TECHNICAL MANUAL

LSIFC949X Dual Channel Fibre Channel I/O Processor

November 2005

Version 2.0

LSI LOGIC®

DB14-000284-02

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.

DB14-000284-02, November 2005
This document describes the LSI Logic Corporation LSIFC949X Dual Channel Fibre Channel I/O Processor and will remain the official reference source for all revisions/releases of this product until rescinded by an update.

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.

TRADEMARK ACKNOWLEDGMENT
LSI Logic, the LSI Logic logo design, Fusion-MPT, GigaBlaze, MyStorage, and SmarthPath are trademarks or registered trademarks of LSI Logic Corporation. ARM is a registered trademark of ARM Ltd., used under license. All other brand and product names may be trademarks of their respective companies.

DB

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

For a current list of our distributors, sales offices, and design resource centers, view our web page located at

http://www.lsilogic.com/contacts/index.html

Preface

This book is the primary reference and technical manual for the LSIFC949X Dual Channel Fibre Channel I/O Processor. It contains a complete functional description for the LSIFC949X and includes complete physical and electrical specifications for the product.

Audience

This document was prepared for logic designers and applications engineers and is intended to provide an overview of the LSI Logic LSIFC949X and to explain how to use the LSIFC949X in the initial stages of system design.

This document assumes that you have some familiarity with microprocessors and related support devices. The people who benefit from this book are

- engineers and managers who are evaluating the LSIFC949X for possible use in a system
- engineers who are designing the LSIFC949X into a system

Organization

This document has the following chapters and appendixes:

- Chapter 1, **Introduction**, provides a general description of the LSIFC949X.
- Chapter 2, **Fibre Channel Overview**, briefly describes some key elements of Fibre Channel, including layers, topologies, and classes of service.

- Chapter 3, **LSIFC949X Overview**, provides an introduction to the basic features of the LSIFC949X, including the message interface, protocol assist engines, and support components.
- Chapter 4, **Signal Descriptions**, lists and describes the signals on the LSIFC949X.
- Chapter 5, **PCI-X Functional Description**, describes the PCI-X features contained in the LSIFC949X.
- Chapter 6, **Registers,** briefly describes the PCI-X address space, the Configuration registers, and the host interface registers.
- Chapter 7, **Specifications**, describes the electrical specifications of the LSIFC949X and provides pinout information and packaging dimensions.
- Appendix A, **Register Summary**, is a register summary.
- Appendix B, **Reference Specifications**, lists several specifications and applicable World Wide Web URLs that may benefit the reader.
- Appendix C, **Glossary of Terms and Abbreviations**, provides definitions for terms and abbreviations used in this manual.

Related Publications

Fusion-MPT™ Message Passing Interface Specification, Version 1.5, Document No. DB14-000174-03

PCI Local Bus Specification, Revision 2.3

PCI-X Addendum to the PCI Local Bus Specification, Revision 2.0

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. Signals that are active LOW end in a slash (“/”).

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 History

Document Number	Date/Version	Remarks
DB14-000284-02	November 2005 Version 2.0	Section 1.1.1 (page 1-8); modified firmware support for concurrent commands and concurrent logins. Throughout document; changed HOTSWAPEN/ signal name to CPCI_EN/. Table 4.1; added text to 64EN/ and CPCI_EN/ description. Table 4.3; rewrote description for MA[23:0] and added reference to SEN #S11066, “LSIFC949X Design Considerations.” Table 4.4; modified description of GPIO[5:0], LED[4:0]/, and MODE[7:0]. Chapter 6; added Per-Vector Masking Capable bit to MSI Message Control register, modified MSI Enable bit description in MSI Message Control register, added MSI Mask Bits, MSI Pending Bits, MSI-X Capability ID, MSI-X Next Pointer, MSI-X Message Control, MSI-X Table Offset, and MSI-X PBA Offset register descriptions. Chapter 6; changed values in Designed Maximum Cumulative Read Size and Designed Maximum Split Transactions bit fields of the PCI-X Status register. Table 7.2; added maximum I_{DDC} and I_{DDIO} specifications. Modified Table 7.9, “PCI Bidirectional Signals” and Table 7.11, “SSRAM Read/Write/Read Timings.” Replaced Figure 7.4 with one-page top view. Added alphanumeric pad listing tables (Tables 7.14 and 7.15). Various additional minor editorial changes throughout document.
DB14-000284-01	September 2004 Version 0.6	Updated to reflect LSIFC949X support of *PCI Local Bus Specification,* Revision 2.3, and *PCI-X Addendum to the PCI Local Bus Specification,* Revision 2.0; added Section 1.2.5, “INCITS T10 Authorized DIF,” page 1-6; deleted the Snooze Control feature, pad name ZZ (BGA pad number K3 becomes “NC”); changed BGA pad numbers V3, AA13, AB13, AC13, AF12, and AC25 to “NC”; corrected BGA pad number B19 to “AD[48]”; corrected BGA pad number P2 to “RXVSS0”; deleted Section 6.3, “Shared Memory”; updated “Device ID” designations on pages 6-4 and 6-5; made several changes to specifications on pages 7-2, 7-4, 7-5, and 7-6; changed MCLK cycle time on pages 7-7 and 7-8; added separate designation for MXSVDD on pages 4-14 and 7-10; updated Figure 7.5 to package code “7T,” deleted Section 7.4, “Package Thermal Considerations”; and corrected several other minor typos throughout the document.
DB14-000284-00	June 2004 Version 0.5	Initial release.

Contents

Chapter 5
PCI-X Functional Description

Chapter 6
Registers

Figures

Tables

Chapter 1
Introduction

This chapter provides an overview of the LSIFC949X Dual Channel Fibre Channel I/O Processor. The chapter contains the following sections:

1.1 Overview

The LSIFC949X is a high-performance, cost-effective, Dual Channel Fibre Channel (FC) I/O processor. It represents the latest system level integration technology in intelligent I/O processors from LSI Logic. The Storage Area Network (SAN) environment is fully supported with Fibre Channel Protocol (FCP) for SCSI.

1.1.1 Hardware Features

The LSIFC949X supports the following list of hardware features:

- Highly integrated, full duplex, Dual Channel FC I/O processor
- Integrated 4 Gbit/s Dual Channel FC serial link
- 64-bit/66 MHz host PCI bus and 133 MHz PCI-X bus (both are backward compatible with 32-bit/33 MHz)
- 32-bit ARM® RISC processor
- Intelligent, high-performance context management

- Integrated bit error rate (BER) testing (with special test utilities)
- Synchronous SRAM (SSRAM) external memory interface
- Full simultaneous target and initiator operations
- Implementation of common Message Passing Interface (MPI)
- Firmware support for concurrent host commands
  - 128 concurrent commands with internal SRAM only
  - 1000 concurrent commands with 1 Mbyte SRAM
  - 4000 concurrent commands with 4 Mbytes SRAM
- Firmware support for concurrent logins
  - 32 concurrent logins with internal SRAM only
  - 256 concurrent logins with 1 Mbyte SRAM
  - 2048 concurrent logins with 4 Mbytes SRAM
- PC2001 compliant
- Peripheral Component Interface (PCI), Revision 2.3 compliant
- JTAG debug interface
- 544-pin Flip Chip Plastic Ball Grid array (FPBGA)

1.1.2 FC Features

The LSIFC949X supports the following list of FC features:

- BB credit of 16, alternate login of 1 (each channel)
- Up to 126 alias addresses (ALPAs)
- Class 3 connectionless service
- FC-PH compliance
- FC-AL 7.0 compliance
- FC-FCP, FC-PLDA compliance
- FC-FLA compliance
- FCA-IP, IETF-IPFC compliance
- NL_Port (Arbitrated Loop)
- N_Port (point-to-point)

- FL_Port (public loop attach)
- F_Port (Fabric attach)
- AutoNegotiation between link speeds under firmware control; provides automatic interoperability between 1 Gbit/s, 2 Gbit/s and 4 Gbit/s links (independent for each channel)
- Autotopology Negotiation enables automatic interoperability of each LSIFC949X port to the current port type

1.1.3 Software Features

The LSIFC949X supports the following list of software features:

- Fusion-MPT™ drivers
- Optimum server I/O profile with low CPU utilization
- Optimum workstation I/O profile with maximum I/O performance
- Diagnostic capability
- Host driver support for failover and load balancing
- SAN Storage Management
- MyStorage® and SmartPath™ support

1.1.4 OS Support

The LSIFC949X supports the following list of operating systems:

- Windows 2000, Windows Server 2003
- Windows XP
- Solaris X86
- Solaris SPARC
- SuSE and Red Hat Linux

1.1.5 Targeted Applications

The LSIFC949X targets the following list of key applications:

- SANs
- Storage virtualization

- Server clustering environments
- Embedded RAID
- Host main boards
- Routers and bridges

1.2 General Description

The LSIFC949X Dual Channel FC I/O processor is a high-performance, Intelligent I/O processor (IOP) that simultaneously supports mass storage and IP protocols on a full duplex 1 Gbit/s, 2 Gbit/s, or 4 Gbit/s FC link. The sophisticated design and local memory architecture work together to reduce the host CPU and PCI bandwidth required to support FC I/O operations.

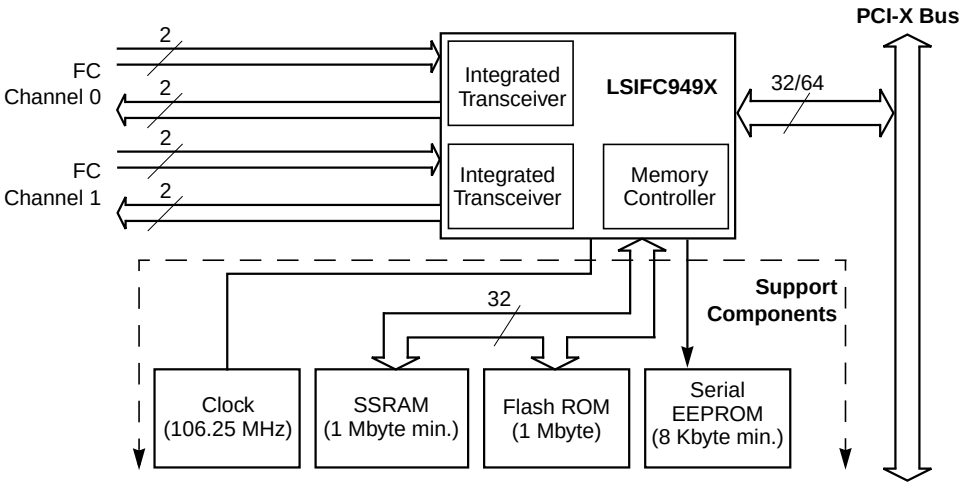
From the host CPU perspective, the LSIFC949X manages the FC link at the exchange level for mass storage (FCP) protocols. The LSIFC949X supports multiple I/O requests per host interrupt in most applications.

From the FC link perspective, the LSIFC949X is a highly efficient NL_Port supporting point-to-point topologies, public and private loop topologies, and the FC switch/attach topology defined under the ANSI X3T11 FC-FS standard. The LSIFC949X uniquely supports FC environments where independent, full duplex transmission is required for maximum FC link efficiency. Special attention has been given to the design to accelerate context switching and link utilization.

The LSIFC949X includes a 64-bit, 66 MHz host PCI interface and a 133 MHz PCI-X interface to the host environment. The host interface minimizes the amount of time spent on the PCI bus for nondata moving activities such as initialization, command, and error recovery. In addition, the host interface has inherent flexibility to support the OEM implementation tradeoffs between CPU, PCI-X, and I/O bandwidth.

The high level of integration in the LSIFC949X controller enables low cost FC implementations. Figure 1.1 shows a typical implementation incorporating the LSIFC949X controller.

Figure 1.1 LSIFC949X Typical Implementation



1.2.1 Multifunction PCI-X

Coupled with the dual channel operation, the LSIFC949X adds multifunction capability on the PCI-X bus. This capability allows the host to see two distinct “channels” or host adapters. Each channel provides full, concurrent support for FCP Initiator and Target protocols.

1.2.2 Autospeed Negotiation

Backward compatibility with 1 Gbit/s and 2 Gbit/s FC devices is maintained through Autospeed Negotiation. After a power-on, loss of signal, or loss of word synchronization for longer than the R_T_TOV time-out, the LSIFC949X performs this operation to determine whether a point-to-point device or all of the devices on a link are either 1, 2, or 4 Gbit/s devices, and it automatically configures itself to be compatible with the devices on the link.

1.2.3 Autotopology Negotiation

The LSIFC949X maintains compatibility with private loop, public loop, and point-to-point topologies through Autotopology Negotiation. The LSIFC949X performs this operation to determine the type of attached link, and automatically configures each LSIFC949X port to the current port type.

1.2.4 Failover and Load Balancing

The LSIFC949X supports two PCI-X functions and two FC ports, which improves performance and provides a redundant path in high-availability systems that require failover capabilities. In case of a Link Failure, the LSIFC949X architecture allows the OS driver to support automatic failover without the need for LSIFC949X intervention. Load Balancing also can be provided in the host driver to partition the I/O workload across each channel of the LSIFC949X.

1.2.5 INCITS T10 Authorized DIF

The LSIFC949X uses the new International Committee for Information Technology Standards (INCITS) T10 authorized Data Integrity Field (DIF) for additional end-to-end data protection. The LSIFC949X provides extended data protection by appending the DIF to block level data, allowing integrity checks at each node of a system for enhanced data integrity and debug capability.

1.3 Hardware Overview

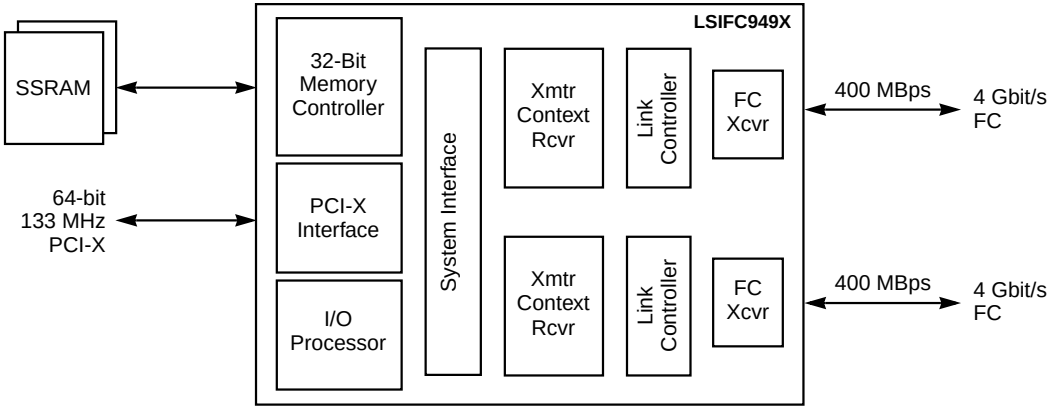
In today's fast growing SAN, storage virtualization, server/workstation, and RAID storage systems marketplaces, higher levels of performance, scalability, and reliability are required to stay competitive.

The LSIFC949X provides the performance and flexibility to meet future FC connectivity requirements.

The LSIFC949X and LSI Logic software drivers provide superior performance and lower host CPU overhead than other competitive solutions. Because of its high level of integration and streamlined architecture, the LSIFC949X provides the highest level of performance in a more cost effective FC solution.

Figure 1.2 shows the functional block diagram for the LSIFC949X. The architecture maximizes performance and flexibility by deploying fixed gates in critical performance areas and utilizing multiple ARM RISC processors (two for context management and one for the I/O processor). Each of the major blocks is described briefly.

Figure 1.2 LSIFC949X Functional Block Diagram



1.3.1 PCI/PCI-X Interface

The LSIFC949X uses a 64-bit (33 MHz, 66 MHz, or 133 MHz) PCI/PCI-X interface or a 32-bit (33 MHz, 66 MHz, or 133 MHz) PCI/PCI-X interface. In addition, support is provided for Dual Address Cycle (DAC), PCI-X power management, Subsystem Vendor ID, Vendor Product Data (VPD), and Message Signaled Interrupt (MSI and MSI-X) and Data Integrity Field (DIF).

1.3.2 32-Bit Memory Controller

The memory controller provides access to Flash ROM and 32-bit Synchronous SRAM. It supports both interleaved and noninterleaved configurations up to a maximum of 4 Mbytes of synchronous SRAM. A general purpose memory expansion bus supports up to 1 Mbyte of Flash ROM.

1.3.3 I/O Processor

The LSIFC949X uses a 32-bit ARM RISC processor to control all system interface and message transport functionality. This frees the host CPU for other processing activity and improves overall I/O performance. The RISC processor and associated firmware can manage an I/O from start to finish without host intervention. The RISC processor also manages the message passing interface.

1.3.4 System Interface

The system interface efficiently passes messages between the LSIFC949X and other I/O agents. It consists of five hardware FIFOs for the message queuing lists: Request Free, Request Post, Reply Free, Reply Post, and High Priority Request. Control logic for the FIFOs is provided within the LSIFC949X system interface with messages stored in external memory.

1.3.5 Integrated 4 Gbit/s Transceivers

The LSIFC949X implements GigaBlaze® 4 Gbit/s transceivers. GigaBlaze is backward-compatible with 2 Gbit/s and 1 Gbit/s systems, using a firmware-implemented “Autospeed Negotiation” for automatic compatibility between 1 Gbit/s, 2 Gbit/s, and 4 Gbit/s links. The integrated 4 Gbit/s transceivers provide a FC-compliant physical interface for cost conscious and real estate limited applications.

1.3.6 Link Controllers

The integrated link controller is FC-AL-2 (Rev. 7.0) compatible and performs all link operations. The controller monitors the Link State and strictly adheres to the Loop Port State Machine, ensuring maximum system interoperability. The link controller interfaces to the integrated transceiver.

1.3.7 Datapath

The transmitter builds sequences based on context information and transmits resulting frames to the FC link using the Link Controller. Each transmitter includes three 2 Kbyte buffers to support frame payloads.

The receivers accept frame data from the Link Controller and DMAs the encapsulated information to local or system memory. Each receiver contains sixteen 2112-byte buffers that support a BB Credit of up to sixteen or an Alternate Login BB Credit of 1 on each channel.

1.3.8 Context Managers

The LSIFC949X uses an ARM RISC processor in each channel to support I/O context swap to external memory and FCP management for both Initiator and Target applications. Context operations include support

for transmit and resource queue management, as well as scatter/gather list management.

1.4 Initiator Operations

The LSIFC949X autonomously handles FCP exchanges upon request from the host. The LSIFC949X generates appropriate sequences and frames necessary to complete the request and provides feedback to the host on the status of the request.

1.5 Target Operations

The LSIFC949X provides for general purpose target functions such as those required for front-end RAID applications.

1.6 Diagnostics

The LSIFC949X provides the capabilities to do a simplified “Link Check” BER test on the link for diagnostic purposes. In a special test mode the controller can transmit and verify a programmed data pattern for link evaluation.

Chapter 2
Fibre Channel Overview

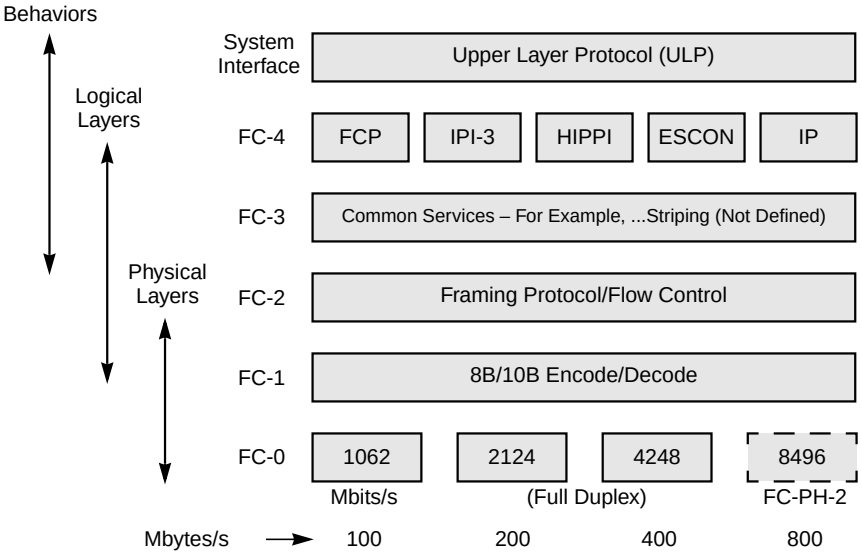
This chapter provides general overview information on Fibre Channel (FC). The chapter contains the following sections:

2.1 Introduction

FC is a high-performance, hybrid interface. It is both a channel and a network interface that contains network features to provide the required connectivity, distance, protocol multiplexing, as well as traditional channel features to retain the required simplicity, repeatable performance, and guaranteed delivery. Popular industry standard networking protocols such as Internet Protocol (IP) and channel protocols such as Small Computer System Interface (SCSI) have been mapped to the FC standard.

The FC structure is defined by five functional layers. These layers, shown in Figure 2.1, define the physical media and transmission rates, encoding scheme, framing protocol and flow control, common services, and the Upper Level Protocol (ULP) interfaces.

Figure 2.1 FC Layers



2.2 FC Layers

The lowest layer, FC-0, is the media interface layer. It defines the physical characteristics of the interface. It includes transceivers, copper-to-optical transducers, connectors, and any other associated circuitry necessary to transmit or receive at 1062 or greater Mbit/s rates over copper or optical cable.

The FC-1 layer defines the 8B/10B encoding/decoding scheme, the transmission protocol necessary to integrate the data and transmit clock, and the receive clock recovery. Implementation of this layer is usually divided between the hardware implementing the FC-0 layer in a transceiver, and the protocol device that implements the FC-2 layer. Specifically, the FC-0 transceivers can include the clock recovery circuitry while the 8B/10B encoding/decoding is provided in the protocol device.

The FC-2 layer defines the rules for the signaling protocol and describes transfer of the frames, sequences, and exchanges. The meaning of the data being transmitted or received is transparent to the FC-2 layer. However, the context between any given set of frames is maintained at the FC-2 layer through the Sequence and Exchange constructs. The

framing protocol creates the constructs necessary to form frames with the data being packetized within the payload of each frame.

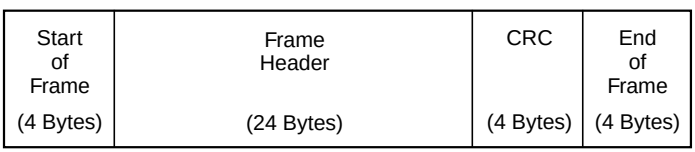
The FC-3 layer provides common services that span multiple N_Ports. Some of these services include Striping, Hunt Groups, and Multicasting. All of these services allow a single port or fabric to communicate to several N_Ports at one time (refer to Section 2.6, “FC Topologies,” on page 2-7 for details).

The FC-4 layer is the top layer defined in the FC. The FC-4 layer provides a seamless integration of existing standards. It specifies the mapping of ULPs to the layers below. Some of these ULPs include SCSI and IP. Each of these ULPs is defined in its own ANSI document.

2.3 Frames

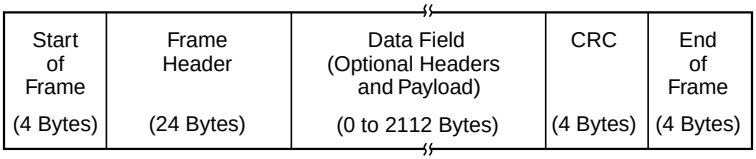
There are two types of frames used in FC: Link Control frames and Data frames. Link Control frames, which contain no payload, are flow control responses to Data frames. An example of a Link Control frame is the ACK frame (Figure 2.2).

Figure 2.2 Link Control Frame



A Data frame is any frame that contains data in the payload field. An example of a Data frame is the LOGIN frame (Figure 2.3).

Figure 2.3 Data Frame



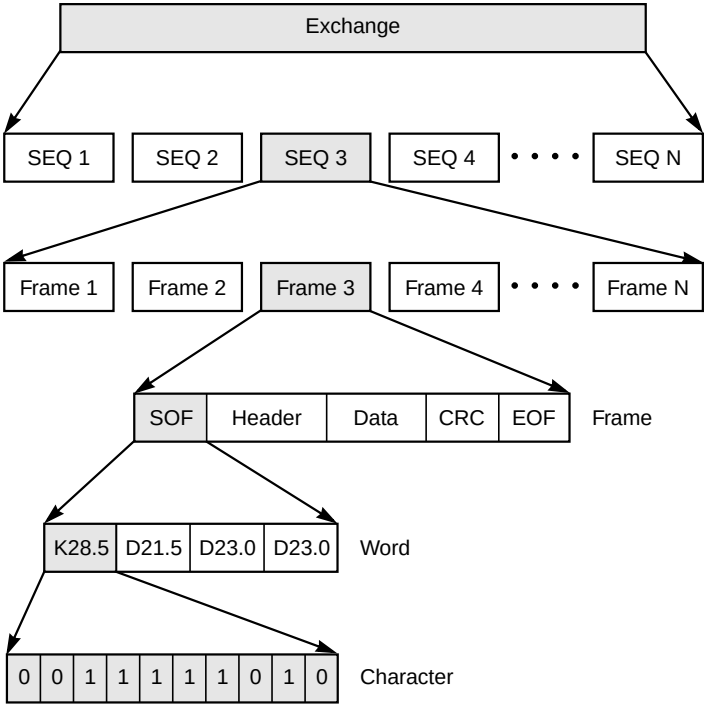
In FC, an Ordered Set is a group of four 10-bit characters that provide low level link functions, such as frame demarcation and signaling

between two ends of a link. All frames start with a Start-of-Frame (SOF) and end with an End-of-Frame (EOF) Ordered Set. Each frame contains at least a 24-byte header defining such things as Destination and Source ID, Class of Service and type of frame (for example, FCP or FC-LE). The biggest field within a frame can be the payload field. If the frame is a Link Control frame, then there is no payload. If it is a Data frame, then the frame contains a payload field of up to 2112 bytes. Finally, the frame includes a CRC field used for detection of transmission errors, followed by the EOF Ordered Set.

2.4 Exchanges

Figure 2.4 outlines the FC hierarchical data structures. At the most elemental level, four 8B/10B encoded characters make up an FC word. An FC frame is a collection of FC words. An FC Sequence is made up of one or more frames, and a FC Exchange is made up of one or more sequences.

Figure 2.4 Exchange to Character



The following discussion illustrates an Exchange by considering a typical parallel SCSI I/O. In parallel SCSI, several phases make up the I/O. These phases include Command, Data, Message, and Status.

Using the FCP for the SCSI ULP, these phases can be mapped into the other lower FC layers. Figure 2.5 shows the components that make up the FCP exchange.

Figure 2.5 FCP Exchange

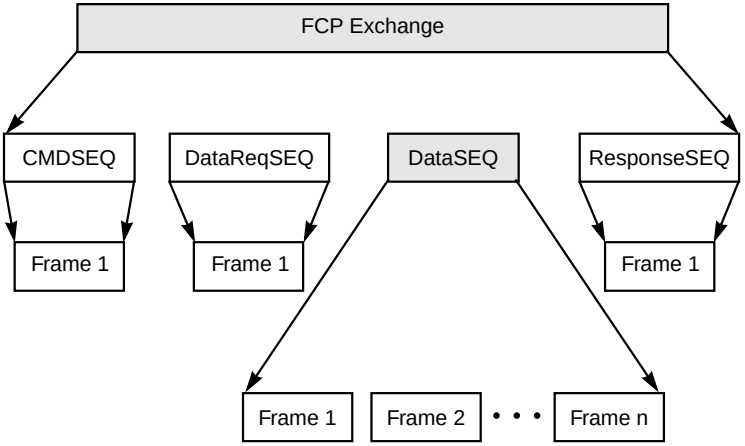
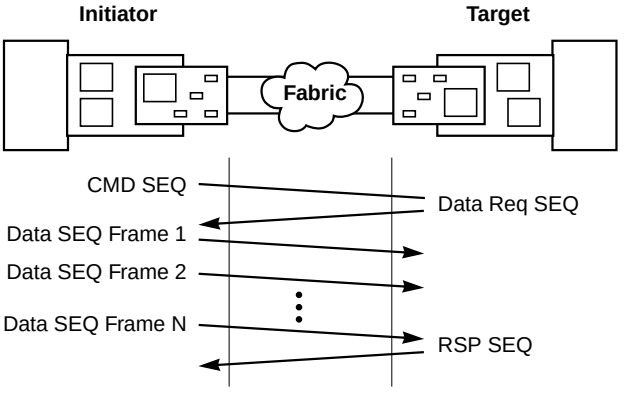


Figure 2.6 shows how the Exchange flows between the Initiator and Target. The Initiator starts the FCP Exchange by sending a Command sequence containing one frame to the Target. The frame payload contains the Command Descriptor Block (CDB). The Target then responds with a Data Delivery Request sequence containing one frame. The payload of this frame contains a XFER_RDY response. When the Initiator receives the Target's response, it begins to send the Data sequence(s), which may contain one or more frames. This is analogous to parallel SCSI DATA_OUT phase. When the Target has received the last frame of the data sequence(s), it sends a Response sequence containing one frame to the Initiator, thus concluding the FCP Exchange.

Figure 2.6 Write Event Trellis



2.5 FC Ports

FC devices are called nodes. Each node has at least one port to provide access to other ports in other nodes. The “port” is the hardware entity within a node that performs data communications over the FC link.

Various types of ports are defined within the FC standard, based on the location of the port and the topology associated with it. The most commonly used ports are N_Ports, NL_Ports, F_Ports, and FL_Ports. These types of ports appear in Figure 2.7, Figure 2.8, and Figure 2.9.

2.6 FC Topologies

Topologies are defined, based on the capability and the presence or absence of Fabric between the N_Ports:

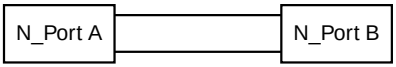
- Point-to-point topology
- Fabric topology
- Arbitrated Loop topology

FC-PH protocols are topology-independent. Attributes of a Fabric may restrict operation to certain communication models.

2.6.1 Point-to-Point Topology

The topology shown in Figure 2.7, in which communication between N_Ports occurs without the use of Fabric, is defined as point-to-point.

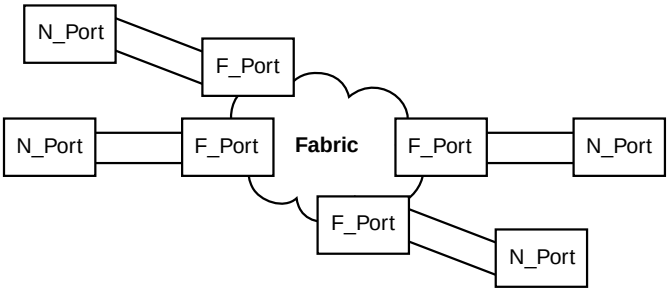
Figure 2.7 Point-to-Point Topology



2.6.2 Fabric Topology

Figure 2.8 illustrates multiple N_Ports interconnected by a Fabric. This topology uses the Destination_Identifier (D_ID) embedded in the frame header to route the frame through a Fabric to the desired Destination N_Port.

Figure 2.8 Fabric Topology

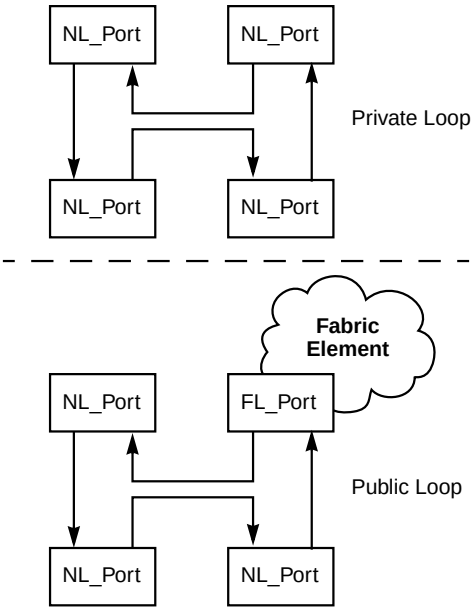


2.6.3 Arbitrated Loop Topology

The Arbitrated Loop topology permits between 2 and 127 L_Ports to communicate without the use of a Fabric, as in Fabric topology. The Arbitrated Loop supports a maximum of one point-to-point circuit at a time. When two L_Ports are communicating, the Arbitrated Loop topology supports simultaneous, symmetrical bidirectional flow.

Figure 2.9 illustrates two independent Arbitrated Loop configurations, each with multiple L_Ports attached. Each line in the figure between L_Ports represents a single fibre. The lower configuration shows an Arbitrated Loop composed of three NL_Ports and one FL_Port (a public loop).

Figure 2.9 Arbitrated Loop Topology



2.7 Classes of Service

There are several classes of service in FC. The different classes are distinguished from each other in three ways: by the level of guarantee for data being delivered, the order in which data is delivered, and how data flow control is maintained.

Class 1 is a dedicated connection between two N_Ports. The data delivered is guaranteed with a required acknowledgement frame (ACK), which a Class 1 device uses for flow control. All frames are received in order.

Class 2 is a connectionless class. The data delivered is guaranteed with an ACK frame. The frames can be received out of order. Class 2 uses both ACK frames and the R_RDY Ordered Set for flow control.

Class 3 is also a connectionless class (the data being delivered is not guaranteed). The frames can be received out of order. Class 3 uses only the R_RDY Ordered Set for flow control.

Intermix is an enhancement of Class 1 service. A dedicated Class 1 connection may waste fabric bandwidth while frames are not being transmitted or received between two N_Ports. To recover some of this bandwidth, Intermix allows Class 2 and Class 3 frames to be transmitted/received between Class 1 frames. N_Ports advertising Intermix capability must be capable of receiving Class 2 and Class 3 frames from other N_Ports while maintaining the original Class 1 link.

Chapter 3
LSIFC949X Overview

This chapter provides a general description of the LSIFC949X Dual Channel Fibre Channel I/O processor firmware. The chapter contains the following sections:

3.1 Introduction

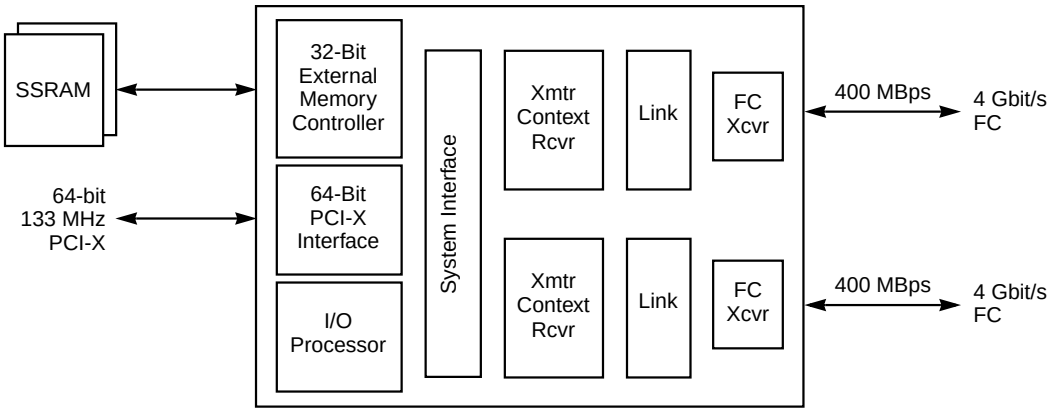
The LSI Logic LSIFC949X connects a host to a high-speed FC link. The FCP ANSI standard, FC Private Loop Direct Attach, and Fabric Loop Attach profiles are supported with a sophisticated firmware implementation. All profiles, specifications, and interoperability maintained by the LSIFC949X are listed in Appendix B, “Reference Specifications.”

Although optimized for a 64-bit PCI-X interface to communicate with the system CPU(s) and memory, the LSIFC949X also supports a 32-bit Peripheral Component Interface (PCI) environment. The system interface to the LSIFC949X minimizes the amount of PCI-X bandwidth required to support I/O requests. A packetized message passing interface reduces the number of single cycle PCI bus cycles. All FC Data traffic on the PCI-X bus occurs with zero wait state bursts across the PCI-X bus.

The intelligent LSIFC949X architecture allows the system to specify I/Os at the command level. The LSIFC949X manages I/Os at the Frame, Sequence and Exchange level. Error detection and I/O retries are also handled by the LSIFC949X, allowing the system to offload part of the exception handling work from the system driver.

Data Flows – The LSIFC949X uses a 64-bit (33 MHz, 66 MHz, or 133 MHz) PCI-X interface to pass control and data information between the system and the protocol controller. This interface is managed by the PCI-X Interface block, as shown in Figure 3.1. It is backward compatible with 32-bit/33 or 66 MHz buses.

Figure 3.1 LSIFC949X Block Diagram



For incoming serial data, the physical link transfers the data to Link Control using the GigaBlaze Integrated Transceiver. The Link Controller analyzes the received frame, and if appropriate, it passes the frame to the Receiver. The Receiver strips off the frame header and places it in a separate header buffer while the data in the frame payload is placed in a data buffer. The Frame Receiver uses the Receive Context Manager to manage the order and priority of the received frame. The data contained in the Receiver buffers is associated with a specific scatter/gather entry and passed on to the PCI-X Interface. The data also requests the PCI-X bus and bursts the data into system memory.

The I/O processor (IOP), with its firmware, provides the translation from FC specific protocols to the high level Block Storage and SCSI message interface. This translation allows the LSIFC949X to be integrated into the system as if it were a native Parallel SCSI device, hiding all FC-unique

characteristics. Internal communication between the IOP and the Context manager occurs over an internal bus, which also is connected to an External Memory Controller. The IOP uses the External Memory Controller to access local memory. This memory contains the firmware, as well as the dynamic data structures used by the firmware.

3.2 Message Interface

The LSIFC949X system interface is a high-performance, packetized, mailbox architecture that leverages the intelligence in the LSIFC949X to minimize traffic on the PCI-X bus.

The Fusion-MPT architecture also provides a High Priority Request FIFO to provide high priority request free messages to the host on reads, and to accept high priority request post messages from the host on writes. The High Priority Request Post FIFO is similar to the Request Post FIFO, except that the LSIFC949X processes requests from the High Priority Request Post FIFO before processing requests from the Request Post FIFO. This high-priority queue has dedicated resources which do not become depleted when the request queue gets full.

There are two basic constructs in the Message Interface. The first construct, the Message, communicates between the system and the LSIFC949X. Messages are moved between the system(s) and the LSIFC949X using the second construct, a Transport mechanism.

3.2.1 Messages

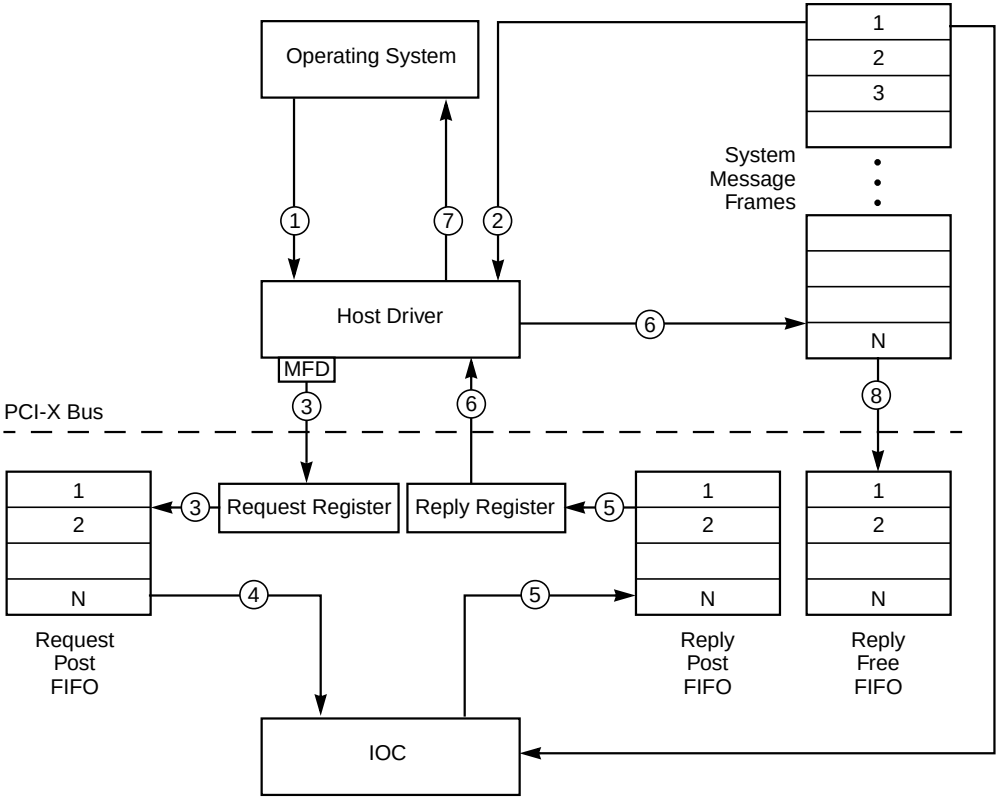
The LSIFC949X uses two types of messages to communicate with the system. Request messages are created by the system to “request” an action by the LSIFC949X. Reply messages are used by the LSIFC949X to send status information back to the system. Request message data structures are up to 128 bytes in length. The message includes a message header and a payload. The header includes information to uniquely identify the message. The payload is specific to the Request itself, and is unique for SCSI and Target messages. For more information regarding the details of the message format, refer to the *Fusion-MPT™ Message Passing Interface Specification,* Version 1.5.

3.2.2 Message Flow

Before Requests can be posted to the LSIFC949X, the system must allocate and initialize a pool of message frames, and provide a mechanism to assign individual message frames on a per-request basis. The host also must provide one message frame per target LUN, and prime the Reply Free FIFOs for each function with the physical address of these message frames. When allocation has been completed, requests flow from the host to the LSIFC949X, as represented below and in Figure 3.2.

1. The host driver receives an I/O request from the operating system.
2. The host driver allocates a system message frame (SMF) and builds an I/O request message within the SMF. The allocation method is the responsibility of the host driver.
3. The host driver creates the Message Frame Descriptor (MFD) and writes the MFD to the Request Post FIFO.
4. The I/O Controller (IOC) reads the MFD from the Request Post FIFO and DMAs the request to a local message frame.
5. The IOC sends the appropriate Fibre Channel request and subsequently receives the reply from the target.
   - If the I/O status is successful, the IOC writes the MessageContext value, plus turbo reply bits, to the Reply Post FIFO, which automatically generates a system interrupt.
   - If the I/O status is not successful, the IOC pops a reply message frame from the Reply Free FIFO and generates a reply message in the reply message frame. The IOC then writes the system physical address of the reply message frame to the Reply Post FIFO, which generates a system interrupt.
6. The host driver receives a system interrupt and reads the Reply register. If there are no posted messages, the system reads the value 0xFFFFFFFF.
7. The host driver responds to the operating system appropriately.
8. If the I/O status is not successful, the host driver returns it to the Reply Free FIFO.

Figure 3.2 LSIFC949X Message Flow



3.3 SCSI Message

The SCSI message interface provides the most direct interface for block-oriented storage media. This includes disk drives and tape devices.

The SCSI I/O path translates a SCSI Command Descriptor Block (CDB) into a Fibre Channel Protocol (FCP) exchange. All FC device and target discovery operations are managed completely within the LSIFC949X. FC target devices are assigned a logical (bus, target ID) identifier, and are accessed by the system as if they were parallel SCSI devices. The system is responsible for scanning the target devices and identifying LUNs on the target devices.

In general, the system is responsible for retrying operations at an I/O request level. The LSIFC949X is responsible for responding to bus protocol-specific errors and exceptions and retrying bus sequences within the scope of an I/O operation. The system is also responsible for maintaining a timer for SCSI I/O operations if this is required by the host system. The host driver may use the provided SCSI Task Management functions to terminate one or more I/O operations when a timeout occurs. For details regarding the SCSI Message Class, refer to the *Fusion-MPT™ Message Passing Interface Specification,* Version 1.5.

3.4 Target Message

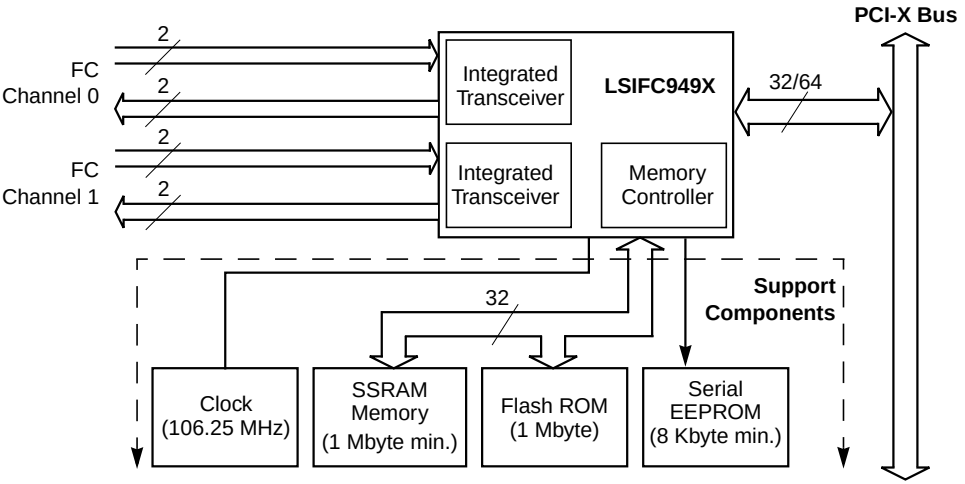
The Target Interface allows the LSIFC949X to be used as the system interface for FC bridge controllers. The LSIFC949X provides an FCP exchange level message interface that routes commands to the system. The system identifies the appropriate data, and passes a Scatter Gather List (SGL) to the LSIFC949X describing the data to transfer. A single Target message directs the LSIFC949X to send a Xfer_Rdy, as needed, and to transfer data and an FCP response. Target specific Process Login/Logout is managed by the system. Refer to the *Fusion-MPT™ Message Passing Interface Specification,* Version 1.5, for details on the Target Message Class.

3.5 Support Components

The memory controller block within the LSIFC949X provides access to external local memory resources required to manage FCP.

The following sections provide guidance in choosing the support components necessary for a fully functional implementation using the LSIFC949X. Figure 3.3 shows an LSIFC949X typical implementation diagram.

Figure 3.3 LSIFC949X Typical Implementation



3.5.1 SSRAM Memory

The primary function of this memory is to store data structures used by the LSIFC949X to manage exchanges and transmit and receive queues. The SSRAM memory also stores part of the run time image of the LSIFC949X firmware, such as initialization and error recovery code. The mainline code is stored within the internal LRAM for performance reasons.

The LSIFC949X uses a 32-bit, nonmultiplexed memory bus to access the SSRAM. This memory bus has the capability to address up to 4 Mbytes of SSRAM.

The LSIFC949X firmware also supports optional byte wide parity error detection. This configurable option is specified as a serial EEPROM parameter.

The amount of SSRAM (1 Mbyte) determines the maximum number of outstanding Request Messages (1024). This roughly equates to the maximum number of outstanding I/O requests pending in the LSIFC949X.

The LSIFC949X also provides an internal SRAM, which allows the chip to function without an external SRAM attached. The number of concurrent commands and concurrent logins is reduced in this mode of operation (128 concurrent commands and 32 concurrent logins).

3.5.2 Flash ROM

The memory controller in the LSIFC949X also manages an optional Flash ROM. If present, the Flash ROM stores the firmware for the LSIFC949X, and if desired, the Intel BIOS and/or Solaris Open Boot BIOS software.

If the Flash ROM is not used, then the host platform is responsible for downloading the IOP firmware to the LSIFC949X through the PCI-X interface. The LSIFC949X supports a simple register handshake interface for firmware download. Firmware may be directly written to the LSIFC949X internal memory and external SSRAM through this interface. Details of this implementation are available in the *Fusion-MPT™ Message Passing Interface Specification,* Version 1.5. Flash ROM is optional for the LSIFC949X, but it is required for applications that require Intel or Solaris BIOS software.

The Flash ROM is accessed using the upper 8 bits of the Memory Interface. If a Flash ROM is to be used, then it should have a capacity of 1 Mbyte with a maximum access time of 150 ns. Refer to the *Fusion-MPT™ Message Passing Interface Specification,* Version 1.5*,* for details on the programming of the Flash ROM.

3.5.3 Serial EEPROM

The serial EEPROM stores nonvolatile data for the LSIFC949X, such as the World Wide Name, VPD, and other vendor-specific information. The SEEPROM data is programmed by the firmware, so the firmware must

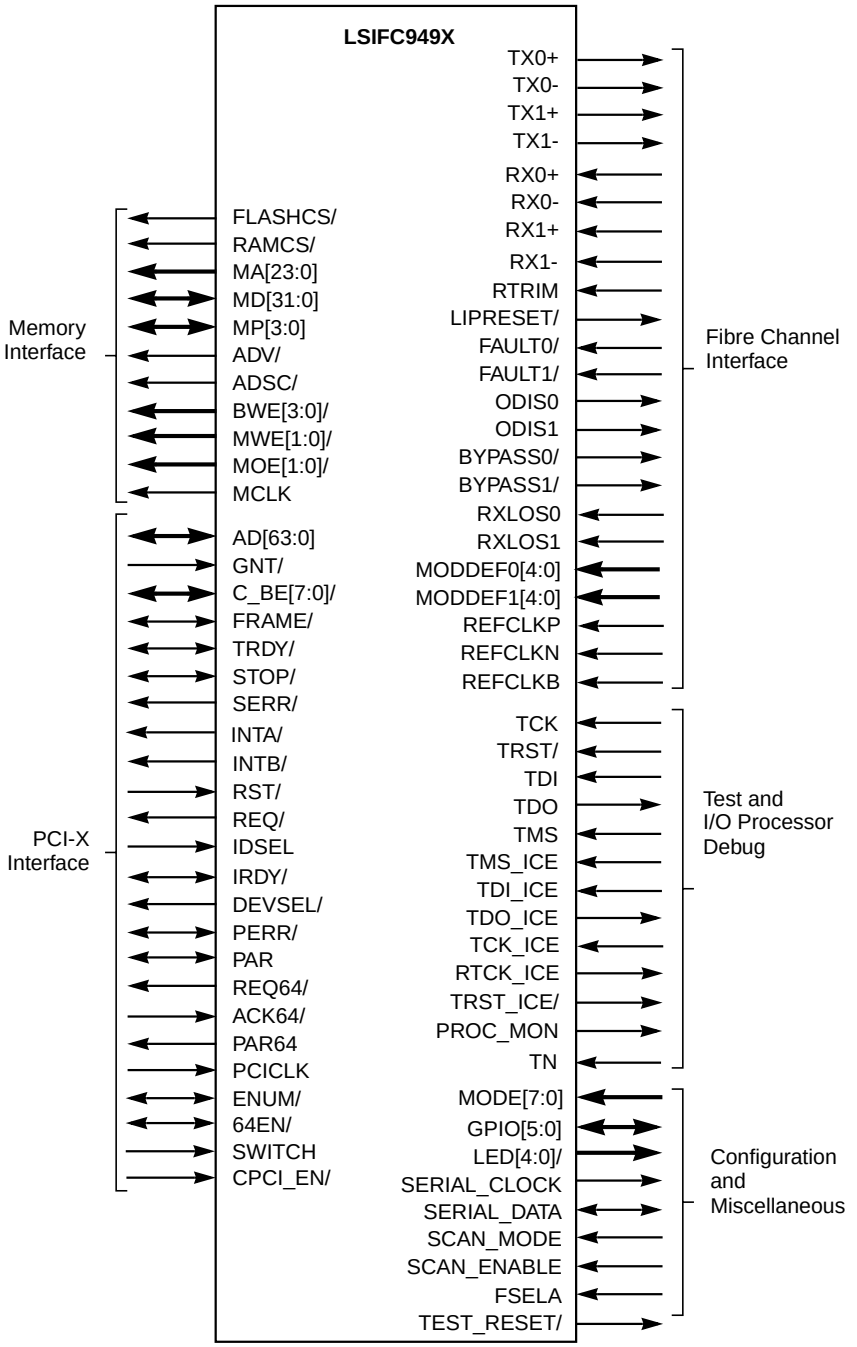
be downloaded and running before the SEEPROM is programmed. The required size of the SEEPROM is 64 Kbits / 8 Kbytes.

Chapter 4
Signal Descriptions

This chapter contains signal descriptions for the LSIFC949X. A slash (/) indicates an active LOW signal, I/O = bidirectional signal, I = input signal, O = output signal, T/S = 3-state, and S/T/S = sustained 3-state. The chapter contains the following sections:

Figure 4.1 on page 4-2 is a functional signal grouping for the chip.

Figure 4.1 LSIFC949X Functional Signal Grouping



4.1 PCI/PCI-X Interface

Table 4.1 lists the PCI/PCI-X Interface signals.

Table 4.1 PCI/PCI-X Interface

Signal	I/O	BGA Pad No.	Pad Type	Description
PCICLK	I	AF22	PCI In	**Clock**. Refer to the *PCI Local Bus Specification, Revision 2.3,* and the *PCI-X Addendum to the PCI Local Bus Specification, Revision 2.0,* for this signal description.
RST/	I	AA26	PCI In	**Reset**. Refer to the *PCI Local Bus Specification, Revision 2.3*, and the *PCI-X Addendum to the PCI Local Bus Specification, Revision 2.0,* for this signal description.
GNT/	I/O	AA21	BiDir PCI	**Grant**. Refer to the *PCI Local Bus Specification, Revision 2.3,* and the *PCI-X Addendum to the PCI Local Bus Specification, Revision 2.0,* for this signal description.
REQ/	I/O	Y25	BiDir PCI	**Request**. Refer to the *PCI Local Bus Specification, Revision 2.3*, and the *PCI-X Addendum to the PCI Local Bus Specification, Revision 2.0,* for this signal description.
REQ64/	I/O	J21	BiDir PCI	**Request64**. Refer to the *PCI Local Bus Specification, Revision 2.3,* and the *PCI-X Addendum to the PCI Local Bus Specification, Revision 2.0,* for this signal description.
ACK64/	S/T/S	J24	BiDir PCI	**Acknowledge64**. Refer to the *PCI Local Bus Specification, Revision 2.3,* and the *PCI-X Addendum to the PCI Local Bus Specification, Revision 2.0,* for this signal description.
AD[63:0]	T/S	See Table 7.14 and Table 7.15 on pages 7-10 and 7-12, respectively.	BiDir PCI	**Address and Data**. Refer to the *PCI Local Bus Specification, Revision 2.3,* and the *PCI-X Addendum to the PCI Local Bus Specification, Revision 2.0,* for this signal description.
C_BE[7:0]/	T/S	K20, J23, J22, J20, AC17, Y19, U26, P22	BiDir PCI	**Command and Byte Enables**. Refer to the *PCI Local Bus Specification, Revision 2.3*, and the *PCI-X Addendum to the PCI Local Bus Specification, Revision 2.0*, for this signal description.
(Sheet 1 of 4)				

Table 4.1 PCI/PCI-X Interface (Cont.)

Signal	I/O	BGA Pad No.	Pad Type	Description
IDSEL	I/O	Y26	BiDir PCI	**Initialization Device Select**. Refer to the *PCI Local Bus Specification, Revision 2.3,* and the *PCI-X Addendum to the PCI Local Bus Specification, Revision 2.0,* for this signal description.
FRAME/	S/T/S	AA22	BiDir PCI	**Cycle Frame**. Refer to the *PCI Local Bus Specification, Revision 2.3,* and the *PCI-X Addendum to the PCI Local Bus Specification, Revision 2.0,* for this signal description.
IRDY/	S/T/S	AB23	BiDir PCI	**Initiator Ready**. Refer to the *PCI Local Bus Specification, Revision 2.3,* and the *PCI-X Addendum to the PCI Local Bus Specification, Revision 2.0,* for this signal description.
TRDY/	S/T/S	V23	BiDir PCI	**Target Ready**. Refer to the *PCI Local Bus Specification, Revision 2.3*, and the *PCI-X Addendum to the PCI Local Bus Specification, Revision 2.0,* for this signal description.
DEVSEL/	I/O	U21	BiDir PCI	**Device Select**. Refer to the *PCI Local Bus Specification, Revision 2.3,* and the *PCI-X Addendum to the PCI Local Bus Specification, Revision 2.0,* for this signal description.
STOP/	S/T/S	W20	BiDir PCI	**Stop**. Refer to the *PCI Local Bus Specification, Revision 2.3,* and the *PCI-X Addendum to the PCI Local Bus Specification, Revision 2.0,* for this signal description.
PERR/	S/T/S	U23	BiDir PCI	**Parity Error**. Refer to the *PCI Local Bus Specification, Revision 2.3,* and the *PCI-X Addendum to the PCI Local Bus Specification, Revision 2.0,* for this signal description.
SERR/	O	V24	BiDir PCI	**System Error**. Refer to the *PCI Local Bus Specification, Revision 2.3,* and the *PCI-X Addendum to the PCI Local Bus Specification, Revision 2.0,* for this signal description.
PAR	T/S	U25	BiDir PCI	**Parity**. Refer to the *PCI Local Bus Specification, Revision 2.3,* and the *PCI-X Addendum to the PCI Local Bus Specification, Revision 2.0,* for this signal description.
(Sheet 2 of 4)				

Table 4.1 PCI/PCI-X Interface (Cont.)

Signal	I/O	BGA Pad No.	Pad Type	Description
PAR64	I/O	H26	BiDir PCI	**Parity64**. Refer to the *PCI Local Bus Specification, Revision 2.3,* and the *PCI-X Addendum to the PCI Local Bus Specification, Revision 2.0,* for this signal description.
INTA/	O	AB24	BiDir PCI	**Interrupt A**. Refer to the *PCI Local Bus Specification, Revision 2.3,* and the *PCI-X Addendum to the PCI Local Bus Specification, Revision 2.0,* for this signal description.
INTB/	O	AA24	BiDir PCI	**Interrupt B**. Refer to the *PCI Local Bus Specification, Revision 2.3,* and the *PCI-X Addendum to the PCI Local Bus Specification, Revision 2.0,* for this signal description.
ENUM/	O	AA23	BiDir PCI	**Enumeration Interrupt**. This signal must be asserted by a hot swap capable card immediately after insertion and during removal. This signal notifies the system host either that a board has been freshly inserted or that one is about to be extracted, and informs the system host that the configuration of the system has changed. The system host then can perform any necessary maintenance such as installing a device driver upon board insertion, or quiescing a device driver and the board, prior to extracting the board.
64EN/	I/O	AC26	BiDir PCI	**PCI Bus Width Enable**. This signal indicates the width of the bus when hot swap capability is enabled. Use an external pull-up on this signal when CompactPCI is enabled. Float this signal when CompactPCI is disabled.
SWITCH	I	AE13	PCI In (pull-down)	**Insertion/Deassertion Indicator**. This signal is an input to the LSIFC949X to signal the insertion or impending extraction of a board. This signal causes the assertion of ENUM/. The operator normally activates the switch (actuator), waits for the illumination of the LED, and then extracts the board.

(Sheet 3 of 4)

Table 4.1 PCI/PCI-X Interface (Cont.)

Signal	I/O	BGA Pad No.	Pad Type	Description
CPCI_EN/	I	Y13	PCI In (pull-up)	**CompactPCI Enable**. When this signal is LOW, the LSIFC949X is configured to conform to hot swap protocol. This includes changing the bus width detection method, the addition of configuration registers, and support for the ENUM/, BLUELED/, and SWITCH pins. When CPCI_EN/ is active (low), pull up MA[7] to force conventional PCI mode.
BLUELED/	O	AE10	3.3 V BiDir 8 mA with pull-up	**BLUELED/**. This signal drives a blue LED that is mounted on the front of hot swap capable host adapters. This signal indicates that the system software has been placed in a state for orderly extraction of the board.
(Sheet 4 of 4)				

4.2 Fibre Channel Interface

Table 4.2 lists the Fibre Channel Interface signals.

Table 4.2 Fibre Channel Interface

Signal	I/O	BGA Pad No.	Pad Type	Description
TX0+	O	N6	Diff Tx	Transmit differential data (Channel 0).
TX1+	O	P7	Diff Tx	Transmit differential data (Channel 1).
TX0–	O	N7	Diff Tx	Transmit differential data (Channel 0).
TX1–	O	P6	Diff Tx	Transmit differential data (Channel 1).
RX0+	I	N4	Diff Rx	Receive differential data (Channel 0).
RX1+	I	P4	Diff Rx	Receive differential data (Channel 1).
RX0–	I	N5	Diff Rx	Receive differential data (Channel 0).
RX1–	I	P5	Diff Rx	Receive differential data (Channel 1).
(Sheet 1 of 4)				

Table 4.2 Fibre Channel Interface (Cont.)

Signal	I/O	BGA Pad No.	Pad Type	Description
RTRIM	I	V2		**Trim Resistor**. This pin is the analog current reference for the integrated transceiver core. A 3.01 kΩ ± 1% resistor should be tied from the RTRIM pad to either the RXVDD0 or the RXVDD1 pin.
LIPRESET/	O	AD9	3.3 V BiDir 4 mA	**Loop Initialization Primitive Reset**. This pin is asserted LOW when a selective reset is received that is targeted to an alias of this device. This pin is asserted for 1–2 ms after the last LIPr is received.
FAULT0/	I	Y8	3.3 V TTL Input with pull-up	**Electrical Fault**. This active-LOW pin indicates that an electrical fault has been detected by the channel 0 PHY device/module and, if the module has a laser, the laser has been turned off. This pin causes no interrupt or other reaction. It is assumed that a Link Failure occurs, and that the register bit reporting the value of this pin diagnoses the problem.
FAULT1/	I	AC10	3.3 V TTL Input with pull-up	**Electrical Fault**. This active-LOW pin indicates that an electrical fault has been detected by the channel 1 PHY device/module and, if the module has a laser, the laser has been turned off. This pin causes no interrupt or other reaction. It is assumed that a Link Failure occurs, and that the register bit reporting the value of this pin diagnoses the problem.
ODIS0	O	AC5	3.3 V BiDir 4 mA	**Output Disable, channel 0.** This output, when asserted, disables an external GBIC or MIA transmitter for channel 0. This output also clears a module fault.
ODIS1	O	AB9	3.3 V BiDir 4 mA	**Output Disable, Channel 1.** This output when asserted disables an external GBIC or MIA transmitter for channel 1. This output also clears a module fault.
(Sheet 2 of 4)				

Table 4.2 Fibre Channel Interface (Cont.)

Signal	I/O	BGA Pad No.	Pad Type	Description
BYPASS0/	O	AC9	3.3 V BiDir 4 mA	**Bypass**. This line is driven LOW when the LSIFC949X Link Controller block determines that channel 0 of the device is operating in a loop environment and that the device has entered a bypassed mode. This may be caused by an internal request or by a loop primitive generated at another node.
BYPASS1/	O	AA10	3.3 V BiDir 4 mA	**Bypass**. This line is driven LOW when the LSIFC949X Link Controller block determines that channel 1 of the device is operating in a loop environment and the device has entered a bypassed mode. This may be caused by an internal request or a loop primitive generated at another node.
RXLOS0	I	AB10	3.3 V 4 mA BiDir with pull-up	**Received Signal Loss**. This line is driven HIGH, disabling the on-chip receiver, when the GBIC for channel 0 of the LSIFC949X detects a loss of signal. If enabled through the Link Control register, this signal becomes an output test strobe.
RXLOS1	I	AA7	3.3 V 4 mA BiDir with pull-up	**Received Signal Loss**. This line is driven HIGH, disabling the on-chip receiver, when the GBIC for channel 1 of the LSIFC949X detects a loss of signal. If enabled through the Link Control register, this signal becomes an output test strobe.
MODDEF0[4:0]	I/O	AE6, AF5, AF6, AA6, Y7	3.3 V BiDir 8 mA with pull-up	**Module Identifiers**. GBIC and pluggable small form factor (SFP) optical module Identifiers (channel 0).
MODDEF1[4:0]	I/O	AE7, AF7, AE8, AF8, AB6	3.3 V BiDir 8 mA with pull-up	**Module Identifiers**. GBIC and pluggable small form factor (SFP) optical module Identifiers (channel 1).
REFCLKP	I	U7	3.3 V Schmitt Input	**FC Reference Clock**. (106.25 MHz ± 100 ppm). Refer to the oscillator requirements section of SEN #S11066, “LSIFC949X Design Considerations,” for further information regarding the REFCLKP and REFCLKN inputs.
(Sheet 3 of 4)				

Table 4.2 Fibre Channel Interface (Cont.)

Signal	I/O	BGA Pad No.	Pad Type	Description
REFCLKN	I	V6	3.3 V Schmitt Input	**FC Reference Clock**. (106.25 MHz ± 100 ppm). If using a single crystal for the FC reference clock, tie the crystal to REFCLKP, and tie REFCLKN to a resistor terminator.
REFCLKB	I	V4	3.3 V Schmitt Input	**Internal Reference Clock**. (use 106.25 MHz). This pin will typically be tied to the REFCLKP pin.
(Sheet 4 of 4)				

4.3 Memory Interface

Table 4.3 shows the Memory Interface signals.

Table 4.3 Memory Interface

Signal	I/O	BGA Pad No.	Pad Type	Description
MD[31:0][1]	I/O	See Table 7.14 and Table 7.15 on pages 7-10 and 7-12, respectively.	3.3 V BiDir 4 mA with pull-down	**SSRAM Read/Write Data**. *See table note [1] on page 4-11*.
MP[3:0]	I/O	F6, G7, G2, G1	3.3 V 4 mA BiDir with pull-up	**Memory Parity**. Byte lane parity is as follows: • MP [0]: Parity for MD[7: 0] • MP [1]: Parity for MD[15: 8] • MP [2]: Parity for MD[23:16] • MP [3]: Parity for MD[31:24] Memory Parity may be optionally even, odd, or none (not used) as defined in the LSIFC949X Programming Model.
(Sheet 1 of 3)				

Table 4.3 Memory Interface (Cont.)

Signal	I/O	BGA Pad No.	Pad Type	Description
MA[23:0]	I/O	See Table 7.14 and Table 7.15 on pages 7-10 and 7-12, respectively.	3.3 V output with pull-down	**SSRAM/FLASH ROM Address**. The MA[19:0] pins are also used at power-on to provide configuration information to the LSIFC949X. The LSIFC949X uses MA[12:11] to determine the external Flash ROM configuration, MA[2] to determine whether to operate as a single-function or dual-function PCI device, and MA[0] to indicate the use of an external 8 KByte EEPROM. The definitions for these power-on sense options are shown below:[2] MA[12:11] 01 = No Flash ROM present 10 = 1 MByte Flash ROM MA[2] 0 = Two PCI-X functions 1 = One PCI-X function MA[0] 0 = N/A 1 = 8 KByte EEPROM Refer to SEN #S11066, “LSIFC949X Design Considerations,” for further information about the power-on sense definitions.
MOE[1:0]/	O	K4, H7	3.3 V BiDir 8 mA	**Memory Output Enable**. When asserted LOW, the selected SRAM or Flash (MOE[1]/) device may drive data. This signal is typically an asynchronous input to SRAM and/or Flash devices. The two output enables allow for interleaving configurations, with MOE[0]/ being the only output enable used for a noninterleaved implementation.
MWE[1:0]/	O	K5, G6	3.3 V BiDir 4 mA	**Memory Write Enables**. These active-LOW bank write enables are required for interleaving configurations. MWE[0]/ is the only write enable used for a noninterleaved implementation.
FLASHCS/	O	J3	3.3 V BiDir 4 mA	**FLASH Chip Select**. This active-LOW chip select allows connection of a single, 8-bit FLASH ROM device.
(Sheet 2 of 3)				

Table 4.3 Memory Interface (Cont.)

Signal	I/O	BGA Pad No.	Pad Type	Description
MCLK	O	F5	3.3 V 8 mA T/S Output	**Memory Clock**. All synchronous RAM control/data signals are referenced to the rising edge of this clock. The exception is MOE/, which is typically an asynchronous input to SRAM and/or FLASH devices.
ADSC/	O	H1	3.3 V 4 mA T/S Output	**Address-Strobe-Controller**. Initiates Read, Write, or chip deselect cycle. When this signal is asserted, it also latches the memory address signals.
ADV/	O	H2	3.3 V 4 mA T/S Output	**Advance**. When asserted LOW, the ADV/ input causes a selected synchronous SRAM to increment its burst address counter.
BWE[3:0]/	O	E4, J5, J4, K6	3.3 V BiDir 4 mA	**Memory Byte Write Enables**. These active-LOW, byte lane write enables allow writing of partial words to memory.
RAMCS/	O	J2	3.3 V BiDir 4 mA	**RAM Chip Select**. This pin is an active-LOW synchronous chip select for all SSRAMs (up to four SSRAMs for interleaved and depth expanded configuration without additional decode logic).
(Sheet 3 of 3)				

1. MD[31:24] are used for the FLASH ROM Read/Write data.
2. “1” means the pin is pulled up on reset.
   “0” means the pin is pulled down on reset.

4.4 Configuration and Miscellaneous

Table 4.4 shows the Configuration and Miscellaneous signals.

Table 4.4 Configuration and Miscellaneous

Signal	I/O	BGA Pad No.	Pad Type	Description
GPIO[5:0]	I/O	AE12, AF11, AA9, Y10, AE11, AF10	3.3 V BiDir 8 mA with pull-up	**General Purpose I/O Pins**. These pins default to input mode on reset. These signals are controlled/observed by firmware. GPIO[1:0] are reserved for LSI Logic only. GPIO[5:2] are available as host-programmable outputs from IOC Unit Page 3.
LED[4:0]/	O	AE9, AD10, Y9, AA8, AF9	3.3 V BiDir 8 mA	**LED Outputs**. These output signals may be controlled by firmware or driven by chip activity. When configured as activity driven, the LED[n] outputs have the following meanings when asserted LOW: • LED[4]: Channel 1 – Fault (ON = no sync) • LED[3]: Channel 1 – Active (ON = frame traffic present) • LED[2]: Channel 0 – Fault (ON = no sync) • LED[1]: Channel 0 – Active (ON = frame traffic present) • LED[0]: Processor heartbeat
SERIAL_ CLOCK	O	AA4	3.3 V 4 mA BiDir with pull-up	**Serial EEPROM clock**.
SERIAL_ DATA	I/O	AB3	3.3 V 4 mA BiDir with pull-up	**Serial EEPROM data**.
MODE[7:0]	I	J7, H6, J1, K2, K1, L2, K7, J6	3.3 V TTL Input with pull-down	**Mode Select**. This 8-bit bus defines operational and test modes for the chip. Valid mode encodings are as follows: • Mode[7:0] = 001xxxxx — Interleaved BSRAM • Mode[7:0] = 000xxxxx — Noninterleaved BSRAM • Mode[7:0] = 00xx01xx — Soft Reset Mode0 • Mode[7:0] = 00xx11xx — Soft Reset Mode1[1] • Mode[7:0] = 00xxxx11 — Normal SEPROM Auto Load • Mode[7:0] = 00xxxx10 — Fast SEPROM Auto Load • Mode[7:0] = 00xxxx01 — Firmware PCI Configuration Mode[2] • Mode[7:0] = 00xxxx00 — PCI Configuration (use default values)

(Sheet 1 of 2)

Table 4.4 Configuration and Miscellaneous (Cont.)

Signal	I/O	BGA Pad No.	Pad Type	Description
TEST_ RESET/	I	V7	3.3 V Schmitt with pull-up	**Test Reset**. Forces the LSIFC949X into Power-On Reset state or Soft-Reset state, depending on the state of the Mode pins.
SCAN_ MODE	I	V5	3.3 V TTL Input with pull-down	**Scan Mode**. Reserved for LSI Logic test purposes only.
SCAN_ ENABLE	I	W6	3.3 V TTL Input with pull-down	**Scan Enable**. Reserved for LSI Logic test purposes only.
TDIODEVSS	I	U5		Reserved for LSI Logic test purposes only.
TDIODEP	I	U6	3.3 V Input with pull-down	Reserved for LSI Logic test purposes only.
FSELA	I	AC2	3.3 V TTL Input with pull-down	**ARM966/AHB Clock Select**. Reserved for LSI Logic test purposes only.
BZRSET		AE21		**Reference resistor node** for the PCI-X impedance controller. A 49.9 Ω ± 1% resistor should be tied between the BZRSET pad and the BZVDD pad.
BZVDD		AE22		**Reference resistor node** for the PCI-X impedance controller. A 49.9 Ω ± 1% resistor should be tied between the BZVDD pad and the BZRSET pad.
UARTRX	I	AB5	3.3 V In	**Receive for on-chip UART**
UARTTX	I/O	AC4	3.3 V BiDir	**Transmit for on-chip UART**
(Sheet 2 of 2)				

1. Soft Reset Mode1 also resets the Link Control logic in addition to the normal soft reset effects.
2. MA[17] must have a pull-up (for power-on sense purposes) to disable ARM booting.

4.5 Test and I/O Processor Debug

Table 4.5 shows the Test and I/O Processor Debug signals.

Table 4.5 Test and I/O Processor Debug

Signal	I/O	BGA Pad No.	Pad Type	Description
TCK	I	AA1	3.3 V Schmitt with pull-up	JTAG/CtxMgr Debug **Test Clock**.
TRST/	I	AB1	3.3 V Schmitt with pull-up	JTAG/Debug **Test Reset**. Asynchronous active LOW.
TDI	I	AA3	3.3 V Schmitt with pull-up	JTAG/CtxMgr Debug **Test Data In**.
TDO	B	AB4	3.3 V 4 mA T/S Output with pull-up	JTAG/CtxMgr Debug **Test Data Out**.
TMS	I	AA2	3.3 V Schmitt with pull-up	JTAG **Test Mode Select**.
TMS_ICE	I	Y2	3.3 V Schmitt with pull-up	CtxMgr Debug **Test Mode Select**.
TDI_ICE	I	Y6	3.3 V Schmitt with pull-up	Multi-ICE Debug **Test Data In.**
TDO_ICE	B	W7	3.3 V 4 mA T/S Output with pull-up	Multi-ICE Debug **Test Data Out.**
TCK_ICE	I	W2	3.3 V Schmitt with pull-up	Multi-ICE Debug **Clock.**
RTCK_ICE	B	W1	3.3 V 4 mA T/S Output with pull-up	Multi-ICE Debug **RClk.**
TRST_ICE/	I	Y1	3.3 V Schmitt with pull-up	**Multi-ICE Debug Reset.** Asynchronous active LOW. When ICE is not used, pull TRST_ICE/ low through a 220 Ω resistor.
PROC_MON	O	AA5		**Process Monitor Test Output Driver**. Reserved for LSI Logic test purposes only.
TN	I	AC1		**3-State Output Enable Control.** Reserved for LSI Logic test purposes only.
IDDTN	I	AB2	In	**IDDTN Test Pad.** Reserved for LSI Logic test purposes only.

4.6 Power and Ground

Table 4.6 shows the Power and Ground signals.

Table 4.6 Power and Ground

Signal	BGA Pad No.	Description	Voltage
VDD12[1]	See Table 7.14 and Table 7.15 on pages 7-10 and 7-12, respectively.	Core power.	1.2 V
VSS	See Table 7.14 and Table 7.15 on pages 7-10 and 7-12, respectively.	Ground.	0 V
VDDIO33	See Table 7.14 and Table 7.15 on pages 7-10 and 7-12, respectively.	I/O power.	3.3 V
VDDIO33_ PCIX	See Table 7.14 and Table 7.15 on pages 7-10 and 7-12, respectively.	PCIX power.	3.3 V
REFPLLVDD	AD2	Analog power for PCI FSN cell.	1.2 V
REFPLLVSS	AC3	Analog ground for PCI FSN cell.	0 V
PCIPLLVDD	AE24	Analog power for ARM clock generation.	1.2 V
PCIPLLVSS	AD23	Analog ground for ARM clock generation.	0 V
MXSVDD	L3, M3, R4, T4	Analog power for integrated transceiver core.	1.2 V
RXBVDD0	P1	Analog power for integrated transceiver core.	1.2 V
RXBVSS0	N1	Analog ground for integrated transceiver core.	0 V
RXBVDD1	U2	Analog power for integrated transceiver core.	1.2 V
RXBVSS1	U1	Analog ground for integrated transceiver core.	0 V
RXVDD0	R1	Analog power for integrated transceiver core.	1.2 V
RXVSS0	P2	Analog ground for integrated transceiver core.	0 V
RXVDD1	V1	Analog power for integrated transceiver core.	1.2 V
RXVSS1	U3	Analog ground for integrated transceiver core.	0 V

(Sheet 1 of 2)

Table 4.6 Power and Ground (Cont.)

Signal	BGA Pad No.	Description	Voltage
TXBVDD0	N3	Analog power for integrated transceiver core.	1.2 V
TXBVSS0	M2	Analog ground for integrated transceiver core.	0 V
TXBVDD1	R2	Analog power for integrated transceiver core.	1.2 V
TXBVSS1	P3	Analog ground for integrated transceiver core.	0 V
TXVDD0	M1	Analog power for integrated transceiver core.	1.2 V
TXVSS0	N2	Analog ground for integrated transceiver core.	0 V
TXVDD1	T2	Analog power for integrated transceiver core.	1.2 V
TXVSS1	T1	Analog ground for integrated transceiver core.	0 V
(Sheet 2 of 2)			

1. The required core voltage on the LSIFC949X is 1.2 V. The PCIX I/O voltage requires 3.3 V, and the GigaBlaze Fibre Channel transceiver interface requires 1.2 V. Configure the power supply to the chip so that the lower voltages power-up in advance of the higher voltages. The recommended power sequencing depends on the number of supplies used. For a PCIX system with PCIX buffers, the recommended power sequence is 1.2 V and then 3.3 V; or make certain that the following conditions are met during the power cycling:
   (VDD1.2 > 1 V) before (VDD3.3 > 1 V)

Chapter 5
PCI-X Functional Description

This chapter provides a general description of the PCI-X features contained in the LSIFC949X Dual Channel Fibre Channel I/O processor chip. The chapter contains the following sections:

5.1 Overview

The host PCI-X interface complies with the *PCI Local Bus Specification,* Revision 2.3, and the *PCI-X Addendum to the PCI Local Bus Specification,* Revision 2.0. The LSIFC949X supports up to a 133 MHz, 64-bit PCI-X bus. The LSIFC949X supports 64-bit addressing with Dual Address Cycle (DAC).

The LSIFC949X is a true multifunction PCI-X device that presents a single electrical load to the PCI-X bus. The LSIFC949X uses a single REQ/-GNT/ pair to arbitrate for PCI-X bus mastership. Separate interrupt signals for PCI Function [0] and PCI Function [1] allow independent control of the two PCI functions.

5.2 PCI-X Addressing

The three physical address spaces the PCI specification defines are:

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

The following sections describe the PCI address spaces.

5.2.1 PCI Configuration Space

The LSIFC949X defines an independent set of PCI Configuration Space registers for each PCI function. Each configuration space is a contiguous, 256-x-8-bit set of addresses. The system BIOS initializes the configuration registers using PCI-X configuration cycles. The LSIFC949X decodes the C_BE[3:0]/ field to determine whether a PCI-X cycle intends to access the configuration register space. The IDSEL signal behaves as a chip select signal that enables access to the configuration register space only. The LSIFC949X ignores configuration read/write cycles when IDSEL is not asserted.

Because the LSIFC949X is a multifunction PCI-X device, bits AD[10:8] decode either the PCI Function [0] Configuration Space (AD[10:8] = 0b000) or the PCI Function [1] Configuration Space (AD[10:8] = 0b001). The LSIFC949X does not respond to any other encodings of AD[10:8]. Bits AD[7:2] select one of the 64 Dword registers in the LSIFC949X PCI Configuration Space. Bits AD[1:0] determine whether the configuration command is a Type 0 Configuration Command (AD[1:0] = 0b00) or a Type 1 Configuration Command (AD[1:0] = 0b01). Because the LSIFC949X is not a PCI Bridge device, all PCI Configuration Commands designated for the LSIFC949X must be Type 0. Bits C_BE[3:0]/ address the individual bytes within each Dword and determine the type of access to perform.

5.2.2 PCI I/O Space

The PCI specification defines I/O space as a contiguous 32-bit, I/O address that all system resources share, including the LSIFC949X. The

I/O Base Address register determines the 256-byte PCI I/O area that the PCI device occupies.

5.2.3 PCI Memory Space

The LSIFC949X contains two PCI memory spaces: PCI Memory Space [0] and PCI Memory Space [1]. PCI Memory Space [0] supports normal memory accesses, while PCI Memory Space [1] supports diagnostic memory accesses. The LSIFC949X requires 64 Kbytes of memory space.

The PCI specification defines memory space as a contiguous, 64-bit memory address that all system resources share. The Memory [0] Base Address Low and Memory [0] Base Address High registers determine which 64 Kbyte memory area PCI Memory Space [0] occupies. The Memory [1] Base Address Low and Memory [1] Base Address High registers determine which 64 Kbyte memory area PCI Memory Space [1] occupies.

5.3 PCI/PCI-X Bus Commands and Implementation

Bus commands indicate to the target the type of transaction the master is requesting. The master encodes the bus commands on the C_BE[3:0]/ lines during the address phase. The PCI/PCI-X bus commands and their encodings appear in Table 5.1.

Table 5.1 PCI/PCI-X Bus Commands and Encodings[1]

C_BE[3:0]/	PCI Bus Command	PCI-X Bus Command	Supports as Master	Supports as Slave
0b0000	Interrupt Acknowledge	Interrupt Acknowledge	No	No
0b0001	Special Cycle	Special Cycle	No	No
0b0010	I/O Read	I/O Read	Yes	Yes
0b0011	I/O Write	I/O Write	Yes	Yes
0b0100	Reserved	Reserved	N/A	N/A
0b0101	Reserved	Reserved	N/A	N/A
0b0110	Memory Read	Memory Read Dword	Yes	Yes

(Sheet 1 of 2)

Table 5.1 PCI/PCI-X Bus Commands and Encodings[1] (Cont.) (Cont.)

C_BE[3:0]/	PCI Bus Command	PCI-X Bus Command	Supports as Master	Supports as Slave
0b0111	Memory Write	Memory Write	Yes	Yes
0b1000	Reserved	Alias to Memory Read Block	PCI: N/A PCI-X: No	PCI: N/A PCI-X: Yes
0b1001	Reserved	Alias to Memory Write Block	PCI: N/A PCI-X: No	PCI: N/A PCI-X: Yes
0b1010	Configuration Read	Configuration Read	No	Yes
0b1011	Configuration Write	Configuration Write	No	Yes
0b1100	Memory Read Multiple	Split Completion	Yes	Yes[2]
0b1101	Dual Address Cycles (DAC)	Dual Address Cycles (DAC)	Yes	Yes
0b1110	Memory Read Line	Memory Read Block	Yes	Yes[2]
0b1111	Memory Write and Invalidate	Memory Write Block	Yes	Yes[3]

(Sheet 2 of 2)

1. The LSIFC949X ignores reserved commands as a slave and never generates them as a master.
2. When acting as a slave in the PCI mode, the LSIFC949X supports this command as the PCI Memory Read command.
3. When acting as a slave in the PCI mode, the LSIFC949X supports this command as the PCI Memory Write command.

The following sections describe how the LSIFC949X implements these commands.

5.3.1 Interrupt Acknowledge Command

The LSIFC949X ignores this command as a slave and never generates it as a master.

5.3.2 Special Cycle Command

The LSIFC949X ignores this command as a slave and never generates it as a master.

5.3.3 I/O Read Command

The I/O Read command reads data from an agent mapped in the I/O address space. When decoding I/O commands, the LSIFC949X decodes the lower 32 address bits and ignores the upper 32 address bits. The

LSIFC949X supports this command when operating in either the PCI or PCI-X bus mode.

5.3.4 I/O Write Command

The I/O Write command writes data to an agent mapped in the I/O address space. When decoding I/O commands, the LSIFC949X decodes the lower 32 address bits and ignores the upper 32 address bits. The LSIFC949X supports this command when operating in either the PCI or PCI-X bus mode.

5.3.5 Memory Read Command

The LSIFC949X uses the Memory Read command to read data from an agent mapped in the memory address space. The target can perform an anticipatory read if such a read produces no side effects. The LSIFC949X supports this command when operating in the PCI bus mode.

5.3.6 Memory Read Dword Command

The Memory Read Dword command reads up to a single Dword of data from an agent mapped in the memory address space and can only be initiated as a 32-bit transaction. The target can perform an anticipatory read if such a read produces no side effects. The LSIFC949X supports this command when operating in the PCI-X bus mode.

5.3.7 Memory Write Command

The Memory Write command writes data to an agent mapped in the memory address space. The target assumes responsibility for data coherency when it returns “ready”. The LSIFC949X supports this command when operating in either the PCI or PCI-X bus mode.

5.3.8 Alias to Memory Read Block Command

This command is reserved for future implementations of the PCI specification. The LSIFC949X never generates this command as a master. When a slave, the LSIFC949X supports this command using the Memory Read Block command.

5.3.9 Alias to Memory Write Block Command

This command is reserved for future implementations of the PCI specification. The LSIFC949X never generates this command as a master. When a slave, the LSIFC949X supports this command using the Memory Write Block command.

5.3.10 Configuration Read Command

The Configuration Read command reads the configuration space of a device. The LSIFC949X never generates this command as a master, but does respond to it as a slave. A device on the PCI bus selects the LSIFC949X by asserting its IDSEL signal when bits AD[1:0] = 0b00. During the address phase of a configuration cycle, bits AD[7:2] address one of the 64 Dword registers in the configuration space of each device. C_BE[3:0]/ address the individual bytes within each Dword register and determine the type of access to perform. Bits AD[10:8] address either the PCI Function [0] Configuration Space (AD[10:8] = 0b000) or the PCI Function [1] Configuration Space (AD[10:8] = 0b001). The LSIFC949X treats AD[63:11] as logical don't cares.

5.3.11 Configuration Write Command

The Configuration Write command writes the configuration space of a device. The LSIFC949X never generates this command as a master, but does respond to it as a slave. A device on the PCI bus selects the LSIFC949X by asserting its IDSEL signal when bits AD[1:0] = 0b00. During the address phase of a configuration cycle, bits AD[7:2] address one of the 64 Dword registers in the configuration space of each device. C_BE[3:0]/ address the individual bytes within each Dword register and determine the type of access to perform. Bits AD[10:8] decode either the PCI Function [0] Configuration Space (AD[10:8] = 0b000) or the PCI Function [1] Configuration Space (AD[10:8] = 0b001). The LSIFC949X treats AD[63:11] as logical don't cares.

5.3.12 Memory Read Multiple Command

The Memory Read Multiple command is identical to the Memory Read command, except it additionally indicates that the master intends to fetch multiple cache lines before disconnecting. The LSIFC949X supports PCI Memory Read Multiple functionality when operating in the PCI mode

and determines when to issue a Memory Read Multiple command instead of a Memory Read command.

Burst Size Selection – The Memory Read Multiple command reads multiple cache lines of data during a single bus ownership. The number of cache lines the LSIFC949X reads is a multiple of the cache line size, which the *PCI Local Bus Specification,* Revision 2.3, provides. The LSIFC949X selects the largest multiple of the cache line size based on the amount of data to transfer.

5.3.13 Split Completion Command

Split transactions in PCI-X replace the delayed transactions in conventional PCI. The LSIFC949X supports one outstanding split transaction when operating in the PCI-X mode. A split transaction consists of at least two separate bus transactions: a split request, which the requester initiates; and one or more split completion commands, which the completer initiates. The *PCI-X Addendum to the PCI Local Bus Specification,* Revision 2.0, permits split transaction completion for the Memory Read Block, Alias to Memory Read Block, Memory Read Dword, Interrupt Acknowledge, I/O Read, I/O Write, Configuration Read, and Configuration Write commands. When operating in the PCI-X mode, the LSIFC949X supports the Split Completion command for all of these commands except the Interrupt Acknowledge command, which the LSIFC949X neither responds to nor generates.

5.3.14 Dual Address Cycles (DAC) Command

The LSIFC949X performs Dual Address Cycles (DAC), according to the *PCI Local Bus Specification,* Revision 2.3. The LSIFC949X supports this command when operating in either the PCI or PCI-X bus mode.

5.3.15 Memory Read Line Command

This command is identical to the Memory Read command except it additionally indicates that the master intends to fetch a complete cache line. The LSIFC949X supports this command when operating in the PCI mode.

5.3.16 Memory Read Block Command

The LSIFC949X uses this command to read from memory. The LSIFC949X supports this command when operating in the PCI-X mode.

5.3.17 Memory Write and Invalidate Command

The Memory Write and Invalidate command is identical to the Memory Write command, except it additionally guarantees a minimum transfer of one complete cache line. The master uses this command when it 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. The LSIFC949X determines when to issue a Write and Invalidate command instead of a Memory Write command, and supports this command when operating in the PCI bus mode.

5.3.17.1 Alignment

The LSIFC949X uses the calculated line size value to determine whether the current address aligns to the cache line size. If the address does not align, the LSIFC949X bursts data using a noncache command. If the starting address aligns, the LSIFC949X issues a Memory Write and Invalidate command using the cache line size as the burst size.

5.3.17.2 Multiple Cache Line Transfers

The Memory Write and Invalidate command can write multiple cache lines of data in a single bus ownership. The LSIFC949X issues a burst transfer as soon as it reaches a cache line boundary. The PCI Local Bus specification states that the transfer size must be a multiple of the cache line size. The LSIFC949X selects the largest multiple of the cache line size based on the transfer size. When the DMA buffer contains less data than the value Cache Line Size register specifies, the LSIFC949X issues a Memory Write command on the next cache boundary to complete the data transfer.

5.3.18 Memory Write Block Command

The LSIFC949X uses this command to burst data to memory. The LSIFC949X supports this command when operating in the PCI-X bus mode.

5.4 PCI Arbitration

The LSIFC949X contains independent bus mastering functions for each of the SCSI functions and for the system interface. The system interface bus mastering function manages DMA operations as well as the request and reply message frames. The SCSI channel bus mastering functions manage data transfers across the SCSI channels.

The LSIFC949X uses a single REQ/-GNT/ signal pair to arbitrate for access to the PCI bus. To ensure fair access to the PCI bus, the internal arbiter uses a round robin arbitration scheme to decide which of the three internal bus mastering functions can arbitrate for access to the PCI bus.

5.5 PCI Cache Mode

The LSIFC949X supports an 8-bit, Cache Line Size register. This register provides the ability to sense and react to nonaligned addresses corresponding to cache line boundaries. The LSIFC949X determines when to issue a PCI cache command (Memory Read Line, Memory Read Multiple, and Memory Write and Invalidate) or PCI noncache command (Memory Read or Memory Write).

Chapter 6
Registers

This chapter describes the PCI host register space. The chapter consists of the following sections:

- Section 6.1, “PCI-X Configuration Space Register Description”
- Section 6.2, “PCI I/O Space and Memory Space Register Description”

The register map at the beginning of each register description provides the default bit settings for the register. Shading indicates a reserved bit or register. Do not access the reserved address areas.

There are two PCI functions on the LSIFC949X. Each PCI function has its own independent interrupt pin and its own PCI Address space. The PCI System Address space consists of three regions: PCI Configuration Space, PCI Memory Space, and PCI I/O Space. PCI Configuration Space supports the identification, configuration, initialization, and error management functions for the LSIFC949X PCI devices. PCI Memory Space [0] and PCI Memory Space [1] form PCI Memory Space. PCI Memory Space [1] provides diagnostic memory accesses. PCI I/O Space and PCI Memory Space [0] provide normal system access to memory.

6.1 PCI-X Configuration Space Register Description

This section provides bit-level descriptions of the PCI Configuration Space registers. Table 6.1 defines the PCI Configuration Space registers. A separate set of PCI Configuration Space registers exists for each PCI function.

The LSIFC949X enables, orders, and locates the PCI-extended capability register structures (Power Management, Messaged Signaled Interrupts, and PCI-X) to optimize device performance. The LSIFC949X does not hardcode the location and order of the PCI-extended capability

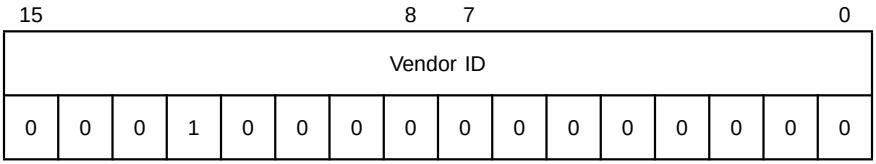
structures. The address and location of the PCI-extended capability structures are subject to change. To access a PCI-extended capability structure, follow the pointers held in the Capability Pointer registers and identify the extended capability structure with the Capability ID register for the given structure.

Shading in the following address map and register descriptions indicates reserved bits.

Table 6.1 LSIFC949X PCI-X Configuration Space Address Map

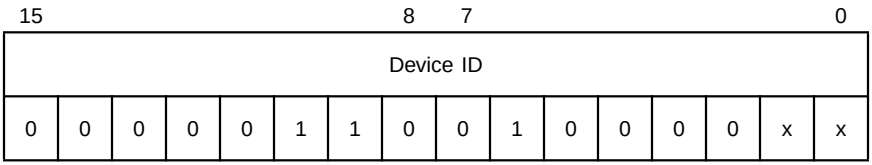
31	24	23	16	15	8	7	0	Offset	Page
Device ID				Vendor ID				0x00	6-3
Status				Command				0x04	6-4
Class Code						Revision ID		0x08	6-8
Reserved		Header Type		Latency Timer		Cache Line Size		0x0C	6-9
I/O Base Address								0x10	6-10
Memory [0] Base Address Low								0x14	6-11
Memory [0] Base Address High								0x18	6-11
Memory [1] Base Address Low								0x1C	6-12
Memory [1] Base Address High								0x20	6-12
Reserved								0x24	–
								0x28	–
Subsystem ID				Subsystem Vendor ID				0x2C	6-13
Expansion ROM Base Address								0x30	6-14
Reserved						Capabilities Pointer		0x34	6-15
								0x38	–
Maximum Latency		Minimum Grant		Interrupt Pin		Interrupt Line		0x3C	6-16
Reserved								0x40–0xFF	–
Power Management Capabilities				PM Next Pointer		PM Capability ID			6-19
PM Data		PM BSE		Power Management Control/Status					6-20
Reserved									–
MSI Message Control				MSI Next Pointer		MSI Capability ID			6-22
MSI Message Lower Address									6-23
MSI Message Upper Address									6-24
Reserved				MSI Message Data					6-24
MSI Mask Bits									6-25
MSI Pending Bits									6-25
Reserved									–
MSI-X Message Control				MSI-X Next Pointer		MSI-X Capability ID			6-25
MSI-X Table Offset									6-27
MSI-X PBA Offset									6-27
Reserved									–
PCI-X Command				PCI-X Next Pointer		PCI-X Capability ID			6-28
PCI-X Status									6-30
Reserved									–

Register: 0x00–0x01
Vendor ID
Read Only



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

Register: 0x02–0x03
Device ID
Read Only



Device ID **[15:0]**
This register identifies the particular device. The most significant 12 bits are hardcoded to a constant of 0x064. The LSB is dependent upon the power-on sense functions corresponding to the states of pins MA[4] and MA[3] as decoded in Table 6.2.

Table 6.2 Device ID Values

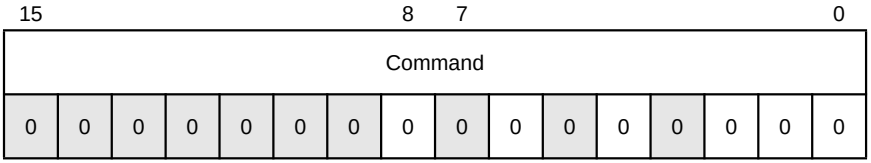
Single/Dual Channel	State of MA[4:3]	Device ID
Function 1		
Dual Channel	MA[4] = 0	0x0640
Dual Channel	MA[4] = 1	0x0641
Single Channel	MA[4] = 0	0x0642
Single Channel	MA[4] = 1	0x0643
(Sheet 1 of 2)		

Table 6.2 Device ID Values (Cont.)

Single/Dual Channel	State of MA[4:3]	Device ID
Function 0		
Dual Channel	MA[3] = 0	0x0640
Dual Channel	MA[3] = 1	0x0641
Single Channel	MA[3] = 0	0x0642
Single Channel	MA[3] = 1	0x0643
(Sheet 2 of 2)		

Register: 0x04–0x05

Command
Read/Write



This register provides coarse control over how the PCI function generates and responds to PCI cycles. Writing a zero to this register logically disconnects the LSIFC949X PCI function from the PCI bus for all accesses except configuration accesses.

Reserved **[15:9]**
This field is reserved.

SERR/ Enable **8**
Setting this bit enables the LSIFC949X to activate the SERR/ driver. Clearing this bit disables the SERR/ driver.

Reserved **7**
This bit is reserved.

Enable Parity Error Response **6**
Setting this bit enables the LSIFC949X PCI function to detect parity errors on the PCI bus and report these errors to the system. Clearing this bit causes the LSIFC949X PCI function to set the Detected Parity Error bit (bit 15 in the Status register (register 0x06–0x07)) but not assert the PERR/ signal when the PCI function

detects a parity error. This bit only affects parity checking. The PCI function always generates parity for the PCI bus.

Reserved **5**
This bit is reserved.

Write and Invalidate Enable **4**
Setting this bit enables the PCI function to generate write and invalidate commands on the PCI bus when operating in the conventional PCI mode.

Reserved **3**
This bit is reserved.

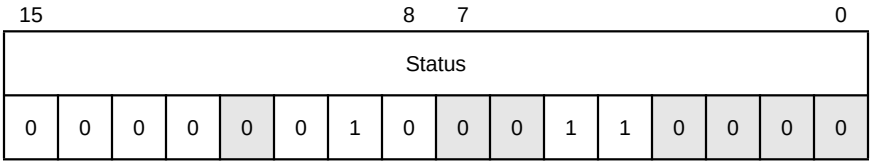
Enable Bus Mastering **2**
Setting this bit allows the PCI function to behave as a PCI bus master. Clearing this bit disables the PCI function from generating PCI bus master accesses.

Enable Memory Space **1**
This bit controls the ability of the PCI function to respond to Memory Space accesses. Setting this bit allows the LSIFC949X to respond to Memory Space accesses at the address range specified by the Memory [0] Base Address Low, Memory [0] Base Address High, Memory [1] Base Address Low, Memory [1] Base Address High, and the Expansion ROM Base Address registers. Clearing this bit disables the PCI function response to memory space accesses.

Enable I/O Space **0**
This bit controls the LSIFC949X PCI function response to I/O space accesses. Setting this bit enables the PCI function to respond to I/O Space accesses at the address range the PCI Configuration Space I/O Base Address register specifies. Clearing this bit disables the PCI function response to I/O space accesses.

Register: 0x06–0x07

Status
Read/Write



Reads to this register behave normally. To clear a bit location that is currently set, write the bit to one (1). For example, to clear bit 15 when it is set, and not affect any other bits, write 0x8000 to the register.

Detected Parity Error (from Slave) **15**
This bit is set according to the *PCI Local Bus Specification,* Revision 2.3, and the *PCI-X Addendum to the PCI Local Bus Specification,* Revision 2.0.

Signaled System Error **14**
The LSIFC949X PCI function sets this bit when asserting the SERR/ signal.

Received Master Abort (from Master) **13**
A master device sets this bit when a Master Abort command terminates its transaction (except for Special Cycle).

Received Target Abort (from Master) **12**
A master device sets this bit when a Target Abort command terminates its transaction.

Reserved **11**
This bit is reserved.

DEVSEL/ Timing **[10:9]**
These two read-only bits encode the timing of the DEVSEL/ signal and indicate the slowest time that a device asserts the DEVSEL/ signal for any bus command except Configuration Read and Configuration Write. The

LSIFC949X only supports medium DEVSEL/ timing. The possible timing values are as follows:

0b00	Fast
0b01	Medium
0b10	Slow
0b11	Reserved

Data Parity Error Reported **8**
This bit is set according to the*PCI Local Bus Specification,* Revision 2.3, and the *PCI-X Addendum to the PCI Local Bus Specification,* Revision 2.0. Refer to bit 0 of the PCI-X Command register for details.

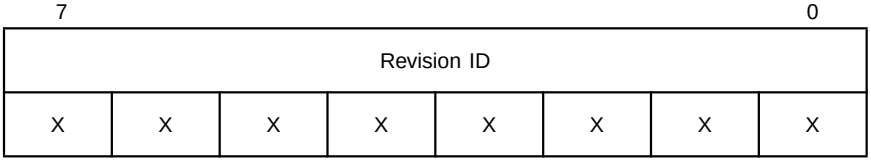
Reserved **[7:6]**
This field is reserved.

66 MHz Capable **5**
The MA[10] Power-On Sense pin controls this bit. Allowing the internal pull-down to pull MA[10] LOW sets this bit and indicates to the host system that the LSIFC949X PCI function is capable of operating at 66 MHz. Pulling MA[10] HIGH clears this bit and indicates to the host system that the LSIFC949X PCI function is not capable of operating at 66 MHz. Refer to Table 4.3 on page 4-9 for details.

New Capabilities **4**
The LSIFC949X PCI function sets this read-only bit to indicate a list of PCI extended capabilities such as PCI Power Management, Message Signaled Interrupt (MSI), and PCI-X support.

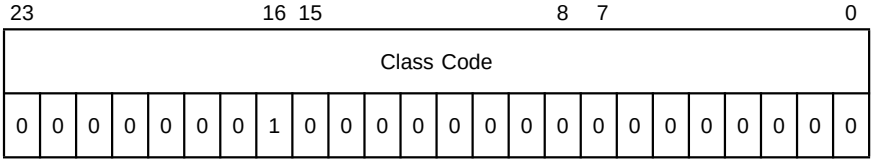
Reserved **[3:0]**
This field is reserved.

Register: 0x08
Revision ID
Read/Write



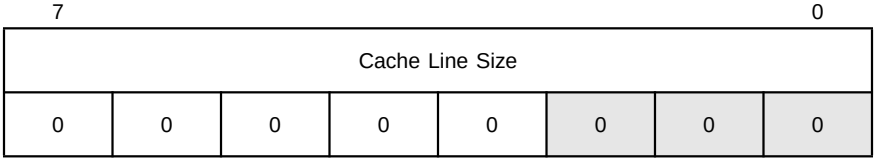
Revision ID **[7:0]**
This register indicates the current revision level of the device.

Register: 0x09–0x0B
Class Code
Read Only



Class Code **[23:0]**
This 24-bit register identifies the generic function of this 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 0x0C0400, and is written by the SEEPROM (provided the SEEPROM is present in the system). If no SEEPROM is present in the system, the default Class Code is 0x010000.

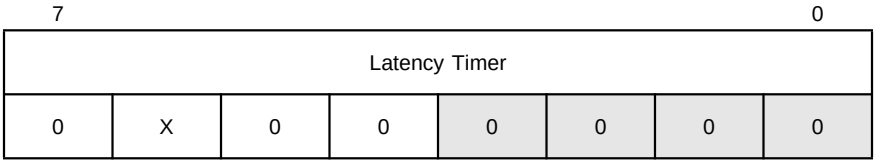
Register: 0x0C
Cache Line Size
Read/Write



Cache Line Size **[7:3]**
This register specifies the system cache line size in units of 32-bit words. In the conventional PCI mode, the LSIFC949X PCI function uses this register to determine whether to use Write and Invalidate or Write commands for performing write cycles. Programming this register to a number other than a nonzero power of two disables the the use of the PCI performance commands to execute data transfers. The LSIFC949X PCI function ignores this register when operating in the PCI-X mode.

Reserved **[2:0]**
This field is reserved.

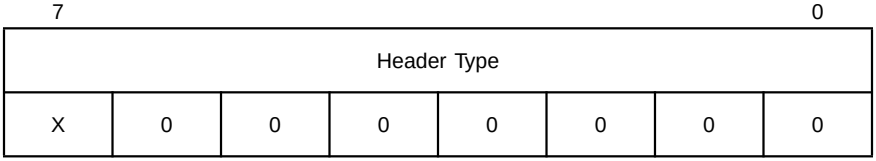
Register: 0x0D
Latency Timer
Read/Write



Latency Timer **[7:4]**
This register specifies, in units of PCI bus clocks, the value of the Latency Timer for this PCI bus master.

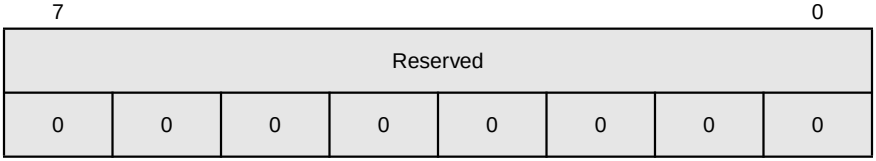
Reserved **[3:0]**
This field is reserved.

Register: 0x0E
Header Type
Read Only



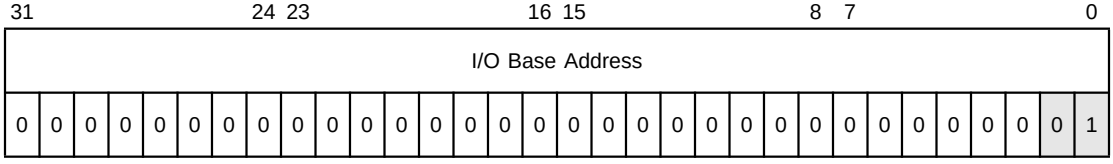
Header Type **[7:0]**
This 8-bit register identifies the layout of bytes 0x10 through 0x3F in configuration space and also indicates whether this device is a single function or multifunction PCI device.

Register: 0x0F
Reserved



Reserved **[7:0]**
This register is reserved.

Register: 0x10–0x13
I/O Base Address
Read/Write

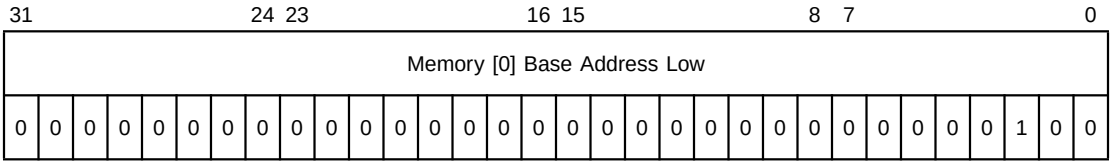


This register maps the operating register set into I/O space. The LSIFC949X requires 256 bytes of I/O space for this register. Hardware sets bit 0 to 0b1. Bit 1 is reserved and returns 0b0 on all reads.

I/O Base Address **[31:2]**
This field contains the I/O Base Address.

Reserved **[1:0]**
This field is reserved.

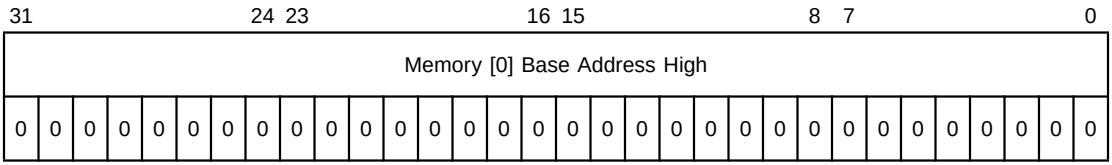
Register: 0x14–0x17
Memory [0] Base Address Low
Read/Write



The Memory [0] Base Address Low register and the Memory [0] Base Address High register map SCSI operating registers into Memory Space [0]. The Memory [0] Base Address Low register contains the lower 32 bits of the Memory Space [0] base address. Hardware programs bits [9:0] to 0b0000000100, which indicates that the Memory Space [0] base address is 64 bits wide and that the memory data is not prefetchable. The LSIFC949X requires 1024 bytes of memory space.

Memory [0] Base Address Low **[31:0]**
This field contains the Memory [0] Base Address Low address.

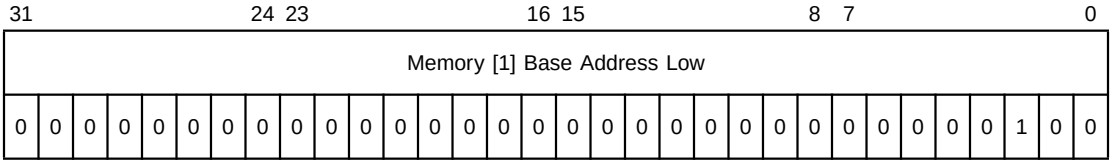
Register: 0x18–0x1B
Memory [0] Base Address High
Read/Write



The Memory [0] Base Address High register and the Memory [0] Base Address Low register map SCSI operating registers into Memory Space [0]. The Memory [0] Base Address High register contains the upper 32 bits of the Memory Space [0] base address. The LSIFC949X requires 1024 bytes of memory space.

Memory [0] Base Address High **[31:0]**
This field contains the Memory [0] Base Address High address.

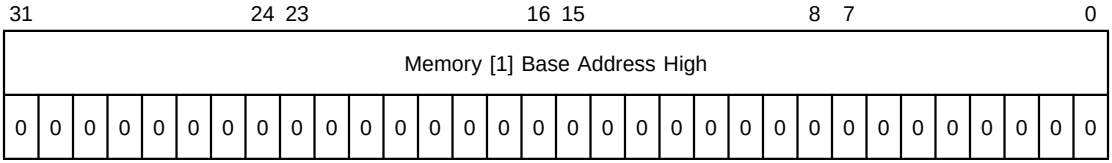
Register: 0x1C–0x1F
Memory [1] Base Address Low
Read/Write



The Memory [1] Base Address Low register and the Memory [1] Base Address High register map the RAM into Memory Space [1]. The Memory [1] Base Address Low register contains the lower 32 bits of the Memory Space [1] base address. Hardware programs bits [12:0] to 0b0000000000100, which indicates that the Memory Space [1] base address is 64 bits wide and that the memory data is not prefetchable. The LSIFC949X requires 64 Kbytes of memory for Memory Space [1].

Memory [1] Base Address Low **[31:0]**
This field contains the Memory [1] Base Address Low address.

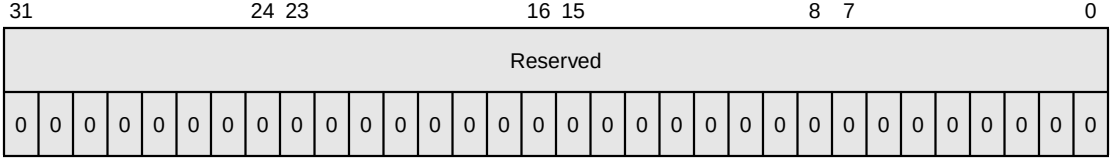
Register: 0x20–0x23
Memory [1] Base Address High
Read/Write



The Memory [1] Base Address Low register and the Memory [1] Base Address High register map the RAM into Memory Space [1]. The Memory [1] Base Address Low register contains the upper 32 bits of the Memory Space [1] base address. The LSIFC949X requires 64 Kbytes of memory for Memory Space [1].

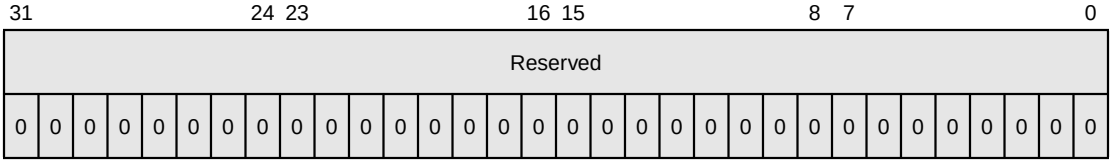
Memory [1] Base Address High **[31:0]**
This field contains the Memory [1] Base Address High address.

Register: 0x24–0x27
Reserved



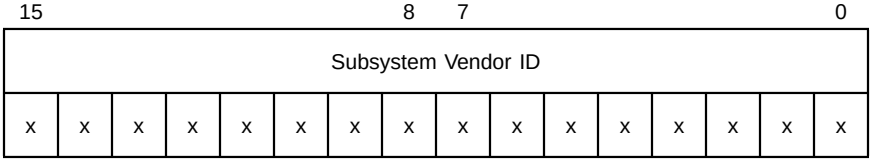
Reserved **[31:0]**
This register is reserved.

Register: 0x28–0x2B
Reserved



Reserved **[31:0]**
This register is reserved.

Register: 0x2C–0x2D
Subsystem Vendor ID
Read Only



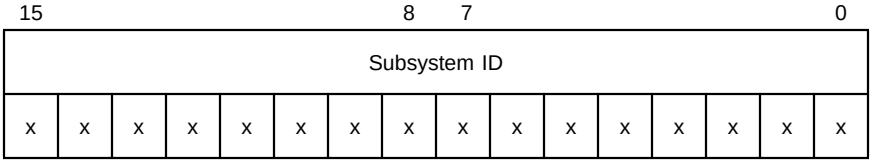
SVID **Subsystem Vendor ID** **[15:0]**
This 16-bit register uniquely identifies the vendor that manufactures the add-in board or subsystem where the LSIFC949X resides. This register provides a mechanism for an add-in card vendor to distinguish their cards from

those of another vendor, even if the cards use the same PCI controller (and have the same Vendor ID and Device ID).

The external serial EEPROM can hold a vendor-specific, 16-bit value for this register, which the board designer must obtain from the PCI Special Interest Group (PCI-SIG).

Register: 0x2E–0x2F

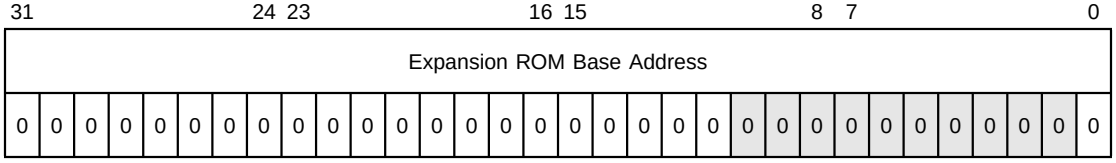
Subsystem ID
Read Only



SID **Subsystem ID** **[15:0]**
This 16-bit register uniquely identifies the add-in board or subsystem where this PCI device resides. This register provides a mechanism for an add-in card vendor to distinguish their cards from one another even if the cards use the same PCI controller (and have the same Vendor ID and Device ID). The board designer can store a vendor-specific, 16-bit value in an external serial EEPROM.

Register: 0x30–0x33

Expansion ROM Base Address
Read/Write



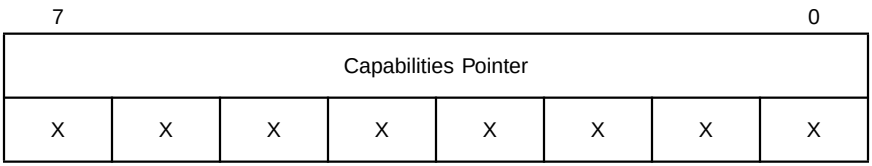
This 32-bit register contains the base address and size information for the expansion ROM.

Expansion ROM Base Address **[31:11]**
These bits correspond to the upper 21 bits of the expansion ROM base address. The host system detects the size of the external memory by first writing 0xFFFFFFFF to this register and then reading the register back. The LSIFC949X responds with zeros in all don't care locations. The least significant one (1) that remains represents the binary version of the external memory size. For example, to indicate an external memory size of 32 Kbytes, this register returns ones in the upper 17 bits when written with 0xFFFFFFFF and read back.

Reserved **[10:1]**
This field is reserved.

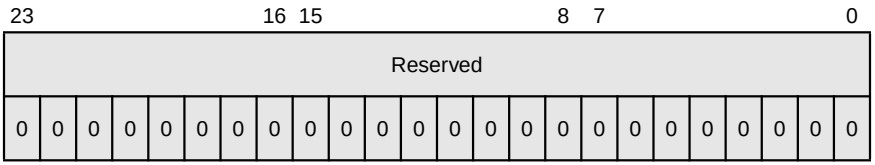
Expansion ROM Enable **0**
This bit controls whether the device accepts accesses to its expansion ROM. Setting this bit enables address decoding. Depending on the system configuration, the device can optionally use an expansion ROM. Note that to access the expansion ROM, the user must also set bit 1 in the PCI Command register.

Register: 0x34
Capabilities Pointer
Read Only



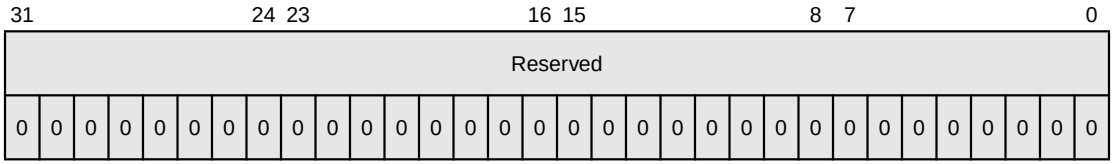
Capabilities Pointer **[7:0]**
This register indicates the location of the first extended capabilities register in PCI Configuration Space. The value of this register varies according to system configuration.

Register: 0x35–0x37
Reserved



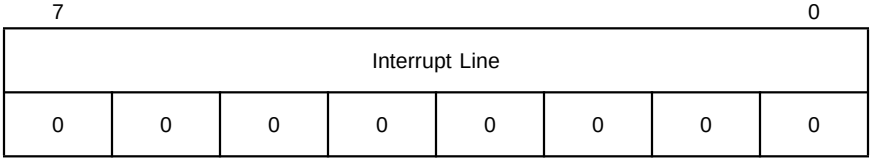
Reserved **[23:0]**
This register is reserved.

Register: 0x38–0x3B
Reserved



Reserved **[31:0]**
This register is reserved.

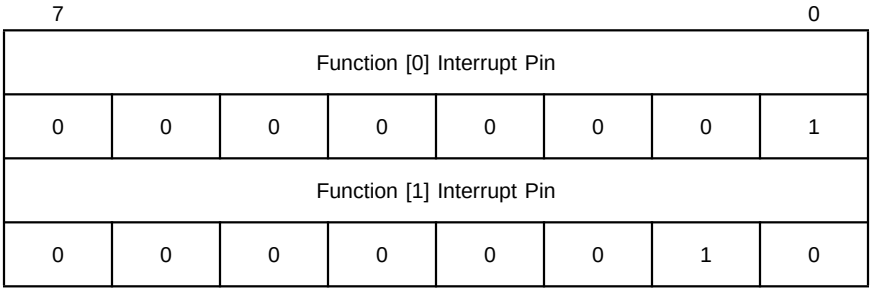
Register: 0x3C
Interrupt Line
Read/Write



Interrupt Line **[7:0]**
This register communicates interrupt line routing information. Power-On-Self-Test (POST) software writes the routing information into this register as it configures the system. This register indicates the system interrupt controller input to which this PCI function interrupt pin connects. System architecture determines the values in this register.

Register: 0x3D

Interrupt Pin
Read Only



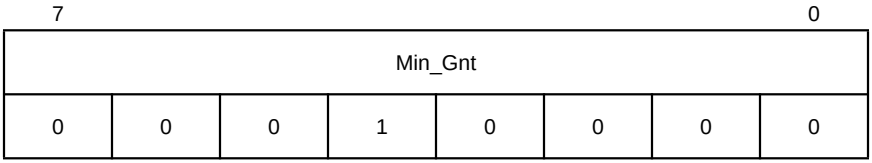
7							0
Function [0] Interrupt Pin							
0	0	0	0	0	0	0	1
Function [1] Interrupt Pin							
0	0	0	0	0	0	1	0

Interrupt Pin **[7:0]**

The encoding of this read-only register is unique to each function on the LSIFC949X. It indicates which interrupt pin the function uses. The value for Function [0] is 0x01, which indicates that Function [0] presents interrupts on the INTA/ pin. The value for Function [1] is 0x02, which indicates that Function [1] presents interrupts on the INTB/ pin.

Register: 0x3E

Minimum Grant
Read Only



7							0
Min_Gnt							
0	0	0	1	0	0	0	0

Min_Gnt **[7:0]**

This register specifies the desired settings for latency timer values in units of 0.25 µs. The Min_Gnt field specifies how long of a burst period the device needs. The LSIFC949X sets this register to 0x10, indicating a burst period of 4.0 µs.

Register: 0x3F
Maximum Latency
Read Only

7							0
Max_Lat							
0	0	0	0	0	1	1	0

Max_Lat **[7:0]**
This register specifies the desired settings for latency timer values in units of 0.25 µs. The Max_Lat field specifies how often the device needs to gain access to the PCI bus. The LSIFC949X sets this register to 0x06, indicating a burst period of 1.5 µs.

Register: 0xXX
Power Management Capability ID
Read Only

7							0
Power Management Capability ID							
0	0	0	0	0	0	0	1

Power Management Capability ID **[7:0]**
This register indicates the type of the current data structure. It is set to 0x01 to indicate the Power Management data structure.

Register: 0xXX
Power Management Next Pointer
Read Only

7							0
Power Management Next Pointer							
X	X	X	X	X	X	X	X

Power Management Next Pointer **[7:0]**
This register contains the pointer to the next item in the PCI function extended capabilities list. The value of this register varies according to system configuration.

Register: 0xXX

Power Management Capabilities
Read Only

15								8	7						0
Power Management Capabilities															
0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0

PME_Support **[15:11]**
These bits define the power management states in which the device asserts the Power Management Event (PME) pin. The LSIFC949X clears these bits since the LSIFC949X does not provide a PME signal.

D2_Support **10**
The PCI function sets this bit since the LSIFC949X supports power management state D2.

D1_Support **9**
The PCI function sets this bit since the LSIFC949X supports power management state D1.

Aux_Current **[8:6]**
The PCI function clears this field since the LSIFC949X does not support Aux_Current.

Device Specific Initialization **5**
The PCI function clears this bit since no special initialization is required before a generic class device driver can use it.

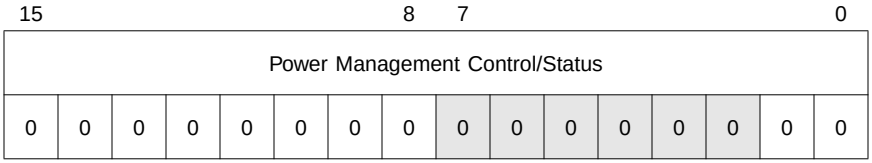
Reserved **4**
This bit is reserved.

PME Clock **3**
The LSIFC949X clears this bit since the chip does not provide a PME pin.

Version **[2:0]**
The PCI function programs these bits to 0b010 to indicate that the LSIFC949X complies with the *PCI Bus Power Management Interface Specification,* Revision 1.2.

Register: 0xXX

Power Management Control/Status
Read/Write



PME_Status **15**
The PCI function clears this bit since the LSIFC949X does not support PME signal generation from $D3_{cold}$.

Data_Scale **[14:13]**
The PCI function clears these bits since the LSIFC949X does not support the Power Management Data register.

Data_Select **[12:9]**
The PCI function clears these bits since the LSIFC949X does not support the Power Management Data register.

PME_Enable **8**
The PCI function clears this bit since the LSIFC949X does not provide a PME signal and disables PME assertion.

Reserved **[7:2]**
This field is reserved.

Power State **[1:0]**
These bits determine the current power state of the LSIFC949X. Power states are:

0b00	D0
0b01	D1
0b10	D2
0b11	$D3_{hot}$

Register: 0xXX

Power Management Bridge Support Extensions
Read Only

7							0
Power Management Bridge Support Extensions							
0	0	0	0	0	0	0	0

Power Management Bridge Support Extensions [7:0]
This register indicates PCI Bridge specific functionality. The LSIFC949X always returns 0x00 in this register.

Register: 0xXX

Power Management Data
Read Only

7							0
Power Management Data							
0	0	0	0	0	0	0	0

Power Management Data [7:0]
This register provides an optional mechanism for the PCI function to report state-dependent operating data. The LSIFC949X always returns 0x00 in this register.

Register: 0xXX

MSI Capability ID
Read Only

7							0
MSI Capability ID							
0	0	0	0	0	1	0	1

MSI Capability ID [7:0]
This register indicates the type of the current data structure. This register always returns 0x05, indicating Message Signaled Interrupts (MSI).

Register: 0xXX
MSI Next Pointer
Read Only

7							0
MSI Next Pointer							
x	x	x	x	x	x	x	x

MSI Next Pointer **[7:0]**
This register points to the next item in the PCI function's extended capabilities list. The value of this register varies according to system configuration.

Register: 0xXX
MSI Message Control
Read/Write

15							8	7							0
MSI Message Control															
0	0	0	0	0	0	0	X	1	0	0	0	0	0	0	0

Reserved **[15:9]**
This field is reserved.

Per-Vector Masking Capable **8**
If this bit is set, the device supports MSI per-vector masking. If this bit is cleared, the function does not support MSI per-vector masking. This bit is read only.

64-Bit Address Capable **7**
The PCI function sets this read only bit to indicate support of a 64-bit message address.

Multiple Message Enable **[6:4]**
These read/write bits indicate the number of messages that the host allocates to the LSIFC949X. The host system software allocates all or a subset of the requested messages by writing to this field. The number of allocated request messages must align to a power of two. Table 6.3 provides the bit encoding of this field.

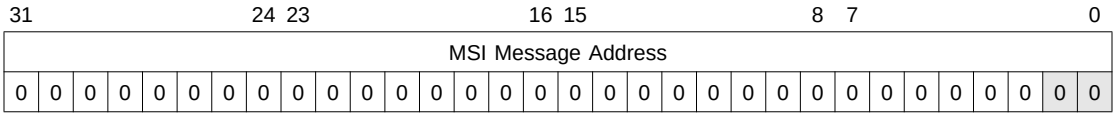
Table 6.3 Multiple Message Enable Field Bit Encoding

Bits [6:4] Encoding	Number of Allocated Messages
0b000	1
0b001	2
0b010	4
0b011	8
0b100	16
0b101	32
0b110	Reserved
0b111	Reserved

Multiple Message Capable [3:1]
These read only bits indicate the number of messages that the LSIFC949X requests from the host. The host system software reads this field to determine the number of requested messages. The number of requested messages must align to a power of two. The LSIFC949X sets this field to 0b000 to request one message. All other encodings of this field are reserved.

MSI Enable 0
System software sets this bit to enable MSI. To enable MSI, the MSI-X bit in the MSI-X Message Control register must also be cleared (“0”). Setting this bit enables the device to use MSI to interrupt the host and request service. Setting this bit prohibits the LSIFC949X from using the INTA/ pin to request service from the host. Setting this bit to mask interrupts on the INTA/ pin is a violation of the PCI specification.

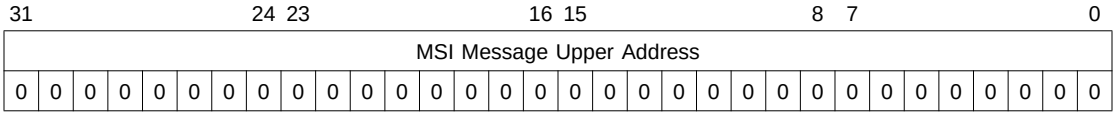
Register: 0xXX
MSI Message Lower Address
Read/Write



MSI Message Address **[31:2]**
This register contains message address bits [31:2] for the MSI memory write transaction. The host system specifies and Dword aligns the message address. During the address phase, the LSIFC949X drives Message Address[1:0] to 0b00.

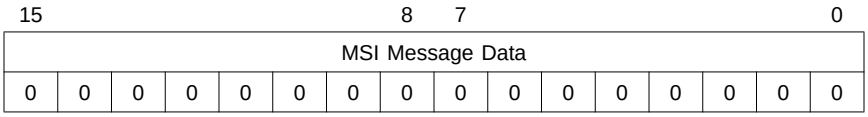
Reserved **[1:0]**
This field is reserved.

Register: 0xXX
MSI Message Upper Address
Read/Write



MSI Message Upper Address **[31:0]**
The LSIFC949X supports 64-bit MSI message. This register contains the upper 32 bits of the 64-bit message address, which the system specifies. The host system software can program this register to 0x0000 to force the PCI function to generate 32-bit message addresses.

Register: 0xXX
MSI Message Data
Read/Write



MSI Message Data **[15:0]**
System software initializes this register by writing to it. The LSIFC949X sends an interrupt message by writing a Dword to the address held in the MSI Message Lower Address and MSI Message Upper Address registers. This register forms bits [15:0] of the Dword message that the PCI function passes to the host. The PCI function drives bits [31:16] of this message to 0x0000.

Register: 0xXX

MSI Mask Bits
Read/Write

31	24	23	16	15	8	7	0
MSI Mask Bits							
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	

MSI Mask Bits **[31:0]**
For each mask bit that is set, the device is prohibited from sending an associated message. Refer to the PCI specification for a complete description of this register.egister

Register: 0xXX

MSI Pending Bits
Read Only

31	24	23	16	15	8	7	0
MSI Pending Bits							
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	

MSI Pending Bits **[31:0]**
For each Pending bit that is set, the function has a pending associated message. Refer to the PCI specification for a complete description of this register.

Register: 0xXX

MSI-X Capability ID
Read Only

7							0
MSI-X Capability ID							
0	0	0	1	0	0	0	1

MSI-X Capability ID **[7:0]**
This register indicates the type of the current data structure. This register always returns 0x11, indicating MSI-X.

Register: 0xXX

MSI-X Next Pointer

Read Only

7							0
MSI-X Next Pointer							
x	x	x	x	x	x	x	x

MSI-X Next Pointer **[7:0]**

This register points to the next item in the extended capabilities list. The value of this register varies according to system configuration.

Register: 0xXX

MSI-X Message Control

Read/Write

15								8	7						0
MSI-X Message Control															
0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x

MSI-X Enable **15**

Setting this bit enables the device to use MSI-X to request service from the host. To enable MSI-X, the MSI Enable bit in the MSI Message Control register must be cleared (“0”). Setting this bit also prohibits the device from using the INTA/ pin to request service from the host. Setting this bit to mask interrupts on the INTA/ pin is a violation of the PCI specification.

Function Mask **14**

Setting this bit masks all of the reset vectors that are associated with the function. This bit overrides the per-vector mask bit settings. Clearing this bit enables the per-vector mask bit to determine if a vector is masked.

Reserved **[13:11]**

This field is reserved.

Table Size **[10:0]**

Host software reads this field to determine the MSI-X table size.

Register: 0xXX
MSI-X Table Offset
Read Only

31	24	23	16	15	8	7	0
MSI-X Table Offset							
x x x x x x x x	x x x x x x x x	x x x x x x x x	x x x x x x x x				

MSI-X Table Offset **[31:3]**
This field provides an offset from the address held in the base address registers of the device to the base of the MSI-X table.

Table BIR **[2:0]**
This field indicates which of the base address registers of the device, which are located at 0x10 in PCI Configuration Space, maps the MSI-X table into memory. Table 6.4 provides the BIR field definitions.

Table 6.4 BIR Field Definitions

BIR Value	Base Address Register
0	0x10
1	0x14
2	0x18
3	0x1C
4	0x20
5	0x24
6	Reserved
7	Reserved

Register: 0xXX
MSI-X PBA Offset
Read Only

31	24	23	16	15	8	7	0
MSI-X PBA Offset							
x x x x x x x x	x x x x x x x x	x x x x x x x x	x x x x x x x x				

MSI-X PBA Offset **[31:3]**
This field contains an offset from one of the base address registers of the device that points to the MSI-X PBA. The lower 3 bits of this register are cleared (“0”) for a 32-bit aligned offset.

PBA BIR **[2:0]**
This field indicates which of the base address registers of the device, which are located at 0x10 in PCI Configuration Space, maps the MSI-X PBA into memory. Table 6.4 provides the BIR field definitions.

Register: 0xXX
PCI-X Capability ID
Read Only

7							0
PCI-X Capability ID							
0	0	0	0	0	1	1	1

PCI-X Capability ID **[7:0]**
This register indicates the type of the current data structure. This register returns 0x07, indicating the PCI-X Data Structure.

Register: 0xXX
PCI-X Next Pointer
Read Only

7							0
PCI-X Next Pointer							
x	x	x	x	x	x	x	x

PCI-X Next Capabilities Pointer **[7:0]**
This register points to the next item in the device’s capabilities list. The value of this register varies according to system configuration.

Register: 0xXX
PCI-X Command
Read/Write

15							8	7							0
PCI-X Command															
0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0

Reserved **[15:7]**
This field is reserved.

Maximum Outstanding Split Transactions **[6:4]**
These bits indicate the maximum number of split transactions the LSIFC949X can have outstanding at one time. The LSIFC949X uses the most recent value of this register each time it prepares a new sequence. Note that if the LSIFC949X prepares a sequence before the setting of this field changes, the PCI function initiates the prepared sequence with the previous setting. Table 6.5 provides the bit encodings for this field.

Table 6.5 Maximum Outstanding Split Transactions

Bits [6:4] Encoding	Maximum Outstanding Split Transactions
0b000	1
0b001	2
0b010	3
0b011	4
0b100	8
0b101	12
0b110	16
0b111	Reserved

Maximum Memory Read Byte Count **[3:2]**
These bits indicate the maximum byte count the LSIFC949X uses when initiating a sequence with one of the burst memory read commands. Table 6.6 provides the bit encodings for this field.

Table 6.6 Maximum Memory Read Count

Bits [3:2] Encoding	Maximum Memory Read Byte Count
0b00	512
0b01	1024
0b10	2048
0b11	Reserved

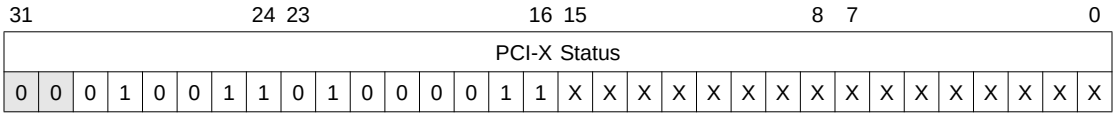
Reserved **1**
This bit is reserved.

Data Parity Error Recovery Enable **0**
The host device driver sets this bit to allow the LSIFC949X to attempt to recover from data parity errors. If the user clears this bit and the LSIFC949X is operating in the PCI-X mode, the LSIFC949X asserts SERR/ whenever the Master Data Parity Error bit in the PCI Status register is set.

Register: 0xXX

PCI-X Status

Read/Write



Reserved **[31:30]**
This field is reserved.

Received Split Completion Error Message **29**
The LSIFC949X sets this bit upon receipt of a split completion message if the split completion error attribute bit is set. Write a one (1) to this bit to clear it.

Designed Maximum Cumulative Read Size **[28:26]**
These read only bits indicate a number greater than or equal to the maximum cumulative size of all outstanding burst memory read transactions for the LSIFC949X PCI device. The PCI function must report the smallest value that correctly indicates its capability. The LSIFC949X

reports 0b100 in this field to indicate a designed maximum cumulative read size of 16 Kbytes.

Designed Maximum Outstanding Split Transactions [25:23]

These read only bits indicate a number greater than or equal to the maximum number of all outstanding split transactions for the LSIFC949X PCI device. The PCI function must report the smallest value that correctly indicates its capability. The LSIFC949X reports 0b110 in this field to indicate that the designed maximum number of outstanding split transactions is sixteen.

Designed Maximum Memory Read Byte Count [22:21]

These read only bits indicate a number greater than or equal to the maximum byte count for the LSIFC949X device. The PCI function uses this count to initiate a sequence with one of the burst memory read commands. The PCI function must report the smallest value that correctly indicates its capability. The LSIFC949X reports 0b10 in this field to indicate that the designed maximum memory read bytes count is 2048.

Device Complexity 20

The PCI function clears this read only bit to indicate that the LSIFC949X is a simple device.

Unexpected Split Completion 19

The PCI function sets this read only bit when it receives an unexpected split completion. Once set, this bit remains set until software clears it. Write a one (1) to this bit to clear it.

Split Completion Discarded 18

The PCI function sets this read only bit when it discards a split completion. Once set, this bit remains set until software clears it. Write a one (1) to this bit to clear it.

133 MHz Capable 17

The MA[8] Power-On Sense pin controls this read only bit. Allowing the internal pull-downs to pull MA[8] LOW sets this bit and enables 133 MHz operation of the PCI bus. Pulling MA[8] HIGH clears this bit and disables 133 MHz operation of the PCI bus. Refer to SEN #S11066,

"LSIFC949X Design Considerations," for details on the Power-On Sense pins.

64-Bit Device **16**
The MA[9] Power-On Sense pin controls this read only bit. Allowing the internal pull-downs to pull MA[9] LOW sets this bit and indicates a 64-bit PCI Address/Data bus. Pulling MA[9] HIGH clears this bit and indicates a 32-bit PCI Address/Data bus. If using the LSIFC949X on an add-in card, this bit must indicate the size of the card's PCI Address/Data bus. Refer to SEN #S11066, "LSIFC949X Design Considerations," for details on the Power-On Sense pins.

Bus Number **[15:8]**
These read only bits indicate the number of the LSIFC949X bus segment. The PCI function uses this number as part of its Requester ID and Completer ID. This field is read for diagnostic purposes only.

Device Number **[7:3]**
These read only bits indicate the device number of the LSIFC949X. The PCI function uses this number as part of its Requester ID and Completer ID. This field is read for diagnostic purposes only.

Function Number **[2:0]**
These read only bits indicate the number in the Function Number field (AD[10:8]) of a Type 0 PCI configuration transaction. The PCI function uses this number as part of its Requester ID and Completer ID. This field is read for diagnostic purposes only.

6.2 PCI I/O Space and Memory Space Register Description

This section describes the host interface registers in the PCI I/O Space and in the PCI Memory Space. These address spaces contain the Fusion-MPT interface register set. PCI Memory Space [0] and PCI Memory Space [1] form the PCI Memory Space. PCI Memory Space [0] supports normal memory accesses while PCI Memory Space [1] supports diagnostic memory accesses. For all registers except the Diagnostic Read/Write Data and Diagnostic Read/Write Address registers, access the address offset through PCI I/O Space. Access to

the Diagnostic Read/Write Data and Diagnostic Read/Write Address registers is available only through PCI I/O Space.

When the LSIFC949X operates as a multifunction PCI device, the entire PCI Memory and PCI I/O Space register sets are visible to both PCI functions. When the LSIFC949X operates as a single function PCI device, only PCI Function [0] register sets are accessible.

Table 6.7 defines the PCI I/O Space address map.

Table 6.7 PCI I/O Space Address Map

31 16 15 0	Offset	Page
System Doorbell	0x00	6-34
Write Sequence	0x04	6-35
Host Diagnostic	0x08	6-36
Test Base Address	0x0C	6-38
Diagnostic Read/Write Data	0x10	6-38
Diagnostic Read/Write Address	0x14	6-39
Reserved	0x18–0x2F	–
Host Interrupt Status	0x30	6-39
Host Interrupt Mask	0x34	6-40
Reserved	0x38–0x3F	–
Request FIFO	0x40	6-42
Reply FIFO	0x44	6-42
High Priority Request FIFO	0x48	6-43
Reserved	0x4C	–
Host Index Register	0x50	6-43
Reserved	0x54–0x7F	–

Table 6.8 defines the PCI Memory Space [0] address map.

Table 6.8 PCI Memory [0] Address Map

31 16 15 0	Offset	Page
System Doorbell	0x00	6-34
Write Sequence	0x04	6-35
Host Diagnostic	0x08	6-36
Test Base Address	0x0C	6-38
Reserved	0x10–0x2F	–
Host Interrupt Status	0x30	6-39
Host Interrupt Mask	0x34	6-40
Reserved	0x38–0x3F	–
Request FIFO	0x40	6-42
Reply FIFO	0x44	6-42
High Priority Request FIFO	0x48	6-43
Reserved	0x4C–0x7F	–

Table 6.9 defines the PCI Memory Space [1] address map.

Table 6.9 PCI Memory [1] Address Map

31 16 15 0	
Diagnostic Memory	0x00– 0x(Sizeof(Mem1)–1)

A bit-level description of the PCI Memory and PCI I/O spaces follows.

Register: 0x00
System Doorbell
Read/Write

31 24	23 16	15 8	7 0
System Doorbell			

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

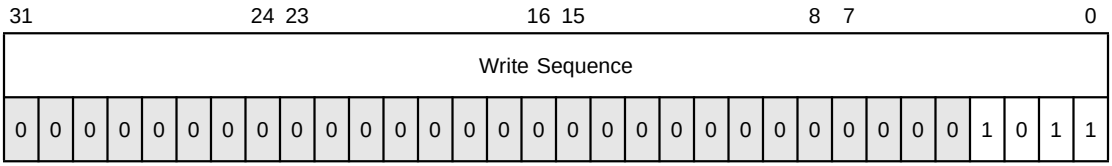
This register is a simple message passing mechanism that allows the system to pass single word messages to the embedded IOP processor and vice versa. There is a unique system doorbell for each PCI function.

When a host system PCI master writes to the Host Registers → Doorbell register, the LSIFC949X generates a maskable interrupt to the IOP. The value written by the host system is available for the IOP to read in the System Interface Registers → Doorbell register. The IOP clears the interrupt status after reading the value.

Conversely, when the IOP processor writes to the System Interface Registers → Doorbell register, the LSIFC949X generates a maskable interrupt to the PCI system. The host system can read the value written by the IOP in the Host Registers → Doorbell register. The host system clears the interrupt status bit and interrupt pin by writing any value to the Host Registers → Interrupt Status register.

Host Doorbell Value **[31:0]**
During a write, this register contains the doorbell value that the host system passes to the IOP. During a read, this register contains the doorbell value that the IOP passes to the host system.

Register: 0x04
Write Sequence
Read/Write



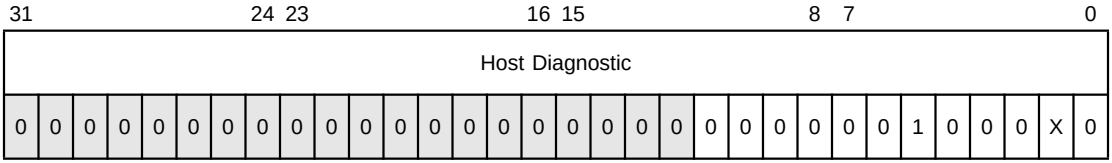
This register provides a protection mechanism against inadvertent writes to the Host Diagnostic register. There is one Write I/O register that is visible to both PCI functions. The two PCI functions physically share this register.

Reserved **[31:4]**
This field is reserved.

Write I/O Key **[3:0]**
To enable write access to the Diagnostic Read/Write Data, Diagnostic Read/Write Address, and Host Diagnostic register, perform five data-specific writes to the Write I/O Key. Writing an incorrect value to the Write I/O Key invalidates the key sequence, and the host must rewrite the entire sequence. The write I/O key sequence is: 0x0004, 0x000B, 0x0002, 0x0007, and 0x000D. To disable write access to the Diagnostic Read/Write Data, Diagnostic Read/Write Address, and Host Diagnostic registers, write any value (except the Write I/O Key sequence) to the Write I/O register. The Diagnostic Write

Enable bit (bit 7 in the Host Diagnostic register) indicates the write access status.

Register: 0x08
Host Diagnostic
Read/Write



This register contains diagnostic controls and status information. There is one Host Diagnostic register that is visible to both PCI functions. The two PCI functions physically share this register. However, the Reset History bit operates independently for each PCI function. This register can only be written when bit 7 of this register is set.

Reserved **[31:12]**
This field is reserved.

BIST Read Enable **11**
Setting this bit enables reading the two BIST results registers (0x18 and 0x1C) from the host.

Clear Flash Bad Signature **10**
Write this bit to clear the Bad Signature bit (bit 6 of this register).

Prevent IOP Boot **9**
Set this bit to keep the IOP in a reset state.

BIST All Done **8**
When this bit is set, all internal built-in self-test (BIST) operations are complete.

Diagnostic Write Enable **7**
The LSIFC949X sets this read-only bit when the host writes the correct Write I/O Key to the Write Sequence register. The LSIFC949X clears this bit when the host writes a value other than the Write I/O Key to the Write Sequence register.

Flash Bad Signature **6**
The LSIFC949X sets this bit if the IOP ARM966E-S processor encounters a bad Flash signature when booting from Flash ROM. The LSIFC949X also sets the DisARM bit (bit 1 in this register) to hold the IOP ARM processor in a reset state. The LSIFC949X maintains this state until the PCI host clears both the Flash Bad Signature and DisARM bits.

Reset History **5**
The LSIFC949X sets this bit if it experiences a Power-On Reset (POR), PCI Reset, or TestReset/. A host driver can clear this bit to help coordinate recovery between multiple driver instances in a multifunction PCI implementation.

Diagnostic Read/Write Enable 4
Setting this bit enables access to the Diagnostic Read/Write Data and Diagnostic Read/Write Address registers.

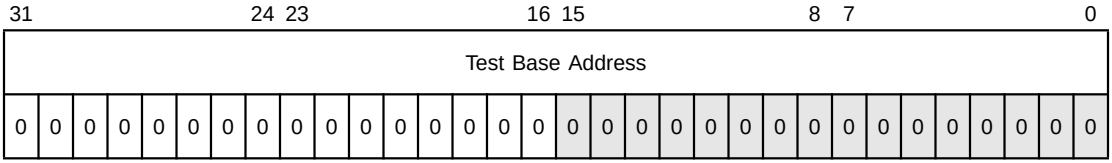
TTL Interrupt **3**
Setting this bit configures PCI INTA/ as a TTL output. Clearing this bit configures PCI INTA/ as an open-drain output. Use this bit for test purposes only.

Reset Adapter **2**
Setting this write-only bit causes a hard reset within the LSIFC949X. The bit self-clears after eight PCI clock periods. After deasserting this bit, the IOP ARM processor executes from its default reset vector.

DisARM **1**
Setting this bit disables the IOP ARM processor.

Diagnostic Memory Enable **0**
Setting this bit enables diagnostic memory accesses through PCI Memory Space [1]. Clearing this bit disables diagnostic memory accesses to PCI Memory Space [1] and returns 0xFFFF on reads.

Register: 0x0C
Test Base Address
Read/Write

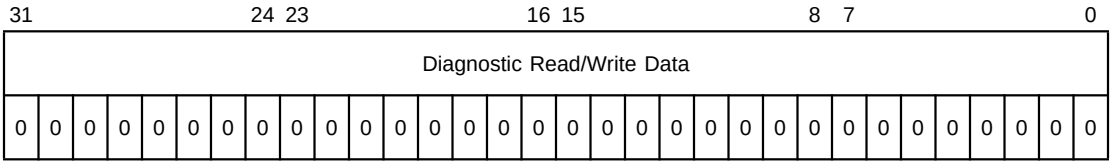


This register specifies the base address for Memory Space [1] accesses. There is one Test Base Address register that is visible to both PCI functions. The two PCI functions physically share this register. Because Diagnostic Memory is visible only to PCI Function [0], PCI Function [1] cannot write to this register.

Test Base Address **[31:16]**
The number of significant bits is determined by the size of PCI Memory Space [1] in the serial EEPROM.

Reserved **[15:0]**
This field is reserved.

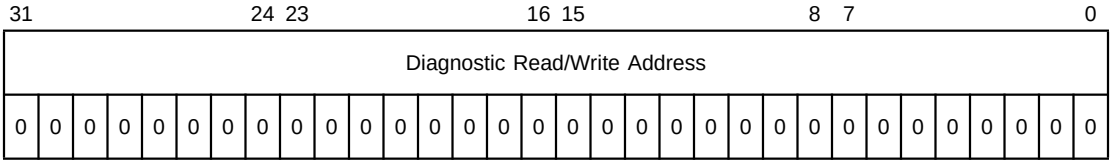
Register: Offset 0x10
Diagnostic Read/Write Data
Read/Write



This register reads or writes Dword locations on the LSIFC949X internal bus. This register is only accessible through PCI I/O space and returns 0xFFFFFFFF if read through PCI Memory Space. The host can enable write access to this register by writing the correct Write I/O Key to the Write Sequence register and setting bit 4, the Diagnostic Write Enable bit, of the Host Diagnostic register. A write of any value other than the correct Write I/O Key to the Write Sequence register disables write access to this register. There is one Diagnostic Read/Write Data register that is visible to both PCI functions. The two PCI functions physically share this register.

Diagnostic Read/Write Data **[31:0]**
Using this register, the LSIFC949X reads/writes data at the address that the Diagnostic Read/Write Address register specifies.

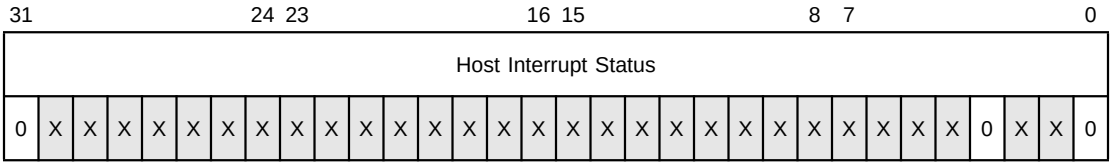
Register: 0x14
Diagnostic Read/Write Address
Read/Write



This register specifies a Dword location on the internal bus. The address increments by a Dword whenever the host system accesses the Diagnostic Read/Write Address register. This register is only accessible through PCI I/O space and returns 0xFFFFFFFF if read through PCI Memory Space. The host can enable write access to this register by writing the correct Write I/O Key to the Write Sequence register and setting bit 4, the Diagnostic Write Enable bit, of the Host Diagnostic register. A write of any value other than the correct Write I/O Key to the Write Sequence register disables write access to this register. There is one Diagnostic Read/Write Address register that is visible to both PCI functions. The two PCI functions physically share this register.

Diagnostic Read/Write Address **[31:0]**
This register holds the address that the Diagnostic Read/Write Data register writes data to or reads data from.

Register: 0x30
Host Interrupt Status
Read Only



This register provides read-only interrupt status information to the PCI Host. A write to this register of any value clears the associated System Doorbell interrupt. There is a unique Host Interrupt Status register for each PCI function.

IOP Doorbell Status **31**
The LSIFC949X sets this bit when the IOP receives a message from the system doorbell but has yet to process it. The IOP processes the System Doorbell message by clearing the corresponding system request interrupt.

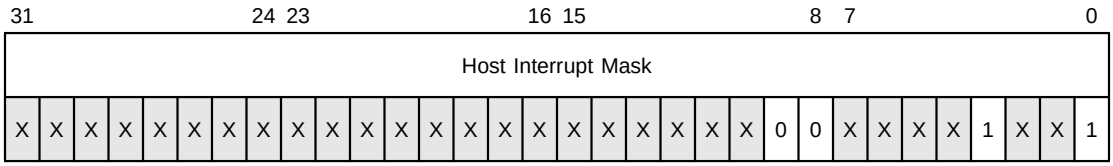
Reserved **[30:4]**
This field is reserved.

Reply Interrupt **3**
The LSIFC949X sets this bit when the Reply Post FIFO is not empty. The LSIFC949X generates a PCI interrupt when this bit is set and the corresponding mask bit in the Host Interrupt Mask register is cleared.

Reserved **[2:1]**
This field is reserved.

System Doorbell Interrupt **0**
The LSIFC949X sets this bit when the IOP writes a value to the System Doorbell. The host can clear this bit by writing any value to this register. The LSIFC949X generates a PCI interrupt when this bit is set and the corresponding mask bit in the Host Interrupt Mask register is cleared.

Register: 0x34
Host Interrupt Mask
Read/Write



This register masks and/or routes the interrupt conditions that the Host Interrupt Status register reports. There is a unique Host Interrupt Mask register for each PCI function.

Reserved **[31:10]**
This field is reserved.

Interrupt Request Routing Mode **[9:8]**
This field routes PCI interrupts to the INTx/ pins according to the bit encodings in Table 6.10. If the host system enables MSI, the LSIFC949X does not signal PCI interrupts on the INTx/ pins.

Table 6.10 Interrupt Signal Routing

Bit [9:8] Encodings	Interrupt Signal Routing
0b00	INTx/ and ALT_INTx/
0b01	INTx/ Only
0b10[1]	ALT_INTx/ Only
0b11	INTx/ and ALT_INTx/

1. The LSIFC949X does not support alternate interrupt signals (no device pins are provided). Programming this field to 0b10 effectively disables PCI interrupts for the given PCI function.

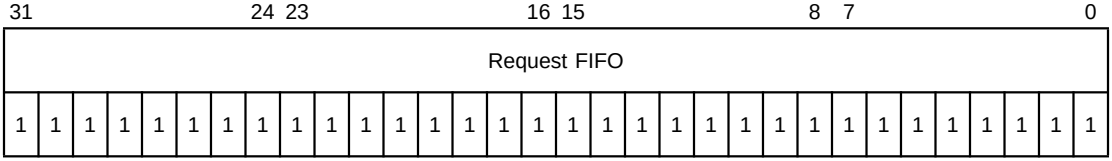
Reserved **[7:4]**
This field is reserved.

Reply Interrupt Mask **3**
Setting this bit masks reply interrupts and prevents the assertion of a PCI interrupt for all reply interrupt conditions.

Reserved **[2:1]**
This field is reserved.

Doorbell Interrupt Mask **0**
Setting this bit masks System Doorbell interrupts and prevents the assertion of a PCI interrupt for all System Doorbell interrupt conditions.

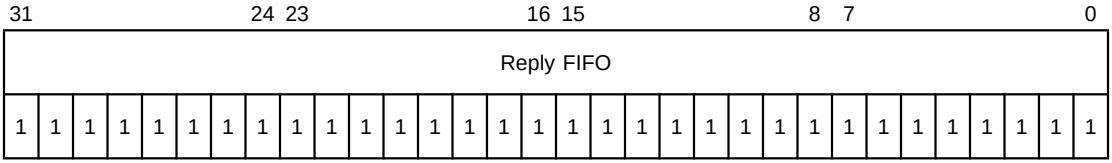
Register: 0x40
Request FIFO
Read/Write



This register provides Request Free Message Frame Addresses (MFAs) to the host system on reads and accepts Request Post MFAs from the host system on writes. There is one Request FIFO register that is visible to both PCI functions. The two PCI functions physically share this register.

Request FIFO **[31:0]**
For reads, the Request Free MFA is empty and this register contains 0xFFFFFFFF. For writes, the register contains the Request Post MFA.

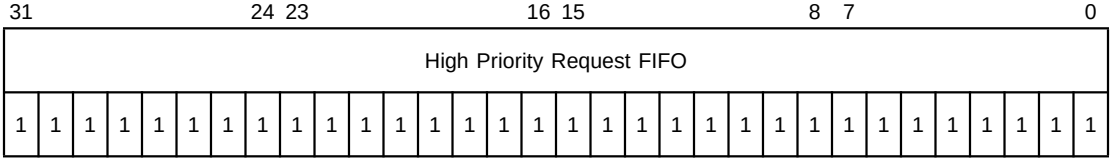
Register: 0x44
Reply FIFO
Read/Write



This register provides Reply Post MFAs to the host system on reads and accepts Reply Free MFAs from the host system on writes. There is one unique Reply FIFO register for each PCI function.

Reply FIFO **[31:0]**
For reads, the Request Free MFA is empty and this register contains 0xFFFFFFFF. For writes, the register contains the Reply Free MFA.

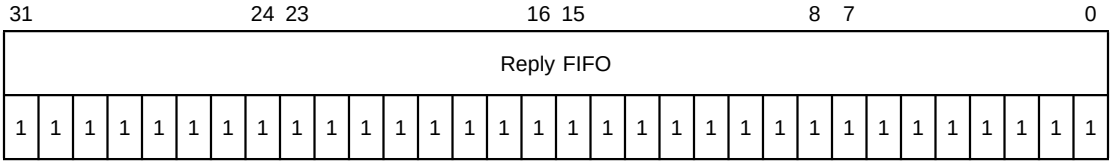
Register: 0x48
High Priority Request FIFO
Read/Write



The High Priority Request FIFO is used to provide High Priority Request Free MFAs to the host on reads, and to accept High Priority Request Post MFAs from the host on writes. These MFAs will not be automatically pulled even if the EnHWPull bit is set. This is a hardware FIFO with a maximum depth of 256 32-bit entries.There is one High Priority Request FIFO register which is visible to both, but physically shared between the two PCI Functions. PCI Function information is saved with the MFA in the singular High Priority Request FIFO.

High Priority Request FIFO **[31:0]**
For reads, the High Priority Request FIFO is empty and this register contains 0xFFFFFFFF. For writes, the register contains the High Priority Request Post MFA.

Register: 0x50
Host Index Register
Read/Write



These registers are used with the Outbound Reply Option (AltReplyPost method) to enable host-resident reply post queues.

Reserved **[31:14]**
This field is reserved.

Host Index Value **[13:0]**
The Host Index provides an indication of which Reply Post MFAs the host system has processed, and

generates Reply Interrupts when the AltReplyPost option is enabled. There is a unique Host Index register associated with each PCI Function.

Chapter 7
Specifications

This chapter provides a description of the DC and AC electrical characteristics of the LSIFC949X Dual Channel Fibre Channel I/O processor chip, and the available packaging. The chapter contains the following sections:

- Section 7.1, “Electrical Requirements”
- Section 7.2, “AC Timing”
- Section 7.3, “Packaging”

7.1 Electrical Requirements

Table 7.1 provides absolute maximum stress ratings for the LSIFC949X, while Table 7.2 specifies the normal operating conditions. Table 7.3 through Table 7.10 specify the input and output electrical characteristics.

Table 7.1 Absolute Maximum Stress Ratings[1]

Symbol	Parameter	Min	Max	Unit	Test Conditions
T_{STG}	Storage temperature	– 55	150	°C	–
V_{DD}	Supply voltage	– 0.5	4.5	V	–
V_{IN}	Input voltage	V_{SS} – 0.3	V_{DD} + 0.3	V	–
I_{LP}[2]	Latch-up current	±150	–	mA	EIA/JESD78
ESD_{HBM}	Electrostatic discharge – Human Body Model (HBM)	–	1.5	kV	JESD-A114-B

1. Stresses beyond those listed in this table may cause permanent damage to the device. These are stress ratings only; functional operation of the device at these or any other conditions beyond those indicated in the Operating Conditions section of the manual is not implied.
2. See EIA/JESD78 for further information on latch-up testing.

Table 7.2 Operating Conditions[1]

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{DDC}[2]	Core supply voltage	1.14	1.26	V	–
V_{DDIO}	I/O supply voltage	3.0	3.6	V	–
I_{DDC}[3]	Core supply current	–	1100	mA	–
I_{DDIO}	I/O supply current	–	950	mA	–
T_A[4]	Operating free air	0	70	˚C	–
θ_{JMA}	Thermal resistance (junction to moving air)	–	11.2	˚C/W	–

1. Conditions that exceed the operating limits may cause the device to function incorrectly.
2. Refer to Note 1 at the end of Table 4.6 (page 4-17) for instructions on power sequencing for the LSIFC949X.
3. The maximum current specification for I_{DDC} includes any current drawn by the analog PLL.
4. The LSIFC949X does not require a heatsink when operating within the temperature range specified in this table.

Table 7.3 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 7.4 Input Signals (FAULT1/, FAULT0/, MODE[7:0], SWITCH, CPCI_EN/)

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IH}	Input high voltage	0.7 V_{DD}	V_{DD} + 0.3	V	–
V_{IL}	Input low voltage	V_{SS} – 0.3	0.2 V_{DD}	V	–
I_{IN}	Input leakage	10	10	µA	–

Table 7.5 Schmitt Input Signals (REFCLK, TCK, TDI, TRST/, TMS_CHIP, TMS_ICE)

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IH}	Input high voltage	2.0	V_{DD} + 0.3	V	–
V_{IL}	Input low voltage	V_{SS} – 0.3	0.8	V	–
I_{IN}	Input leakage	10	10	µA	–

Table 7.6 4 mA Bidirectional Signals (LIPRESET/, ODIS1, ODIS0, BYPASS1/, BYPASS0/, MD[31:0], MA[23:0], MWE[1:0]/, FLASHCS/, BWE[3:0]/, RAMCS/, MP[3:0], SCL, SDA, RXLOS1, RXLOS0, ADSC/, ADV/, TDO)

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IH}	Input high voltage	0.7 V_{DD}	V_{DD} + 0.3	V	–
V_{IL}	Input low voltage	V_{SS} – 0.3	0.2 V_{DD}	V	–
V_{OH}	Output high voltage	2.4	V_{DD}	V	–4 mA
V_{OL}	Output low voltage	V_{SS}	0.4	V	4 mA
I_{OZ}	3-state leakage	–10	10	µA	–

Table 7.7 8 mA Bidirectional Signals (MODDEF1[2:0], MODDEF0[2:0], GPIO[5:0], MOE[1:0]/, LED[4:0]/, MCLK)

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IH}	Input high voltage	0.7 V_{DD}	V_{DD} + 0.3	V	–
V_{IL}	Input low voltage	V_{SS} – 0.3	0.2 V_{DD}	V	–
V_{OH}	Output high voltage	2.4	V_{DD}	V	–8 mA
V_{OL}	Output low voltage	V_{SS}	0.4	V	8 mA
I_{OZ}	3-state leakage	–10	10	µA	–

Table 7.8 PCI Input Signals (PCICLK, GNT/, IDSEL, RST/)

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IH}	Input high voltage	0.5 V_{DD}	5.5	V	3.3 V PCI System
V_{IL}	Input low voltage	–0.5	0.3 V_{DD}	V	3.3 V PCI System

Table 7.9 PCI Bidirectional Signals (AD[63:0], C_BE[7:0]/, FRAME/, IRDY/, TRDY/, STOP/, PERR/, PAR, ACK64/, ENUM/, 64EN/)

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IH}	Input high voltage	0.5 V_{DD}	5.5	V	3.3 V PCI System
V_{IL}	Input low voltage	–0.5	0.3 V_{DD}	V	3.3 V PCI System
V_{OH}	Output high voltage	0.9 V_{DD}	V_{DD}	V	–0.5 mA
V_{OL}	Output low voltage	V_{SS}	0.1 V_{DD}	V	1.5 mA
I_{OZ}	3-state leakage	–10	10	µA	–

Table 7.10 PCI Output Signals (PAR64, REQ/, REQ64/, DEVSEL/, SERR/, INTA/, INTB/)

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{OH}	Output high voltage	0.9 V_{DD}	V_{DD}	V	–0.5 mA
V_{OL}	Output low voltage	V_{SS}	0.1 V_{DD}	V	1.5 mA
I_{OZ}	3-state leakage	–10	10	µA	–

7.2 AC Timing

The AC Timing characteristics described in this section apply over the entire range of operating conditions. Chip timings are based on simulation at worst-case voltage, temperature, and processing. Timings have been developed with a load capacitance of 50 pF.

7.2.1 PCI/PCI-X Interface Timings

The LSIFC949X PCI/PCI-X signals conform to the electrical and timing standards as shown in the *PCI Local Bus Specification,* Revision 2.3, and the *PCI-X Addendum to the PCI Local Bus Specification,* Revision 2.0. All hardware validation testing performed by LSI Logic guarantees that the LSIFC949X meets or exceeds the specifications contained in those documents.

7.2.2 Fibre Channel Interface Timings

The LSIFC949X receiver and transmitter serial differential signal pairs conform to the electrical and timing standards as shown in the Fibre Channel Physical Interface specification (FC-PI, Rev. 11). All hardware validation testing performed by LSI Logic guarantees that the LSIFC949X meets or exceeds the specifications contained in that document.

7.2.3 Memory Interface Timings

See the following sections for memory interface timings descriptions:

- Section 7.2.3.1, “SSRAM Timings,” on page 7-6
- Section 7.2.3.2, “Flash ROM Read Timings,” on page 7-7
- Section 7.2.3.3, “Flash ROM Write Timings,” on page 7-8

7.2.3.1 SSRAM Timings

Figure 7.1 SSRAM Read/Write/Read Timing Waveforms

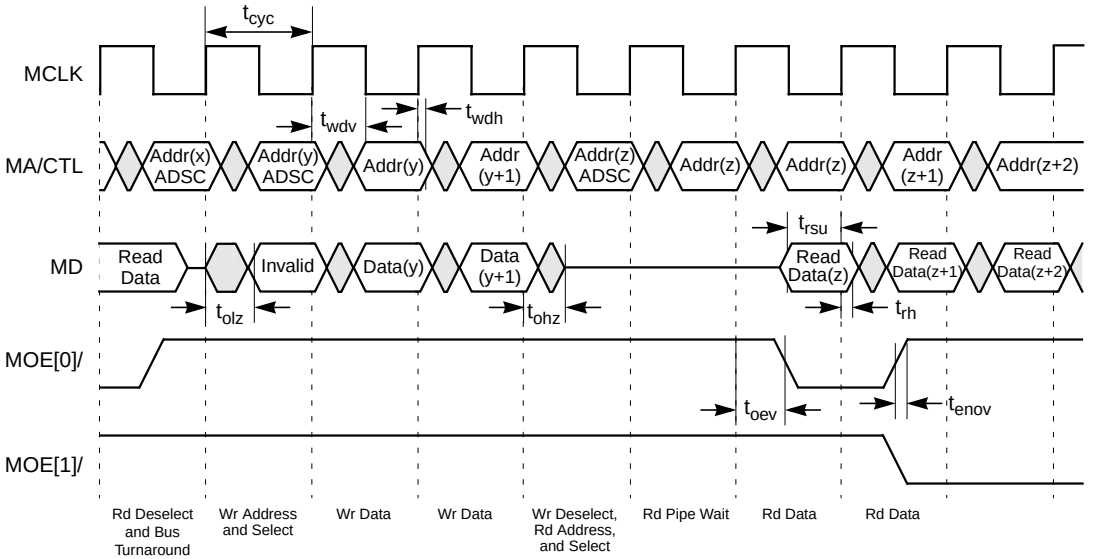


Table 7.11 SSRAM Read/Write/Read Timings

Symbol	Parameter		Min	Max	Unit
t_{cyc}	MCLK cycle time		9.411	9.413	ns
t_{rsu}	Read setup time		4.1	–	ns
t_{rh}	Read hold time		0	–	ns
t_{wdv} [1]	Write valid time	MD[31:0], MP[3:0]	–	5.0	ns
		MA[21:0]	–	6.4	ns
		Control Signals[2]	–	6.6	ns
t_{wdh} [1]	Write hold time	MD[31:0], MP[3:0]	0.1	–	ns
		MA[21:0]	0.3	–	ns
		Control Signals[2]	0.6	–	ns
t_{oev}	Output enable valid		–	0.75	ns
t_{olz}	Data low impedance		0.5	2.5	ns
t_{ohz}	Data high impedance		0	1.75	ns
t_{enov}	Output enable nonoverlap		0	–	ns

1. Refer to SEN #11066, “LSIFC949X Design Considerations”, for further details regarding write valid and write hold times for MD[31:0].
2. Control signals include MWE[1:0]/, BWE[3:0]/, RAMCS/, ADSC/, and ADV/.

7.2.3.2 Flash ROM Read Timings

Figure 7.2 Flash ROM Read Timing Waveforms

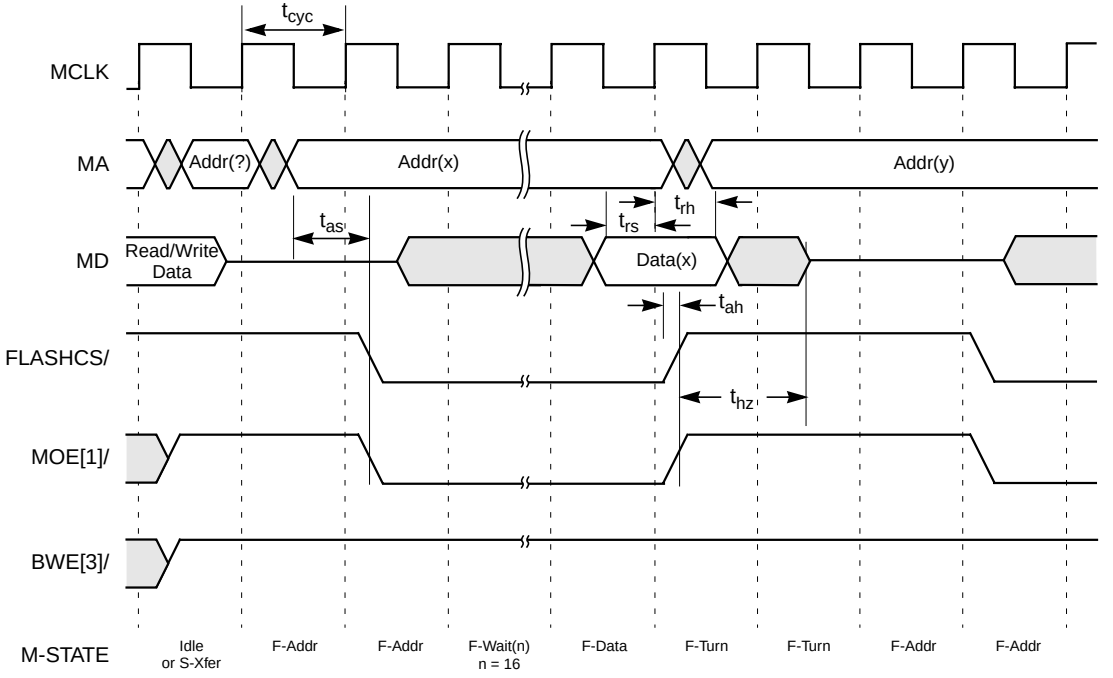


Table 7.12 FLASH ROM Read Timings

Symbol	Parameter	Min	Max	Unit
t_{cyc}	MCLK cycle time	9.410	9.413	ns
t_{as}	Address setup time	– 5.0 [1]	1 – MCLK[2]	ns
t_{ah}	Address hold time	0	–	ns
t_{rs}	Read setup time	7	–	ns
t_{rh}	Read hold time	0	–	ns
t_{hz}	Data high impedance	0	32	ns

1. Address setup time defaults to one (1) MCLK but may be programmed to zero (0) MCLKs using the serial EEPROM.

7.2.3.3 Flash ROM Write Timings

Figure 7.3 Flash ROM Write Timing Waveforms

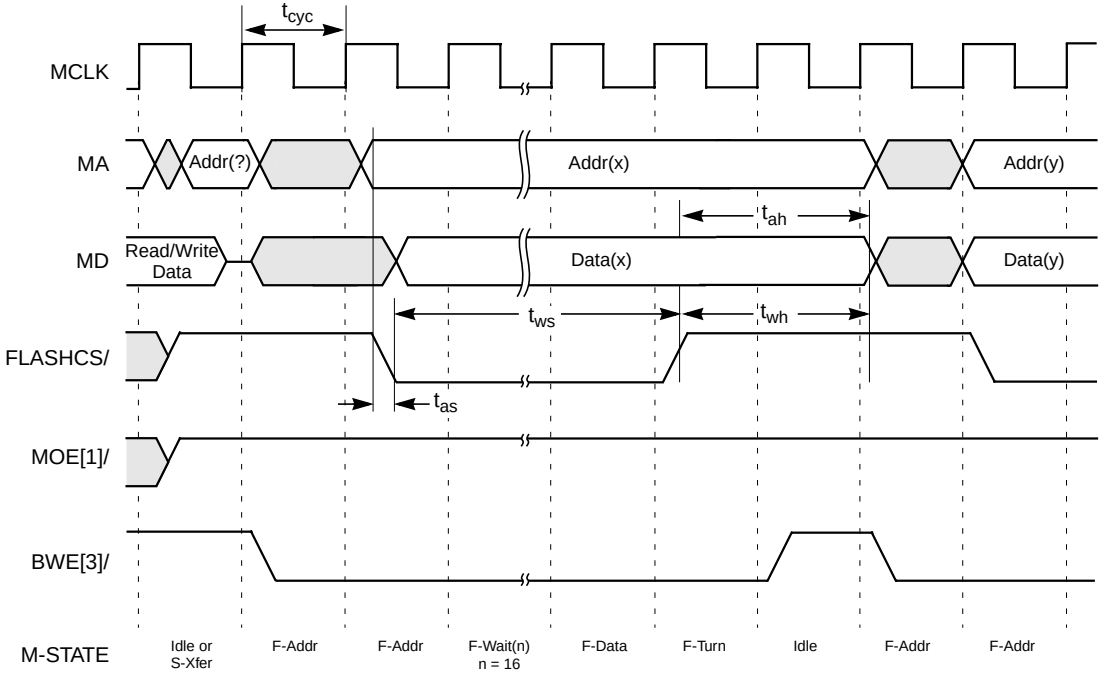


Table 7.13 Flash ROM Write Timings

Symbol	Parameter	Min	Max	Unit
t_{cyc}	MCLK cycle time	9.410	9.413	ns
t_{as}	Address setup time	– 5.0[1]	1 – MCLK[2]	ns
t_{ah}	Address hold time	1 MCLK	–	ns
t_{ws}[2]	Write setup time	2[3]	32	MCLK
t_{wh}	Write hold time	1 – MCLK	–	ns

1. Address setup time defaults to one (1) MCLK but may be programmed to zero (0) MCLKs using the serial EEPROM.
2. The default write setup time is 17 ns.
3. Programmed using the serial EEPROM.

7.3 Packaging

Figure 7.4 illustrates the signal locations for the 544 Flip Chip Plastic Ball Grid Array (FPBGA).

Also in this section are two listings of he alphanumeric pads: Table 7.14 lists them by PBGA position, and Table 7.15 by signal name. And a mechanical drawing of the package for the LSIFC949X (Figure 7.5 on page 7-14).

Figure 7.4 LSIFC949X 544-Pin FPBGA Top View

*	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
A		VSS	MD[11]	MD[15]	MD[21]	MD [22]	MD [26]	MD [28]	MA [12]	MA [10]	MA[6]	MA[0]	VSS	VSS	VSS	VSS	AD[38]	VSS	VSS	AD[49]	VSS	AD[52]	AD[55]	VSS	VDDIO 33_PCIX	
B	VDDIO 33	VSS	VSS	MD[12]	MD[16]	MD [20]	MD [25]	MD [27]	MA [16]	MA [11]	MA[9]	MA[5]	MA[3]	VSS	AD[35]	VSS	AD[39]	VSS	AD[48]	VSS	AD[53]	AD[54]	VSS	VSS	VSS	VSS
C	MD[9]	VSS	VSS	MD[10]	MD[13]	MD [19]	VDDIO 33	VDDIO 33	MA [17]	MA [15]	VSS	VSS	MA[4]	AD[34]	VDDIO 33	VDDIO 33	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
D	MD[6]	MD[8]	VSS	VSS	MD[18]	MD [14]	VSS	VSS	MA [23]	MA [18]	VDDIO 33	VDDIO 33	MA [13]	MA[1]	VSS	VSS	AD[42]	VSS	VDDIO 33_PCIX	VDDIO 33_PCIX	VSS	AD[46]	VSS	VSS	VSS	AD[57]
E	MD[1]	MD[5]	MD[4]	BWE3/	VSS	MD [17]	VDDIO 33	VDDIO 33	MD [31]	MA [21]	VSS	VSS	MA [14]	MA [2]	VDDIO 33_PCIX	VDDIO 33_PCIX	AD[44]	VSS	VSS	VSS	AD[47]	VSS	AD[58]	VSS	VSS	AD[60]
F	MD[0]	MD[2]	MD[3]	MD[7]	MCLK	MP[3]	MD [24]	MD [30]	MA [19]	MA [22]	VDDIO 33	VDDIO 33	MA[7]	AD[32]	VSS	VSS	AD[45]	AD[36]	AD[40]	VSS	AD[51]	VSS	AD[56]	AD[59]	VSS	AD[61]
G	MP[0]	MP[1]	VSS	VDDIO 33	VSS	MWE0/	MP[2]	MD [23]	MD [29]	MA [20]	VSS	VSS	MA[8]	AD[33]	VDDIO 33_PCIX	VDDIO 33_PCIX	AD[37]	AD[41]	AD[43]	AD[50]	AD[62]	VDDIO 33_PCIX	VSS	VDDIO 33_PCIX	AD[63]	VSS
H	ADSC/	ADV/	VSS	VDDIO 33	VSS	MODE [6]	MOE0/													VSS	VSS	VDDIO 33_PCIX	VSS	VDDIO 33_PCIX	VSS	PAR64
J	MODE [5]	RAM CS/	FLASH CS/	BWE1/	BWE2/	MODE [0]	MODE [7]													C_BE [4]/	REQ64/	C_BE [5]/	C_BE [6]/	ACK64/	AD[0]	VSS
K	MODE [3]	MODE [4]	NC	MOE1/	MWE1/	BWE0/	MODE [1]													C_BE [7]/	VSS	VSS	VSS	VSS	AD[3]	AD[4]
L	VSS	MODE [2]	MXSV DD	VSS	VDDIO 33	VSS	VDDIO 33													VSS	VDDIO 33_PCIX	VSS	VDDIO 33_PCIX	VSS	VSS	AD[6]
M	TX VDD0	TXB VSS0	MXS VDD	VSS	VDDIO 33	VSS	VDDIO 33					VSS	VDD12	VSS	VDD12					VSS	VDDIO 33_PCIX	VSS	VDDIO 33_PCIX	VSS	VSS	AD[8]
N	RXB VSS0	TX VSS0	TXB VDD0	RX0+	RX0-	TX0+	TX0-					VDD12	VSS	VDD12	VSS					VSS	AD[5]	AD[1]	AD[2]	VSS	AD[7]	AD[9]
P	RXB VDD0	RX VSS0	TXB VSS1	RX1+	RX1-	TX1-	TX1+					VSS	VDD12	VSS	VDD12					VSS	AD[11]	C_BE [0]/	VSS	AD[12]	VSS	VSS
R	RX VDD0	TXB VDD1	VSS	MXS VDD	VSS	VDDIO 33	VSS					VDD12	VSS	VDD12	VSS					VDDIO 33_PCIX	VSS	VDDIO 33_PCIX	VSS	VDDIO 33_PCIX	AD[13]	AD[10]
T	TX VSS1	TX VDD1	VSS	MXS VDD	VSS	VDDIO 33	VSS													VDDIO 33_PCIX	VSS	VDDIO 33_PCIX	VSS	VDDIO 33_PCIX	VSS	VSS
U	RXB VSS1	RXB VDD1	RX VSS1	VSS	TDIODE VSS	TDIODEP	REF CLKP													AD[15]	DEV SEL/	VSS	PERR/	VSS	PAR	C_BE [1]/
V	RX VDD1	RTRIM	NC	REFCL KB	SCAN_ MODE	REF CLKN	TEST_ RESET/													VSS	AD[14]	VSS	TRDY/	SERR/	VSS	VSS
W	RTCK_ ICE	TCK_ ICE	VDDIO 33	VSS	VDDIO 33	SCAN_ ENABLE	TDO_ ICE													STOP/	VSS	VSS	VDDIO 33_PCIX	VSS	VSS	VSS
Y	TRST_ ICE/	TMS_ ICE	VDDIO 33	VSS	VDDIO 33	TDI_ICE	MOD DEF0[0]	FAULT 0/	LED [2]/	GPIO [2]	VDDIO 33	VDDIO 33	CPCI_ EN/	AD[28]	VSS	VSS	VSS	AD[19]	C_BE [2]/	VSS	VSS	VSS	VDDIO 33_PCIX	VSS	REQ/	IDSEL
AA	TCK	TMS	TDI	SERIAL_ CLOCK	PROC_ MON	MOD DEF0[1]	RX LOS1	LED [1]/	GPIO [3]	BY PASS1	VSS	VSS	NC	AD[29]	VDDIO 33	VDDIO 33	AD[21]	AD[23]	AD[20]	AD[16]	GNT/	FRAME/	ENUM/	INTB/	VSS	RST/
AB	TRST/	IDDTN	SERIAL_ DATA	TDO	UART RX	MOD DEF1[0]	VSS	VSS	ODIS1	RX LOS0	VDDIO 33	VDDIO 33	NC	AD[24]	VSS	VSS	AD[22]	VSS	VDDIO 33	VDDIO 33	VSS	VSS	IRDY/	INTA/	VSS	VSS
AC	TN	FSELA	REFPLL VSS	UART TX	ODIS0	VSS	VDDIO 33	VDDIO 33	BY PASS0/	FAULT 1/	VSS	VSS	NC	AD[25]	VDDIO 33_PCIX	VDDIO 33_PCIX	C_BE [3]/	VSS	VSS	VSS	VSS	VSS	VSS	VSS	NC	64_ EN/
AD	VSS	REFPLL VDD	VSS	VSS	VSS	VSS	VSS	VSS	LIP RESET/	LED [3]/	VDDIO 33_PCIX	VDDIO 33_PCIX	NC	AD[31]	VSS	VSS	VSS	VSS	VDDIO 33	VDDIO 33	VSS	VSS	PCIPLL VSS	VSS	VSS	VSS
AE	VSS	VSS	VSS	VSS	VSS	MOD DEF0[4]	MOD DEF1[4]	MOD DEF1[2]	LED [4]/	BLUE LED/	GPIO[1]	GPIO[5]	SWITCH	VSS	AD[30]	VSS	AD[26]	VSS	AD[18]	VSS	BZRSET	BZVDD	VSS	PCIPLL VDD	VSS	VDDIO 33
AF		VDDIO 33	VSS	VSS	MOD DEF0[3]	MOD DEF0[2]	MOD DEF1[3]	MOD DEF1[1]	LED [0]/	GPIO [0]	GPIO[4]	NC	VSS	VSS	VSS	VSS	AD[27]	VSS	VSS	AD[17]	VSS	PCICLK	VSS	VSS	VSS		

Table 7.14 Alphanumeric Pad Listing by PBGA Position

Ball #	Signal	Ball #	Signal	Ball #	Signal	Ball #	Signal	Ball #	Signal
A2	VSS	C11	VSS	E19	VSS	H1	ADSC/	M5	VDDIO33
A3	MD[11]	C12	VSS	E20	VSS	H2	ADV/	M6	VSS
A4	MD[15]	C13	MA[4]	E21	AD[47]	H3	VSS	M7	VDDIO33
A5	MD[21]	C14	AD[34]	E22	VSS	H4	VDDIO33	M12	VSS
A6	MD[22]	C15	VDDIO33	E23	AD[58]	H5	VSS	M13	VDD12
A7	MD[26]	C16	VDDIO33	E24	VSS	H6	MODE[6]	M14	VSS
A8	MD[28]	C17	VSS	E25	VSS	H7	MOE0/	M15	VDD12
A9	MA[12]	C18	VSS	E26	AD[60]	H20	VSS	M20	VSS
A10	MA[10]	C19	VSS	F1	MD[0]	H21	VSS	M21	VDDIO33_PCIX
A11	MA[6]	C20	VSS	F2	MD[2]	H22	VDDIO33_PCIX	M22	VSS
A12	MA[0]	C21	VSS	F3	MD[3]	H23	VSS	M23	VDDIO33_PCIX
A13	VSS	C22	VSS	F4	MD[7]	H24	VDDIO33_PCIX	M24	VSS
A14	VSS	C23	VSS	F5	MCLK	H25	VSS	M25	VSS
A15	VSS	C24	VSS	F6	MP[3]	H26	PAR64	M26	AD[8]
A16	VSS	C25	VSS	F7	MD[24]	J1	MODE[5]	N1	RXBVSS0
A17	AD[38]	C26	VSS	F8	MD[30]	J2	RAMCS/	N2	TXVSS0
A18	VSS	D1	MD[6]	F9	MA[19]	J3	FLASHCS/	N3	TXBVDD0
A19	VSS	D2	MD[8]	F10	MA[22]	J4	BWE1/	N4	RX0+
A20	AD[49]	D3	VSS	F11	VDDIO33	J5	BWE2/	N5	RX0-
A21	VSS	D4	VSS	F12	VDDIO33	J6	MODE[0]	N6	TX0+
A22	AD[52]	D5	MD[18]	F13	MA[7]	J7	MODE[7]	N7	TX0-
A23	AD[55]	D6	MD[14]	F14	AD[32]	J20	C_BE[4]/	N12	VDD12
A24	VSS	D7	VSS	F15	VSS	J21	REQ64/	N13	VSS
A25	VDDIO33_PCIX	D8	VSS	F16	VSS	J22	C_BE[5]/	N14	VDD12
B1	VDDIO33	D9	MA[23]	F17	AD[45]	J23	C_BE[6]/	N15	VSS
B2	VSS	D10	MA[18]	F18	AD[36]	J24	ACK64/	N20	VSS
B3	VSS	D11	VDDIO33	F19	AD[40]	J25	AD[0]	N21	AD[5]
B4	MD[12]	D12	VDDIO33	F20	VSS	J26	VSS	N22	AD[1]
B5	MD[16]	D13	MA[13]	F21	AD[51]	K1	MODE[3]	N23	AD[2]
B6	MD[20]	D14	MA[1]	F22	VSS	K2	MODE[4]	N24	VSS
B7	MD[25]	D15	VSS	F23	AD[56]	K3	NC	N25	AD[7]
B8	MD[27]	D16	VSS	F24	AD[59]	K4	MOE1/	N26	AD[9]
B9	MA[16]	D17	AD[42]	F25	VSS	K5	MWE1/	P1	RXBVDD0
B10	MA[11]	D18	VSS	F26	AD[61]	K6	BWE0/	P2	RXVSS0
B11	MA[9]	D19	VDDIO33_PCIX	G1	MP[0]	K7	MODE[1]	P3	TXBVSS1
B12	MA[5]	D20	VDDIO33_PCIX	G2	MP[1]	K20	C_BE[7]/	P4	RX1+
B13	MA[3]	D21	VSS	G3	VSS	K21	VSS	P5	RX1-
B14	VSS	D22	AD[46]	G4	VDDIO33	K22	VSS	P6	TX1-
B15	AD[35]	D23	VSS	G5	VSS	K23	VSS	P7	TX1+
B16	VSS	D24	VSS	G6	MWE0/	K24	VSS	P12	VSS
B17	AD[39]	D25	VSS	G7	MP[2]	K25	AD[3]	P13	VDD12
B18	VSS	D26	AD[57]	G8	MD[23]	K26	AD[4]	P14	VSS
B19	AD[48]	E1	MD[1]	G9	MD[29]	L1	VSS	P15	VDD12
B20	VSS	E2	MD[5]	G10	MA[20]	L2	MODE[2]	P20	VSS
B21	AD[53]	E3	MD[4]	G11	VSS	L3	MXSVDD	P21	AD[11]
B22	AD[54]	E4	BWE3/	G12	VSS	L4	VSS	P22	C_BE[0]/
B23	VSS	E5	VSS	G13	MA[8]	L5	VDDIO33	P23	VSS
B24	VSS	E6	MD[17]	G14	AD[33]	L6	VSS	P24	AD[12]
B25	VSS	E7	VDDIO33	G15	VDDIO33_PCIX	L7	VDDIO33	P25	VSS
B26	VSS	E8	VDDIO33	G16	VDDIO33_PCIX	L20	VSS	P26	VSS
C1	MD[9]	E9	MD[31]	G17	AD[37]	L21	VDDIO33_PCIX	R1	RXVDD0
C2	VSS	E10	MA[21]	G18	AD[41]	L22	VSS	R2	TXBVDD1
C3	VSS	E11	VSS	G19	AD[43]	L23	VDDIO33_PCIX	R3	VSS
C4	MD[10]	E12	VSS	G20	AD[50]	L24	VSS	R4	MXSVDD
C5	MD[13]	E13	MA[14]	G21	AD[62]	L25	VSS	R5	VSS
C6	MD[19]	E14	MA[2]	G22	VDDIO33_PCIX	L26	AD[6]	R6	VDDIO33
C7	VDDIO33	E15	VDDIO33_PCIX	G23	VSS	M1	TXVDD0	R7	VSS
C8	VDDIO33	E16	VDDIO33_PCIX	G24	VDDIO33_PCIX	M2	TXBVSS0	R12	VDD12
C9	MA[17]	E17	AD[44]	G25	AD[63]	M3	MXSVDD	R13	VSS
C10	MA[15]	E18	VSS	G26	VSS	M4	VSS	R14	VDD12

1. *NC* pins are not connected.

Table 7.14 Alphanumeric Pad Listing by PBGA Position (Cont.)

Ball #	Signal	Ball #	Signal	Ball #	Signal	Ball #	Signal	Ball #	Signal
R15	VSS	V26	VSS	AA9	GPIO[3]	AC6	VSS	AE3	VSS
R20	VDDIO33_PCIX	W1	RTCK_ICE	AA10	BYPASS1/	AC7	VDDIO33	AE4	VSS
R21	VSS	W2	TCK_ICE	AA11	VSS	AC8	VDDIO33	AE5	VSS
R22	VDDIO33_PCIX	W3	VDDIO33	AA12	VSS	AC9	BYPASS0/	AE6	MODDEF0[4]
R23	VSS	W4	VSS	AA13	NC	AC10	FAULT1/	AE7	MODDEF1[4]
R24	VDDIO33_PCIX	W5	VDDIO33	AA14	AD[29]	AC11	VSS	AE8	MODDEF1[2]
R25	AD[13]	W6	SCAN_ENABLE	AA15	VDDIO33	AC12	VSS	AE9	LED[4]/
R26	AD[10]	W7	TDO_ICE	AA16	VDDIO33	AC13	NC	AE10	BLUELED/
T1	TXVSS1	W20	STOP/	AA17	AD[21]	AC14	AD[25]	AE11	GPIO[1]
T2	TXVDD1	W21	VSS	AA18	AD[23]	AC15	VDDIO33_PCIX	AE12	GPIO[5]
T3	VSS	W22	VSS	AA19	AD[20]	AC16	VDDIO33_PCIX	AE13	SWITCH
T4	MXSVDD	W23	VDDIO33_PCIX	AA20	AD[16]	AC17	C_BE[3]/	AE14	VSS
T5	VSS	W24	VSS	AA21	GNT/	AC18	VSS	AE15	AD[30]
T6	VDDIO33	W25	VSS	AA22	FRAME/	AC19	VSS	AE16	VSS
T7	VSS	W26	VSS	AA23	ENUM/	AC20	VSS	AE17	AD[26]
T20	VDDIO33_PCIX	Y1	TRST_ICE/	AA24	INTB/	AC21	VSS	AE18	VSS
T21	VSS	Y2	TMS_ICE	AA25	VSS	AC22	VSS	AE19	AD[18]
T22	VDDIO33_PCIX	Y3	VDDIO33	AA26	RST/	AC23	VSS	AE20	VSS
T23	VSS	Y4	VSS	AB1	TRST/	AC24	VSS	AE21	BZRSET
T24	VDDIO33_PCIX	Y5	VDDIO33	AB2	IDDTN	AC25	NC	AE22	BZVDD
T25	VSS	Y6	TDI_ICE	AB3	SERIAL_DATA	AC26	64_EN/	AE23	VSS
T26	VSS	Y7	MODDEF0[0]	AB4	TDO	AD1	VSS	AE24	PCIPLLVDD
U1	RXBVSS1	Y8	FAULT0/	AB5	UARTRX	AD2	REFPLLVDD	AE25	VSS
U2	RXBVDD1	Y9	LED[2]/	AB6	MODDEF1[0]	AD3	VSS	AE26	VDDIO33
U3	RXVSS1	Y10	GPIO[2]	AB7	VSS	AD4	VSS	AF2	VDDIO33
U4	VSS	Y11	VDDIO33	AB8	VSS	AD5	VSS	AF3	VSS
U5	TDIODEVSS	Y12	VDDIO33	AB9	ODIS1	AD6	VSS	AF4	VSS
U6	TDIODEP	Y13	CPCI_EN/	AB10	RXLOS0	AD7	VSS	AF5	MODDEF0[3]
U7	REFCLKP	Y14	AD[28]	AB11	VDDIO33	AD8	VSS	AF6	MODDEF0[2]
U20	AD[15]	Y15	VSS	AB12	VDDIO33	AD9	LIPRESET/	AF7	MODDEF1[3]
U21	DEVSEL/	Y16	VSS	AB13	NC	AD10	LED[3]/	AF8	MODDEF1[1]
U22	VSS	Y17	VSS	AB14	AD[24]	AD11	VDDIO33_PCIX	AF9	LED[0]/
U23	PERR/	Y18	AD[19]	AB15	VSS	AD12	VDDIO33_PCIX	AF10	GPIO[0]
U24	VSS	Y19	C_BE[2]/	AB16	VSS	AD13	NC	AF11	GPIO[4]
U25	PAR	Y20	VSS	AB17	AD[22]	AD14	AD[31]	AF12	NC
U26	C_BE[1]/	Y21	VSS	AB18	VSS	AD15	VSS	AF13	VSS
V1	RXVDD1	Y22	VSS	AB19	VDDIO33	AD16	VSS	AF14	VSS
V2	RTRIM	Y23	VDDIO33_PCIX	AB20	VDDIO33	AD17	VSS	AF15	VSS
V3	NC	Y24	VSS	AB21	VSS	AD18	VSS	AF16	VSS
V4	REFCLKB	Y25	REQ/	AB22	VSS	AD19	VDDIO33	AF17	AD[27]
V5	SCAN_MODE	Y26	IDSEL	AB23	IRDY/	AD20	VDDIO33	AF18	VSS
V6	REFCLKN	AA1	TCK	AB24	INTA/	AD21	VSS	AF19	VSS
V7	TEST_RESET/	AA2	TMS	AB25	VSS	AD22	VSS	AF20	AD[17]
V20	VSS	AA3	TDI	AB26	VSS	AD23	PCIPLLVSS	AF21	VSS
V21	AD[14]	AA4	SERIAL_CLOCK	AC1	TN	AD24	VSS	AF22	PCICLK
V22	VSS	AA5	PROC_MON	AC2	FSELA	AD25	VSS	AF23	VSS
V23	TRDY/	AA6	MODDEF0[1]	AC3	REFPLLVSS	AD26	VSS	AF24	VSS
V24	SERR/	AA7	RXLOS1	AC4	UARTTX	AE1	VSS	AF25	VSS
V25	VSS	AA8	LED[1]/	AC5	ODIS0	AE2	VSS		

1. *NC* pins are not connected.

Table 7.15 Alphanumeric Pad Listing by Signal Name

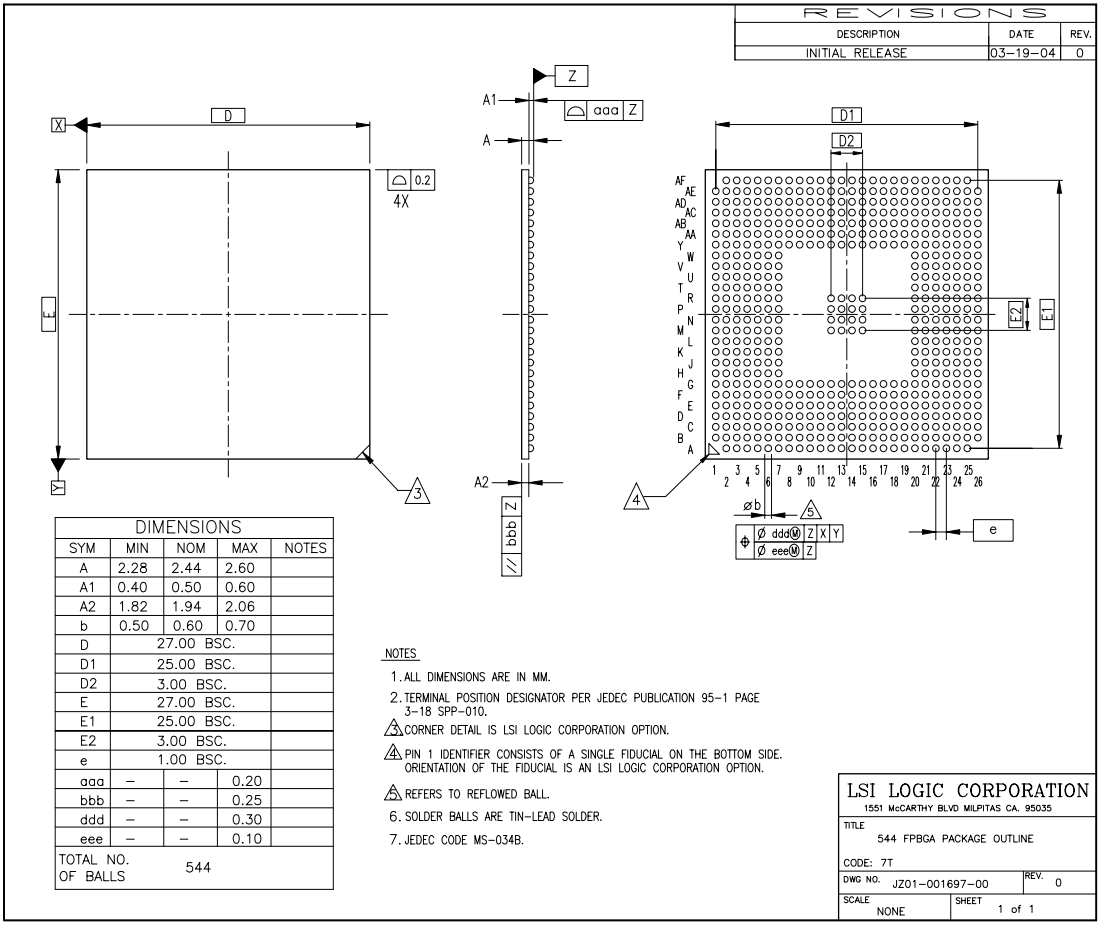
Signal	Ball #	Signal	Ball #	Signal	Ball #	Signal	Ball #	Signal	Ball #l
64_EN/	AC26	AD[56]	F23	MA[9]	B11	MODE[2]	L2	SCAN_ENABLE	W6
ACK64/	J24	AD[57]	D26	MA[10]	A10	MODE[3]	K1	SCAN_MODE	V5
ADSC/	H1	AD[58]	E23	MA[11]	B10	MODE[4]	K2	SERIAL_CLOCK	AA4
ADV/	H2	AD[59]	F24	MA[12]	A9	MODE[5]	J1	SERIAL_DATA	AB3
AD[0]	J25	AD[60]	E26	MA[13]	D13	MODE[6]	H6	SERR/	V24
AD[1]	N22	AD[61]	F26	MA[14]	E13	MODE[7]	J7	STOP/	W20
AD[2]	N23	AD[62]	G21	MA[15]	C10	MOE0/	H7	SWITCH	AE13
AD[3]	K25	AD[63]	G25	MA[16]	B9	MOE1/	K4	TCK	AA1
AD[4]	K26	BLUELED/	AE10	MA[17]	C9	MP[0]	G1	TCK_ICE	W2
AD[5]	N21	BWE0/	K6	MA[18]	D10	MP[1]	G2	TDI	AA3
AD[6]	L26	BWE1/	J4	MA[19]	F9	MP[2]	G7	TDIODEP	U6
AD[7]	N25	BWE2/	J5	MA[20]	G10	MP[3]	F6	TDIODEVSS	U5
AD[8]	M26	BWE3/	E4	MA[21]	E10	MWE0/	G6	TDI_ICE	Y6
AD[9]	N26	BYPASS0/	AC9	MA[22]	F10	MWE1/	K5	TDO	AB4
AD[10]	R26	BYPASS1/	AA10	MA[23]	D9	MXSVDD	L3	TDO_ICE	W7
AD[11]	P21	BZRSET	AE21	MCLK	F5	MXSVDD	M3	TEST_RESET/	V7
AD[12]	P24	BZVDD	AE22	MD[0]	F1	MXSVDD	R4	TMS	AA2
AD[13]	R25	CPCI_EN/	Y13	MD[1]	E1	MXSVDD	T4	TMS_ICE	Y2
AD[14]	V21	C_BE[0]/	P22	MD[2]	F2	NC	K3	TN	AC1
AD[15]	U20	C_BE[1]/	U26	MD[3]	F3	NC	V3	TRDY/	V23
AD[16]	AA20	C_BE[2]/	Y19	MD[4]	E3	NC	AA13	TRST/	AB1
AD[17]	AF20	C_BE[3]/	AC17	MD[5]	E2	NC	AB13	TRST_ICE/	Y1
AD[18]	AE19	C_BE[4]/	J20	MD[6]	D1	NC	AC13	TX0+	N6
AD[19]	Y18	C_BE[5]/	J22	MD[7]	F4	NC	AC25	TX0-	N7
AD[20]	AA19	C_BE[6]/	J23	MD[8]	D2	NC	AD13	TX1+	P7
AD[21]	AA17	C_BE[7]/	K20	MD[9]	C1	NC	AF12	TX1-	P6
AD[22]	AB17	DEVSEL/	U21	MD[10]	C4	ODIS0	AC5	TXBVDD0	N3
AD[23]	AA18	ENUM/	AA23	MD[11]	A3	ODIS1	AB9	TXBVDD1	R2
AD[24]	AB14	FAULT0/	Y8	MD[12]	B4	PAR	U25	TXBVSS0	M2
AD[25]	AC14	FAULT1/	AC10	MD[13]	C5	PAR64	H26	TXBVSS1	P3
AD[26]	AE17	FLASHCS/	J3	MD[14]	D6	PCICLK	AF22	TXVDD0	M1
AD[27]	AF17	FRAME/	AA22	MD[15]	A4	PCIPLLVDD	AE24	TXVDD1	T2
AD[28]	Y14	FSELA	AC2	MD[16]	B5	PCIPLLVSS	AD23	TXVSS0	N2
AD[29]	AA14	GNT/	AA21	MD[17]	E6	PERR/	U23	TXVSS1	T1
AD[30]	AE15	GPIO[0]	AF10	MD[18]	D5	PROC_MON	AA5	UARTRX	AB5
AD[31]	AD14	GPIO[1]	AE11	MD[19]	C6	RAMCS/	J2	UARTTX	AC4
AD[32]	F14	GPIO[2]	Y10	MD[20]	B6	REFCLKB	V4	VDD12	M13
AD[33]	G14	GPIO[3]	AA9	MD[21]	A5	REFCLKN	V6	VDD12	M15
AD[34]	C14	GPIO[4]	AF11	MD[22]	A6	REFCLKP	U7	VDD12	N12
AD[35]	B15	GPIO[5]	AE12	MD[23]	G8	REFPLLVDD	AD2	VDD12	N14
AD[36]	F18	IDDTN	AB2	MD[24]	F7	REFPLLVSS	AC3	VDD12	P13
AD[37]	G17	IDSEL	Y26	MD[25]	B7	REQ/	Y25	VDD12	P15
AD[38]	A17	INTA/	AB24	MD[26]	A7	REQ64/	J21	VDD12	R12
AD[39]	B17	INTB/	AA24	MD[27]	B8	RST/	AA26	VDD12	R14
AD[40]	F19	IRDY/	AB23	MD[28]	A8	RTCK_ICE	W1	VDDIO33	B1
AD[41]	G18	LED[0]/	AF9	MD[29]	G9	RTRIM	V2	VDDIO33	C7
AD[42]	D17	LED[1]/	AA8	MD[30]	F8	RX0+	N4	VDDIO33	C8
AD[43]	G19	LED[2]/	Y9	MD[31]	E9	RX0-	N5	VDDIO33	C15
AD[44]	E17	LED[3]/	AD10	MODDEF0[0]	Y7	RX1+	P4	VDDIO33	C16
AD[45]	F17	LED[4]/	AE9	MODDEF0[1]	AA6	RX1-	P5	VDDIO33	D11
AD[46]	D22	LIPRESET/	AD9	MODDEF0[2]	AF6	RXBVDD0	P1	VDDIO33	D12
AD[47]	E21	MA[0]	A12	MODDEF0[3]	AF5	RXBVDD1	U2	VDDIO33	E7
AD[48]	B19	MA[1]	D14	MODDEF0[4]	AE6	RXBVSS0	N1	VDDIO33	E8
AD[49]	A20	MA[2]	E14	MODDEF1[0]	AB6	RXBVSS1	U1	VDDIO33	F11
AD[50]	G20	MA[3]	B13	MODDEF1[1]	AF8	RXLOS0	AB10	VDDIO33	F12
AD[51]	F21	MA[4]	C13	MODDEF1[2]	AE8	RXLOS1	AA7	VDDIO33	G4
AD[52]	A22	MA[5]	B12	MODDEF1[3]	AF7	RXVDD0	R1	VDDIO33	H4
AD[53]	B21	MA[6]	A11	MODDEF1[4]	AE7	RXVDD1	V1	VDDIO33	L5
AD[54]	B22	MA[7]	F13	MODE[0]	J6	RXVSS0	P2	VDDIO33	L7
AD[55]	A23	MA[8]	G13	MODE[1]	K7	RXVSS1	U3	VDDIO33	M5

1. *NC* pins are not connected.

Table 7.15 Alphanumeric Pad Listing by Signal Name (Cont.)

Signal	Ball #	Signal	Ball #	Signal	Ball #	Signal	Ball #	Signal	Ball #
VDDIO33	M7	VSS	A13	VSS	E22	VSS	P26	VSS	AC11
VDDIO33	R6	VSS	A14	VSS	E24	VSS	R3	VSS	AC12
VDDIO33	T6	VSS	A15	VSS	E25	VSS	R5	VSS	AC18
VDDIO33	W3	VSS	A16	VSS	F15	VSS	R7	VSS	AC19
VDDIO33	W5	VSS	A18	VSS	F16	VSS	R13	VSS	AC20
VDDIO33	Y3	VSS	A19	VSS	F20	VSS	R15	VSS	AC21
VDDIO33	Y5	VSS	A21	VSS	F22	VSS	R21	VSS	AC22
VDDIO33	Y11	VSS	A24	VSS	F25	VSS	R23	VSS	AC23
VDDIO33	Y12	VSS	B2	VSS	G3	VSS	T3	VSS	AC24
VDDIO33	AA15	VSS	B3	VSS	G5	VSS	T5	VSS	AD1
VDDIO33	AA16	VSS	B14	VSS	G11	VSS	T7	VSS	AD3
VDDIO33	AB11	VSS	B16	VSS	G12	VSS	T21	VSS	AD4
VDDIO33	AB12	VSS	B18	VSS	G23	VSS	T23	VSS	AD5
VDDIO33	AB19	VSS	B20	VSS	G26	VSS	T25	VSS	AD6
VDDIO33	AB20	VSS	B23	VSS	H3	VSS	T26	VSS	AD7
VDDIO33	AC7	VSS	B24	VSS	H5	VSS	U4	VSS	AD8
VDDIO33	AC8	VSS	B25	VSS	H20	VSS	U22	VSS	AD15
VDDIO33	AD19	VSS	B26	VSS	H21	VSS	U24	VSS	AD16
VDDIO33	AD20	VSS	C2	VSS	H23	VSS	V20	VSS	AD17
VDDIO33	AE26	VSS	C3	VSS	H25	VSS	V22	VSS	AD18
VDDIO33	AF2	VSS	C11	VSS	J26	VSS	V25	VSS	AD21
VDDIO33_PCIX	A25	VSS	C12	VSS	K21	VSS	V26	VSS	AD22
VDDIO33_PCIX	D19	VSS	C17	VSS	K22	VSS	W4	VSS	AD24
VDDIO33_PCIX	D20	VSS	C18	VSS	K23	VSS	W21	VSS	AD25
VDDIO33_PCIX	E15	VSS	C19	VSS	K24	VSS	W22	VSS	AD26
VDDIO33_PCIX	E16	VSS	C20	VSS	L1	VSS	W24	VSS	AE1
VDDIO33_PCIX	G15	VSS	C21	VSS	L4	VSS	W25	VSS	AE2
VDDIO33_PCIX	G16	VSS	C22	VSS	L6	VSS	W26	VSS	AE3
VDDIO33_PCIX	G22	VSS	C23	VSS	L20	VSS	Y4	VSS	AE4
VDDIO33_PCIX	G24	VSS	C24	VSS	L22	VSS	Y15	VSS	AE5
VDDIO33_PCIX	H22	VSS	C25	VSS	L24	VSS	Y16	VSS	AE14
VDDIO33_PCIX	H24	VSS	C26	VSS	L25	VSS	Y17	VSS	AE16
VDDIO33_PCIX	L21	VSS	D3	VSS	M4	VSS	Y20	VSS	AE18
VDDIO33_PCIX	L23	VSS	D4	VSS	M6	VSS	Y21	VSS	AE20
VDDIO33_PCIX	M21	VSS	D7	VSS	M12	VSS	Y22	VSS	AE23
VDDIO33_PCIX	M23	VSS	D8	VSS	M14	VSS	Y24	VSS	AE25
VDDIO33_PCIX	R20	VSS	D15	VSS	M20	VSS	AA11	VSS	AF3
VDDIO33_PCIX	R22	VSS	D16	VSS	M22	VSS	AA12	VSS	AF4
VDDIO33_PCIX	R24	VSS	D18	VSS	M24	VSS	AA25	VSS	AF13
VDDIO33_PCIX	T20	VSS	D21	VSS	M25	VSS	AB7	VSS	AF14
VDDIO33_PCIX	T22	VSS	D23	VSS	N13	VSS	AB8	VSS	AF15
VDDIO33_PCIX	T24	VSS	D24	VSS	N15	VSS	AB15	VSS	AF16
VDDIO33_PCIX	W23	VSS	D25	VSS	N20	VSS	AB16	VSS	AF18
VDDIO33_PCIX	Y23	VSS	E5	VSS	N24	VSS	AB18	VSS	AF19
VDDIO33_PCIX	AC15	VSS	E11	VSS	P12	VSS	AB21	VSS	AF21
VDDIO33_PCIX	AC16	VSS	E12	VSS	P14	VSS	AB22	VSS	AF23
VDDIO33_PCIX	AD11	VSS	E18	VSS	P20	VSS	AB25	VSS	AF24
VDDIO33_PCIX	AD12	VSS	E19	VSS	P23	VSS	AB26	VSS	AF25
VSS	A2	VSS	E20	VSS	P25	VSS	AC6		

Figure 7.5 LSIFC949X 544-Pad FPBGA 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 7U.

Appendix A
Register Summary

Table A.1 and Table A.2 list the register summary for the LSIFC949X.

Table A.1 LSIFC949X Multifunction PCI Registers

Register Name	Address	Read/Write	Page
Vendor ID	0x00	Read Only	6-3
Device ID	0x02	Read Only	6-3
Command	0x04	Read/Write	6-4
Status	0x06	Read/Write	6-6
Revision ID	0x08	Read/Write	6-8
Class Code	0x09	Read/Write	6-8
Cache Line Size	0x0C	Read/Write	6-9
Latency Timer	0x0D	Read/Write	6-9
Header Type	0x0E	Read Only	6-10
Reserved	0x0F	Read Only	6-10
I/O Base Address	0x10	Read/Write	6-10
Memory[0] Base Address Low	0x14	Read/Write	6-11
Memory[0] Base Address High	0x18	Read/Write	6-11
Memory[1] Base Address Low	0x1C	Read/Write	6-12
Memory[1] Base Address High	0x20	Read/Write	6-12
Reserved	0x24–0x28	Read Only	6-12
(Sheet 1 of 3)			

Table A.1 LSIFC949X Multifunction PCI Registers (Cont.)

Register Name	Address	Read/Write	Page
Subsystem Vendor ID	0x2C	Read Only	6-13
Subsystem ID	0x2E	Read Only	6-13
Expansion ROM Base Address	0x30	Read/Write	6-14
Capabilities Pointer	0x34	Read Only	6-15
Reserved	0x38	Read Only	6-16
Interrupt Line	0x3C	Read/Write	6-16
Interrupt Pin	0x3D	Read Only	6-17
Minimum Grant	0x3E	Read Only	6-17
Minimum Latency	0x3F	Read Only	6-18
Power Management Capability ID	0xXX	Read Only	6-18
Power Management Next Pointer	0xXX	Read Only	6-20
Power Management Capabilities	0xXX	Read Only	6-19
Power Management Control/Status	0xXX	Read/Write	6-20
Power Management Bridge Support Extensions	0xXX	Read Only	6-21
Power Management Data	0xXX	Read Only	6-21
MSI Capability ID	0xXX	Read Only	6-21
MSI Next Pointer	0xXX	Read Only	6-22
MSI Message Control	0xXX	Read/Write	6-22
MSI Message Lower Address	0xXX	Read/Write	6-23
MSI Message Upper Address	0xXX	Read/Write	6-24
MSI Message Data	0xXX	Read/Write	6-24
MSI Mask Bits	0xXX	Read/Write	6-25
MSI Pending Bits	0xXX	Read Only	6-25
MSI-X Capability ID	0xXX	Read Only	6-25

(Sheet 2 of 3)

Table A.1 LSIFC949X Multifunction PCI Registers (Cont.)

Register Name	Address	Read/Write	Page
MSI-X Next Pointer	0xXX	Read Only	6-26
MSI-X Message Control	0xXX	Read/Write	6-26
MSI-X Table Offset	0xXX	Read Only	6-27
MSI-X PBA Offset	0xXX	Read Only	6-27
PCI-X Capability ID	0xXX	Read Only	6-28
PCI-X Next Pointer	0xXX	Read Only	6-28
PCI-X Command	0xXX	Read/Write	6-29
PCI-X Status	0xXX	Read/Write	6-30
(Sheet 3 of 3)			

Table A.2 LSIFC949X Host Interface Registers

Register Name	Address	Read/Write	Page
System Doorbell	0x00	Read/Write	6-34
Write Sequence	0x04	Read/Write	6-35
Host Diagnostic	0x08	Read/Write	6-36
Test Base Address	0x0C	Read/Write	6-38
Diagnostic Read/Write Data	0x10	Read/Write	6-38
Diagnostic Read/Write Address	0x14	Read/Write	6-39
Host Interrupt Status	0x30	Read Only	6-39
Host Interrupt Mask	0x34	Read/Write	6-40
Request FIFO	0x40	Read/Write	6-42
Reply FIFO	0x44	Read/Write	6-42
High Priority Request FIFO	0x48	Read/Write	6-43
Host Index Register	0x50	Read/Write	6-43

Appendix B Reference Specifications

The LSIFC949X is compliant with the specifications in Table B.1.

Table B.1 Reference Specifications

Specification	Revision
Fibre Channel Physical Interface (FC-PI-2)	5
Fibre Channel Arbitrated Loop (FC-AL-2)	7.0
FC Private Loop Direct Attach (FC-PLDA)	1.5
Fibre Channel Protocol for SCSI (FCP)	12
PCI Local Bus	2.3
PCI-X Addendum to the PCI Local Bus	2.0
PCI Bus Power Management Interface Specification	1.2

Appendix C
Glossary of Terms and Abbreviations

8B/10B	A data encoding scheme, developed by IBM, that translates byte wide data to an encoded 10-bit format.
ANSI	American National Standards Institute, the coordinating organization for voluntary standards in the United States.
Arbitrated Loop Topology (FC-AL)	A FC topology that provides a low-cost solution to attach multiple ports in a loop without switches.
BER	Bit error rate.
Bit	A binary digit. The smallest unit of information a computer uses. The value of a bit (0 or 1) represents a two-way choice, such as on or off, and true or false.
Broadcast	Sending a transmission to all N_Ports on a Fabric.
Bus	A collection of unbroken signal lines across which information is transmitted from one part of a computer system to another. Connections to the bus are made using taps on the lines.
Bus Mastering	A high-performance way to transfer data. The host adapter controls the transfer of data directly to and from system memory without bothering the computer's microprocessor. This is the fastest way for multitasking operating systems to transfer data.
Byte	A unit of information consisting of 8 bits.
Channel	A point-to-point link, the main task of which is to transport data from one point to another.

Configuration	Refers to the way a computer is set up; the combined hardware components (computer, monitor, keyboard, and peripheral devices) that make up a computer system; or the software settings that allow the hardware components to communicate with each other.
CPU	Central processing unit. The “brain” of the computer that performs the actual computations. The term *microprocessor* unit (MPU) is also used.
Crosspoint-Switched Topology (FC-XS)	Highest performance FC Fabric, providing a choice of multiple path routings between pairs of F_Ports.
DMA	Direct memory access. A method of moving data from a storage device directly to RAM without using the resources of the CPU.
DMA Bus Master	A feature that allows a peripheral to control the flow of data to and from system memory by blocks, as opposed to PIO (programmed I/O), where the processor is in control and the flow is by byte.
Device Driver	A program that allows a microprocessor (through the operating system) to direct the operation of a peripheral device.
EEPROM	Electronically Erasable Programmable Read Only Memory. A memory chip that typically stores configuration information.
EISA	Extended Industry Standard Architecture. An extension of the 16-bit ISA bus standard. It allows devices to perform 32-bit data transfers.
Exchange	A term that refers to one of the FC “building blocks,” composed of one or more nonconcurrent sequences for a single operation.
Fabric	FC-defined interconnection methodology that handles routing in FC networks.
FC	Fibre Channel.
FC-PH	FC physical standard, consisting of the three lower levels: FC-0, FC-1, and FC-2.
FC-0	Lowest level of FC-PH, covering the physical characteristics of the interface and media.
FC-1	Middle level of FC-PH, defining the 8B/10B encoding/decoding and transmission protocol.

FC-2	Highest level of FC-PH, defining the rules for signaling protocol and describing transfer of the frame, sequence, and exchanges.
FC-3	The hierarchical level in the FC standard that provides common services, such as striping definition.
FC-4	The hierarchical level in the FC standard that specifies the mapping of Upper Layer Protocols (ULPs) to levels below.
FCC	Federal Communications Commission.
FCP	Fibre Channel Protocol.
FDDI	Fiber Distributed Data Interface. The ANSI option for a Metropolitan Area Network (MAN); a network based on the use of optical fiber cable to transmit data at 100 Mbits/s.
Fibre Channel Service Protocol (FSP)	The common FC-4 level protocol for all services, transparent to the Fabric type or topology.
File	A named collection of information stored on a disk.
Firmware	Software that is permanently stored in ROM. Therefore, it can be accessed during boot time.
F_Port	A Fabric port, the access point of the fabric for physically connecting the N_Port.
FL_Port	A Fabric port configured for loop functionality.
Frame	A linear set of transmitted bits that define a basic transport element.
Hard Disk	A disk made of metal and permanently sealed into a drive cartridge. A hard disk can store very large amounts of information.
HAL	Hardware Abstraction Layer.
HIPPI	High Performance Parallel Interface, an 800 Mbit/s interface to supercomputer networks (formerly known as high speed channel) developed by ANSI.
Host	The computer system in which a SCSI host adapter is installed. It uses the SCSI host adapter to transfer information to and from devices attached to the SCSI bus.

Host Adapter	A circuit board or integrated circuit that provides a SCSI bus connection to the computer system.
IOP	I/O processor.
IP	Internet Protocol.
IPI	Intelligent Peripheral Interface.
ISA	Industry Standard Architecture. A type of computer bus used in most PCs. It allows devices to send and receive data up to 16 bits at a time.
Kbyte	Kilobyte. A measure of computer storage equal to 1024 bytes.
LCT	Logical Configuration Table.
Link_Control_ Facility	A termination card that handles the logical and physical control of the FC link for each mode of use.
LLC	Logical link control.
Local Bus	A way to connect peripherals directly to computer memory. It bypasses the slower ISA and EISA buses. PCI is a local bus standard.
Login Server	Entity within the FC fabric that receives and responds to login requests.
L_Port	An FC port which supports the Arbitrated Loop topology.
LUN	Logical Unit Number. An identifier, zero to seven, for a logical unit.
Mbyte	Megabyte. A measure of computer storage equal to 1024 kilobytes.
MFA	Message Frame Address.
MSI	Message Signaled interrupt.
Multicast	Refers to delivering a single transmission to multiple destination N_Ports.
NIC	Network interface card.
N_Port	A Node port, an FC-defined hardware entity at the node end of a link.
NL_Port	A Node port configured for loop functionality.

Operating System — A program that organizes the internal activities of the computer and its peripheral devices. An operating system performs basic tasks such as moving data to and from devices, and managing information in memory. It also provides the user interface.

Operation — A term, defined in FC-2, that refers to one of the FC building blocks composed of one or more, possibly concurrent, exchanges.

Ordered Set — An FC term referring to four 10-bit characters (a combination of data and special characters) that provide low level link functions, such as frame demarcation and signaling between two ends of a link. It provides for initialization of the link after power-on and for some basic recovery actions.

Originator — An FC term referring to the initiating device.

Parity Checking — A way to verify the accuracy of data transmitted over the SCSI bus. One bit in the transfer makes the sum of all the 1 bits either odd or even (for odd or even parity). If the sum is not correct, an error message appears.

PCI — Peripheral Component Interconnect. A local bus specification that allows connection of peripherals directly to computer memory. It bypasses the slower ISA and EISA buses.

PDB — Packet Descriptor Block.

PIO — Programmed Input/Output. A way the CPU can transfer data to and from memory using the computer I/O ports. PIO is usually faster than DMA, but requires CPU time.

Port — The hardware entity within a node that performs data communications over the FC link.

Port Address — Also Port Number. The address through which commands are sent to a host adapter board. This address is assigned by the PCI bus.

Port Number — See Port Address.

RAM — Random Access Memory. The primary working memory of the computer in which program instructions and data are stored and are accessible to the CPU. Information can be written to and read from RAM. The contents of RAM are lost when the computer is turned off.

Responder — An FC term referring to the answering device.

RISC Core LSIFC949X chips contain a RISC (Reduced Instruction Set Computer) processor, programmed through microcode scripts.

ROM Read Only Memory. Memory from which information can be read but not changed. The contents of ROM are not erased when the computer is turned off.

SAN Storage Area Network.

SCAM SCSI Configuration Automatically. A method that automatically allocates SCSI IDs using software when SCAM compliant SCSI devices are attached.

Scatter/Gather A device driver feature that lets the host adapter modify a transfer data pointer so that a single host adapter transfer can access many segments of memory. This minimizes interrupts and transfer overhead.

SCB SCSI Command Block.

SCSI Small Computer System Interface. A specification for a high-performance peripheral bus and command set. The original standard is referred to as SCSI-1.

SCSI-2 The current SCSI specification, which adds features to the original SCSI-1 standard.

SCSI ID A way that uniquely identifies each SCSI device on the SCSI bus. Each SCSI bus has eight available SCSI IDs numbered 0–7 (or 0–15 for Wide SCSI). The host adapter usually gets ID 7, giving it priority to control the bus.

Sequence A term referring to one of the FC building blocks, which are composed of one or more related frames for a single operation.

SFF small form factor.

SGL Scatter-gather list.

SNAP Subnetwork Access Protocol.

Synchronous Data Transfer One of the ways data is transferred over the SCSI bus. Transfers are clocked with fixed frequency pulses. This is faster than asynchronous data transfer. Synchronous data transfers are negotiated between the SCSI host adapter and each SCSI device.

System BIOS	Controls the low level POST (Power-On Self Test), and basic operation of the CPU and computer system.
TID	Target ID.
Topology	The logical and/or physical arrangement of stations on a network.
ULP	Upper Layer Protocol.
VCCI	Voluntary Control Council for Interference.
Virtual Memory	Space on a hard disk that can be used as if it were RAM.
VPD	Vendor Product Data.
Word	A 2-byte (or 16-bit) unit of information.
X3T9	A technical committee of the Accredited Standards Committee X3, titled X3T9 I/O Interfaces. It develops standards for moving data in and out of central computers.

Index

C

D

E

F

G

H

R

U

V

W

Customer Feedback

We would appreciate your feedback on this document. Please copy the following page, add your comments, and fax it to us at the number shown.

If appropriate, please also fax copies of any marked-up pages from this document.

Important: Please include your name, phone number, fax number, and company address so that we may contact you directly for clarification or additional information.

Thank you for your help in improving the quality of our documents.

Reader's Comments

Fax your comments to: LSI Logic Corporation
Technical Publications
M/S AF-198
Fax: 408.433.4333

Please tell us how you rate this document: *LSIFC949X Dual Channel Fibre Channel 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 _____