significant byte full (ORF1) 4-46
  - software reset (SRST) 4-49
  - source
    - I/O-memory enable (SIOM) 4-65
  - stacked interrupts 2-39
  - start
    - address 5-13, 5-23
    - DMA operation (STD) 4-69
    - SCSI transfer (SST) 4-25
    - sequence (START) 4-21
  - Storage Device Management System (SDMS) 2-12
  - stress ratings 6-2
  - subsystem ID
    - (SID[15:0]) 4-11
  - subsystem vendor ID
    - (SVID[15:0]) 4-10



- synchronous
  - clock conversion factor (SCF[2:0]) [4-28](#)
- synchronous data transfer rates [2-33](#)

## T

- table indirect [5-21](#)
  - mode [5-20](#)
- table relative [5-21](#)
- target
  - mode [5-9](#), [5-15](#)
    - SATN/ active (M/A) [4-74](#)
  - mode (TRG) [4-23](#)
  - timing [6-15](#)
- target asynchronous receive [6-48](#)
- target asynchronous send [6-47](#)
- TEMP register [5-38](#)
- temporary (TEMP) [4-55](#)
- termination [2-29](#)
- third dword [5-38](#)
- timer test mode (TTM) [4-91](#)
- TolerANT [6-7](#)
  - enable (TE) [4-90](#)
  - technology
    - electrical characteristics [6-7](#)
- TolerANT technology [1-3](#)
  - benefits [1-4](#)
- totem pole output [3-4](#)
- transfer
  - control instructions [5-29](#)
  - count [5-38](#)
  - counter [5-12](#)
- transfer rate
  - synchronous [2-33](#)

## U

- unexpected disconnect (UDC) [4-72](#), [4-75](#)
- use data8/SFBR [5-26](#)

## V

- vendor
  - unique enhancement, bit 1 (VUE1) [4-27](#)
  - unique enhancements, bit 0 (VUE0) [4-27](#)

## W

- wait
  - disconnect instruction [5-18](#)
  - for valid phase [5-36](#)
  - reselect instruction [5-19](#)
  - select instruction [5-16](#)
- wide SCSI
  - chained block moves [2-42](#)
  - receive (WSR) [4-28](#)
  - send (WSS) [4-27](#)
- won arbitration (WOA) [4-44](#)
- write
  - read instructions [5-24](#)
  - read system memory from SCRIPTS [5-38](#)
- write and invalidate
  - enable (WRIE) [4-55](#)



# 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: *LSI53C825A/825AE PCI to SCSI I/O Processor 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  
I. E. Tel: 256.830.1222  
W. E. Tel: 800.964.9953

### Alaska

A. E. Tel: 800.332.8638

### Arkansas

W. E. Tel: 972.235.9953

### 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

### California

Agoura Hills  
B. M. Tel: 818.865.0266  
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

### 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  
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: 800.231.5575

### 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: 508.480.9099  
Woburn  
B. M. Tel: 781.933.9010

### Michigan

Brighton  
I. E. Tel: 810.229.7710  
Detroit  
A. E. Tel: 734.416.5800  
W. E. Tel: 888.318.9953

### 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: 609.222.9566  
Pine Brook  
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: 503.524.0787  
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  
Pittsburgh  
A. E. Tel: 412.281.4150  
W. E. Tel: 440.248.9996  
Philadelphia  
A. E. Tel: 800.526.4812  
B. M. Tel: 215.741.4080  
W. E. Tel: 800.871.9953

#### 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

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

#### Washington

Kirkland  
I. E. Tel: 425.820.8100  
Seattle  
A. E. Tel: 425.882.7000  
W. E. Tel: 800.248.9953

#### West Virginia

A. E. Tel: 800.638.5988

#### Wisconsin

Milwaukee  
A. E. Tel: 414.513.1500  
W. E. Tel: 800.867.9953  
Wauwatosa  
I. E. Tel: 414.258.5338

#### Wyoming

A. E. Tel: 800.332.9326  
W. E. Tel: 801.974.9953

## Direct Sales Representatives by State (Component and 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**  
**Tel: 408.433.8000**  
**Fax: 408.433.8989**

### **NORTH AMERICA**

**California**  
Costa Mesa - Mint Technology  
Tel: 949.752.6468  
Fax: 949.752.6868

Irvine  
◆ Tel: 949.809.4600  
Fax: 949.809.4444

Pleasanton Design Center  
Tel: 925.730.8800  
Fax: 925.730.8700

San Diego  
Tel: 858.467.6981  
Fax: 858.496.0548

Silicon Valley  
◆ Tel: 408.433.8000  
Fax: 408.954.3353

Wireless Design Center  
Tel: 858.350.5560  
Fax: 858.350.0171

**Colorado**  
Boulder  
◆ Tel: 303.447.3800  
Fax: 303.541.0641

Colorado Springs  
Tel: 719.533.7000  
Fax: 719.533.7020

Fort Collins  
Tel: 970.223.5100  
Fax: 970.206.5549

**Florida**  
Boca Raton  
Tel: 561.989.3236  
Fax: 561.989.3237

**Georgia**  
Alpharetta  
Tel: 770.753.6146  
Fax: 770.753.6147

**Illinois**  
Oakbrook Terrace  
Tel: 630.954.2234  
Fax: 630.954.2235

**Kentucky**  
Bowling Green  
Tel: 270.793.0010  
Fax: 270.793.0040

**Maryland**  
Bethesda  
Tel: 301.897.5800  
Fax: 301.897.8389

**Massachusetts**  
Waltham  
◆ Tel: 781.890.0180  
Fax: 781.890.6158

Burlington - Mint Technology  
Tel: 781.685.3800  
Fax: 781.685.3801

**Minnesota**  
Minneapolis  
◆ Tel: 612.921.8300  
Fax: 612.921.8399

**New Jersey**  
Red Bank  
Tel: 732.933.2656  
Fax: 732.933.2643

Cherry Hill - Mint Technology  
Tel: 609.489.5530  
Fax: 609.489.5531

**New York**  
Fairport  
Tel: 716.218.0020  
Fax: 716.218.9010

**North Carolina**  
Raleigh  
Tel: 919.785.4520  
Fax: 919.783.8909

**Oregon**  
Beaverton  
Tel: 503.645.0589  
Fax: 503.645.6612

**Texas**  
Austin  
Tel: 512.388.7294  
Fax: 512.388.4171

Plano  
◆ Tel: 972.244.5000  
Fax: 972.244.5001

Houston  
Tel: 281.379.7800  
Fax: 281.379.7818

**Canada**  
**Ontario**  
Ottawa  
◆ Tel: 613.592.1263  
Fax: 613.592.3253

### **INTERNATIONAL**

**France**  
Paris  
**LSI Logic S.A.**  
**Immeuble Europa**  
◆ Tel: 33.1.34.63.13.13  
Fax: 33.1.34.63.13.19

**Germany**  
Munich  
◆ **LSI Logic GmbH**  
Tel: 49.89.4.58.33.0  
Fax: 49.89.4.58.33.108

Stuttgart  
Tel: 49.711.13.96.90  
Fax: 49.711.86.61.428

**Italy**  
Milano  
**LSI Logic S.P.A.**  
◆ Tel: 39.039.687371  
Fax: 39.039.6057867

**Japan**  
Tokyo  
◆ **LSI Logic K.K.**  
Tel: 81.3.5463.7821  
Fax: 81.3.5463.7820

Osaka  
◆ Tel: 81.6.947.5281  
Fax: 81.6.947.5287

**Korea**  
Seoul  
**LSI Logic Corporation of Korea Ltd**  
Tel: 82.2.528.3400  
Fax: 82.2.528.2250

**The Netherlands**  
Eindhoven  
**LSI Logic Europe Ltd**  
Tel: 31.40.265.3580  
Fax: 31.40.296.2109

**Singapore**  
Singapore  
**LSI Logic Pte Ltd**  
Tel: 65.334.9061  
Fax: 65.334.4749

Tel: 65.835.5040  
Fax: 65.732.5047

**Sweden**  
Stockholm  
**LSI Logic AB**  
◆ Tel: 46.8.444.15.00  
Fax: 46.8.750.66.47

**Taiwan**  
Taipei  
**LSI Logic Asia, Inc.**  
**Taiwan Branch**  
Tel: 886.2.2718.7828  
Fax: 886.2.2718.8869

**United Kingdom**  
Bracknell  
◆ **LSI Logic Europe Ltd**  
Tel: 44.1344.426544  
Fax: 44.1344.481039

◆ Sales Offices with  
Design Resource Centers



# International Distributors

---

## Australia

New South Wales

### ◆ Reptechnic Pty Ltd

Tel: 612.9953.9844

Fax: 612.9953.9683

## Belgium

### Acal nv/sa

Tel: 32.2.7205983

Fax: 32.2.7251014

## China

Beijing

### LSI Logic International Services Inc.

Tel: 86.10.6804.2534

Fax: 86.10.6804.2521

## France

Rungis Cedex

### Azzurri Technology France

Tel: 33.1.41806310

Fax: 33.1.41730340

## Germany

Haar

### EBV Elektronik

Tel: 49.89.4600980

Fax: 49.89.46009840

Munich

### Avnet Emg GmbH

Tel: 49.89.45110102

Fax: 49.89.42.27.75

Wuennenberg-Haaren

### Peacock AG

Tel: 49.2957.79.1692

Fax: 49.2957.79.9341

## Hong Kong

Hong Kong

### AVT Industrial Ltd

Tel: 852.2428.0008

Fax: 852.2401.2105

## EastEle

Tel: 852.2798.8860

Fax: 852.2305.0640

## India

Bangalore

### Spike Technologies India Private Ltd

◆ Tel: 91.80.664.5530

Fax: 91.80.664.9748

## Israel

Tel Aviv

### Eastronics Ltd

Tel: 972.3.6458777

Fax: 972.3.6458666

## Japan

Tokyo

### Global Electronics Corporation

Tel: 81.3.3260.1411

Fax: 81.3.3260.7100

Technical Center

Tel: 81.471.43.8200

Yokohama-City

### Macnica Corporation

Tel: 81.45.939.6140

Fax: 81.45.939.6141

## The Netherlands

Eindhoven

### Acal Nederland b.v.

Tel: 31.40.2.502602

Fax: 31.40.2.510255

## Switzerland

Brugg

### LSI Logic Sulzer AG

Tel: 41.32.3743232

Fax: 41.32.3743233

## Taiwan

Taipei

### Avnet-Mercuries Corporation, Ltd

Tel: 886.2.2516.7303

Fax: 886.2.2505.7391

### Lumax International Corporation, Ltd

Tel: 886.2.2788.3656

Fax: 886.2.2788.3568

### Prospect Technology Corporation, Ltd

Tel: 886.2.2721.9533

Fax: 886.2.2773.3756

### Serial Semiconductor Corporation, Ltd

Tel: 886.2.2579.5858

Fax: 886.2.2570.3123

## United Kingdom

Maidenhead

### Azzurri Technology Ltd

Tel: 44.1628.826826

Fax: 44.1628.829730

Swindon

### EBV Elektronik

Tel: 44.1793.849933

Fax: 44.1793.859555

◆ Sales Offices with  
Design Resource Centers

