TECHNICAL MANUAL

LSIFC929XL
Dual Channel
Fibre Channel
I/O Processor

October 2005

Version 2.0

LSI LOGIC®

DB14-000272-01

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

Document DB14-000272-01, October 2005
This document describes the LSI Logic Corporation LSIFC929XL 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Channel I/O Processor and will remain the official reference source for all revisions/releases of this product until rescinded by an update.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, and GigaBlaze are trademarks or registered trademarks of LSI Logic Corporation. ARM is a registered trademark and ARM966E-S is a 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 LSIFC929XL Dual Channel Fibre Channel I/O Processor. It contains a complete functional description for the LSIFC929XL 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 LSIFC929XL and to explain how to use the LSIFC929XL 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 LSIFC929XL for possible use in a system
- Engineers who are designing the LSIFC929XL into a system

Organization

This document has the following chapters and appendixes:

- Chapter 1, **Introduction**, provides a general description of the LSIFC929XL.
- Chapter 2, **Fibre Channel Overview**, briefly describes some key elements of Fibre Channel, including layers, topologies, and classes of service.

- Chapter 3, **LSIFC929XL Overview**, provides an introduction to the basic features of the LSIFC929XL, including the message interface, protocol assist engines, and support components.
- Chapter 4, **Signal Descriptions**, lists and describes the signals on the LSIFC929XL.
- Chapter 5, **PCI-X Functional Description**, describes the PCI-X features contained in the LSIFC929XL.
- 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 LSIFC929XL, 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, *Volume 1.2,* Document No. DB14-000174-02

PCI Local Bus Specification, Version 2.2

PCI-X Addendum to the PCI Local Bus Specification, Version 1.0a

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 "/."

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

Revision Record

Revision	Date	Remarks
1.0	12/2003	First preliminary release.
2.0	10/2005	Final release.

Contents

Chapter 5
PCI-X Functional Description

Chapter 6
Registers

Figures

Tables

Chapter 1
Introduction

This chapter provides an overview of the LSIFC929XL Dual Channel Fibre Channel I/O Processor. The chapter contains the following sections:

1.1 Overview

The LSIFC929XL 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 supported with Fibre Channel Protocol (FCP) for SCSI.

1.1.1 Hardware Features

The LSIFC929XL supports the following list of hardware features:

- Highly integrated, full duplex, Dual Channel FC I/O processor
- Integrated 2 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)
- Integrated bit error rate (BER) link testing
- 32-bit ARM® RISC processor

- Intelligent, high-performance context management
- 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
  - 1000 concurrent commands with 1 Mbyte SRAM (default)
  - 2000 concurrent commands with 2 Mbytes SRAM
  - 4000 concurrent commands with 4 Mbytes SRAM
- PC2001 compliant
- Peripheral Component Interface (PCI) 2.2 compliant
- JTAG debug interface
- 456-pin plastic ball grid array (PBGA)

1.1.2 FC Features

The LSIFC929XL supports the following list of FC features:

- Class 2 and Class 3 support (with optional confirmed delivery)
- BB credit of 16, alternate login of 1 (each channel)
- 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 and 2Gbit/s links (independent for each channel)

1.1.3 Software Features

The LSIFC929XL 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

1.1.4 OS Support

The LSIFC929XL supports the following list of operating systems:

- Windows 2000
- Windows NT 4.0 SP4 and Windows NT 5.0
- Windows XP
- NetWare 4.11 and 5.0
- UnixWare 2.12 and Gemini
- Solaris 2.6, 2.7–X86
- Solaris SPARC
- Linux susi, Turbolinux, and Red Hat Linux

1.1.5 Targeted Applications

The LSIFC929XL targets the following list of key applications:

- SANs
- Storage virtualization
- Server clustering environments
- Embedded RAID
- Low cost PCI-X/FC host adapters
- Host main boards
- Routers and bridges

1.2 General Description

The LSIFC929XL 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, 2 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.

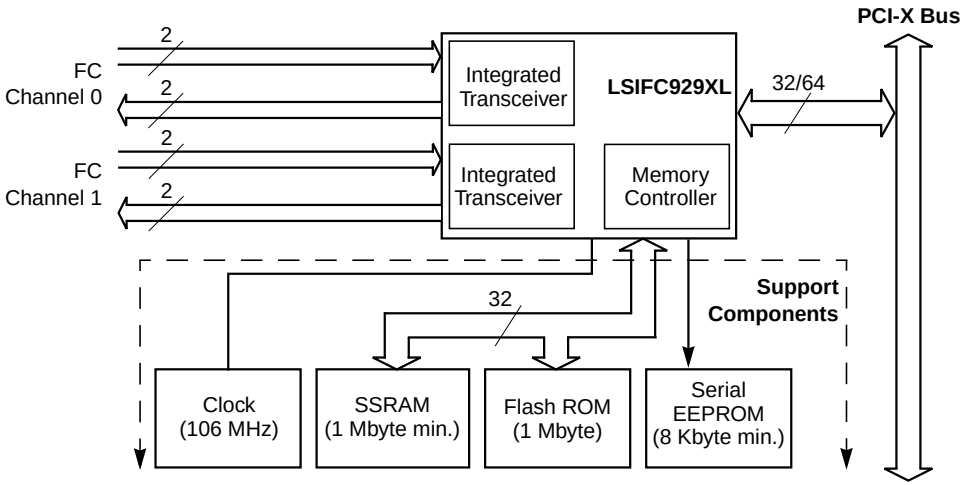
From the host CPU perspective, the LSIFC929XL manages the FC link at the exchange level for mass storage (FCP) protocols. The LSIFC929XL supports multiple I/O requests per host interrupt in most applications.

From the FC link perspective, the LSIFC929XL is a highly efficient NL_Port supporting point-to-point, public and private loop topologies, and the FC switch/attach topology defined under the ANSI X3T11 FC-FS standard. The LSIFC929XL 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 LSIFC929XL 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 LSIFC929XL controller enables low cost FC implementations. Figure 1.1 shows a typical implementation incorporating the LSIFC929XL controller.

Figure 1.1 LSIFC929XL Typical Implementation



1.2.1 Multifunction PCI-X

Coupled with the dual channel operation, the LSIFC929XL 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 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 LSIFC929XL performs this operation to determine whether a point-to-point device or all of the devices on a link are either 1 Gbit/s or 2 Gbit/s devices, and it automatically configures itself to be compatible with the devices on the link.

1.2.3 Autotopology Negotiation

The LSIFC929XL maintains compatibility with private loop, public loop, and point-to-point topologies through Autotopology Negotiation. The LSIFC929XL performs this operation to determine the type of attached link, and automatically configures each LSIFC929XL port to the current port type.

1.2.4 Failover and Load Balancing

The LSIFC929XL 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 LSIFC929XL architecture allows the OS driver to support automatic failover without the need for LSIFC929XL intervention. Load Balancing also can be provided in the host driver to partition the I/O workload across each channel of the LSIFC929XL.

1.2.5 Single Channel FC Support

To use the LSIFC929XL as a single-channel FC device, assert the SingleFunctionPCI bit in the Hardware Configuration Settings field in SEEPROM (see Section 3.8 of SEN #11082, “LSIFC929XL Design Considerations”). With this bit asserted, FC Channel 1 is disabled, and FC Channel 0 remains active.

1.3 Hardware Overview

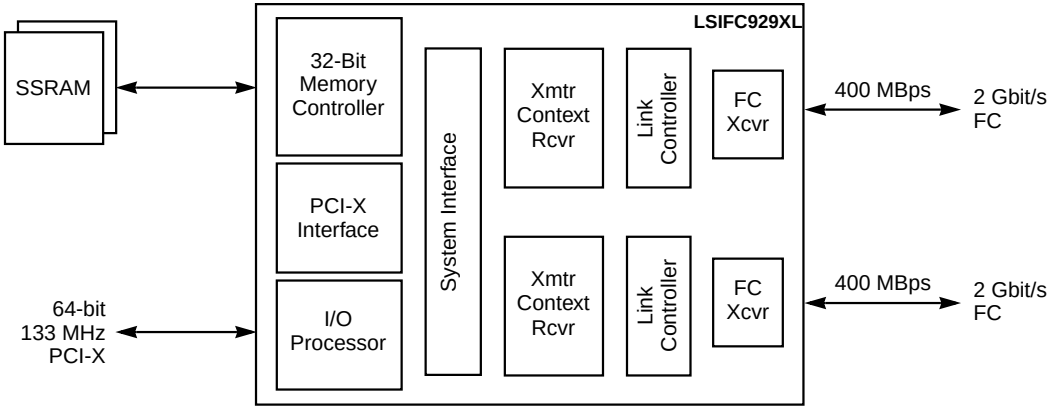
In today’s fast growing SAN, storage virtualization, server/workstation, and RAID storage systems marketplaces, higher levels of performance, scalability, and reliability are required to stay competitive.

The LSIFC929XL provides the performance and flexibility to meet future FC connectivity requirements.

The LSIFC929XL 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 LSIFC929XL provides the highest level of performance in a more cost effective FC solution.

Figure 1.2 shows the functional block diagram for the LSIFC929XL. 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 LSIFC929XL Functional Block Diagram



1.3.1 PCI/PCI-X Interface

The LSIFC929XL 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).

1.3.2 32-Bit Memory Controller

The memory controller provides access to Flash ROM and 32-bit Synchronous SRAM and nonvolatile SRAM (NVSRAM). 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 LSIFC929XL 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 LSIFC929XL and other I/O agents. It consists of four hardware FIFOs for the message queuing lists: Request Free, Request Post, Reply Free, and Reply Post. Control logic for the FIFOs is provided within the LSIFC929XL system interface with messages stored in external memory.

1.3.5 Integrated 2 Gbit/s Transceivers

The LSIFC929XL implements GigaBlaze® 2 Gbit/s transceivers. GigaBlaze transceivers are backward-compatible with 1 Gbit/s systems, using a firmware-implemented “Autospeed Negotiation” for automatic compatibility between 1 Gbit/s and 2 Gbit/s links. The integrated 2 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 two 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 LSIFC929XL 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 LSIFC929XL autonomously handles FCP exchanges upon request from the host. The LSIFC929XL 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 LSIFC929XL provides for general purpose target functions such as those required for front-end RAID applications.

1.6 Diagnostics

The LSIFC929XL 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

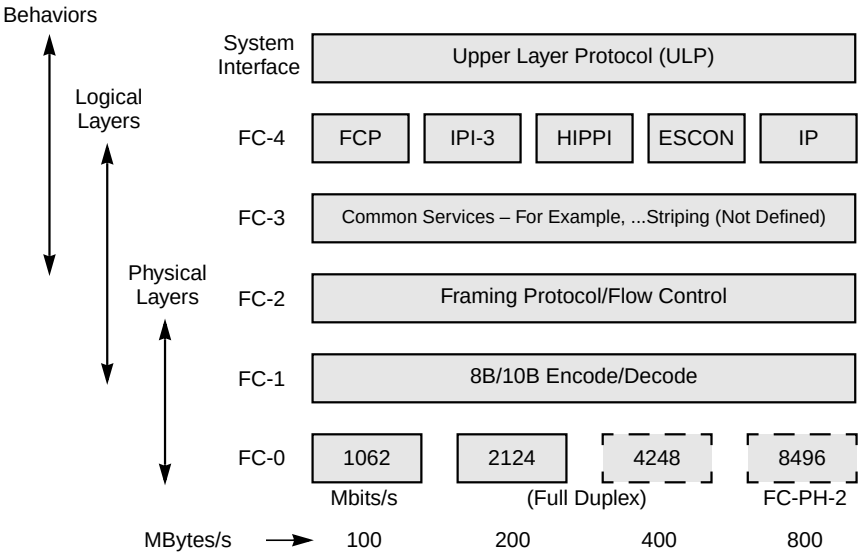
This chapter provides general overview information on Fibre Channel (FC). The chapter contains the following sections:

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.

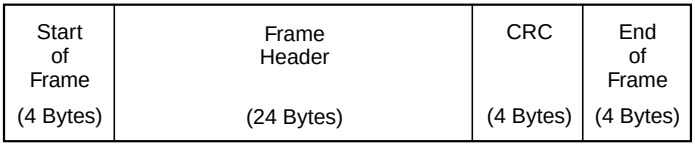
The FC-3 layer provides common services that span multiple N_Ports (refer to Section 2.5, “FC Ports,” on page 2-7 for details). Some of these services include Striping, Hunt Groups, and Multicasting. All of these services allow a single port or fabric to communicate to several N_Ports at one time.

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

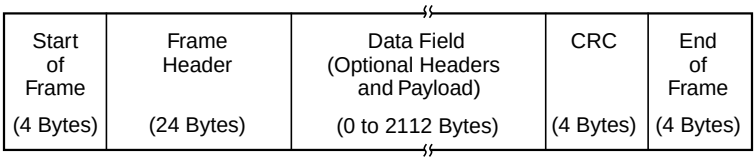
There are two types of frames used in FC: Link Control frames and Data frames. Link Control frames, which contain no payload, are flow control responses to Data frames. An example of a Link Control frame is the ACK frame.

Figure 2.2 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 Data Frame

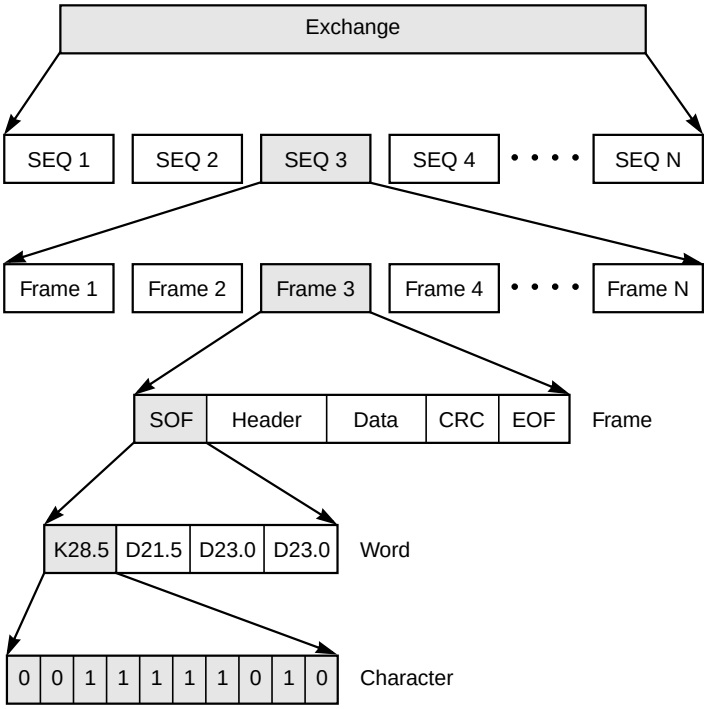


In FC, an Ordered Set is a group of four 10-bit characters that provide low level link functions, such as frame demarcation and signaling between two ends of a link. All frames start with a Start-of-Frame (SOF) and end with an End-of-Frame (EOF) Ordered Set. Each frame contains at least a 24-byte header 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

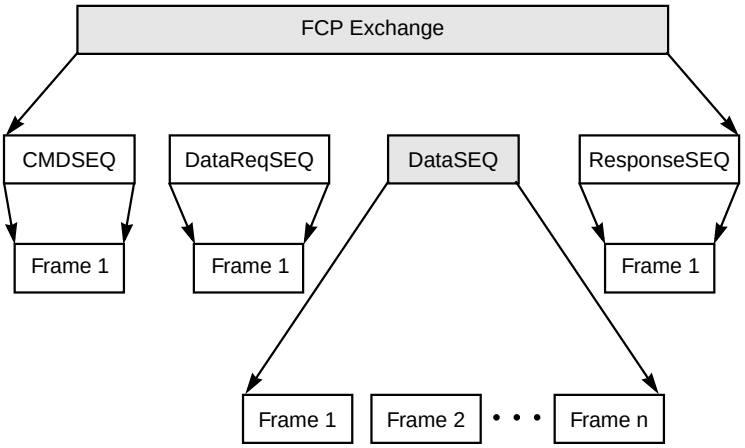
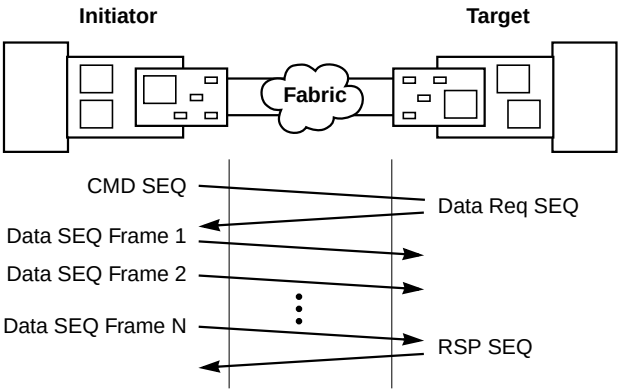


Figure 2.6 shows how the Exchange flows between the Initiator and Target. The Initiator starts the FCP Exchange by sending a Command Sequence containing one frame to the Target. The Frame payload contains the Command Descriptor Block (CDB). The Target then responds with a Data Delivery Request Sequence 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:

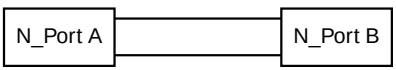
- Point-to-Point topology
- Fabric topology
- Arbitrated Loop topology

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.

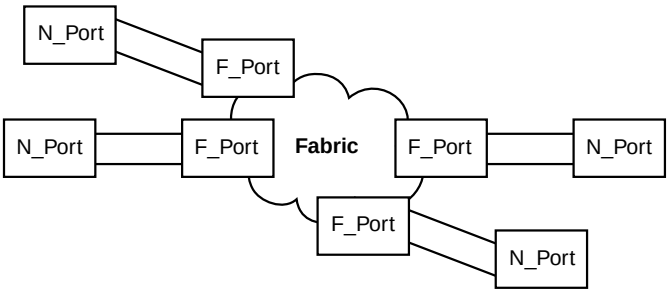
Figure 2.7 Point-to-Point Topology



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

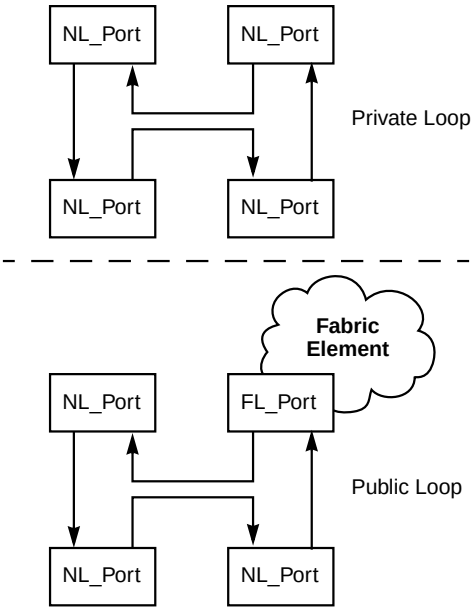


2.6.3 Arbitrated Loop Topology

The Arbitrated Loop topology permits 2–127 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
LSIFC929XL Overview

This chapter provides a general description of the LSIFC929XL Dual Channel Fibre Channel I/O processor firmware. The chapter contains the following sections:

3.1 Introduction

The LSI Logic LSIFC929XL 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 LSIFC929XL are listed in Appendix B, “Reference Specifications.”

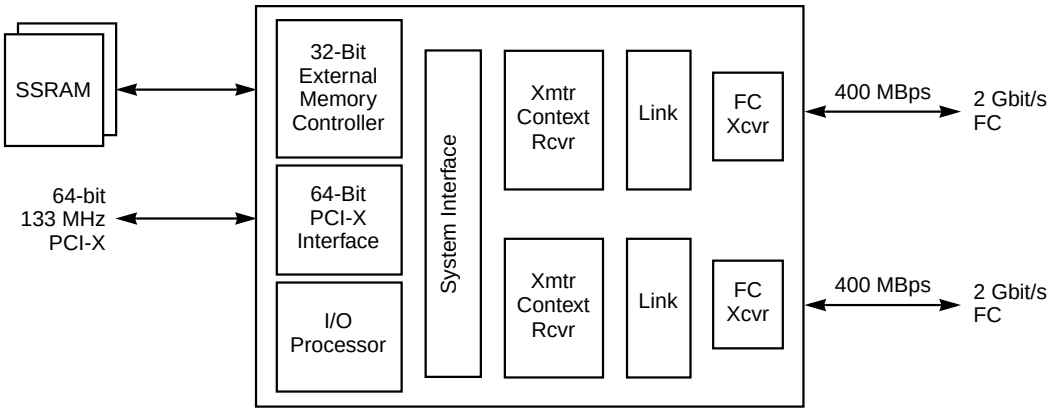
Although optimized for a 64-bit PCI-X interface to communicate with the system CPU(s) and memory, the LSIFC929XL also supports a 32-bit Peripheral Component Interface (PCI) environment. The system interface to the LSIFC929XL minimizes the amount of PCI-X bandwidth required to support I/O requests. A 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 LSIFC929XL architecture allows the system to specify I/Os at the command level. The LSIFC929XL manages I/Os at the Frame, Sequence and Exchange level. Error detection and I/O retries are

also handled by the LSIFC929XL, allowing the system to offload part of the exception handling work from the system driver.

Data Flows – The LSIFC929XL 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 LSIFC929XL 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 LSIFC929XL 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 LSIFC929XL system interface is a high-performance, packetized, mailbox architecture that leverages the intelligence in the LSIFC929XL to minimize traffic on the PCI-X bus.

There are two basic constructs in the Message Interface. The first construct, the Message, communicates between the system and the LSIFC929XL. Messages are moved between the system(s) and the LSIFC929XL using the second construct, a Transport mechanism.

3.2.1 Messages

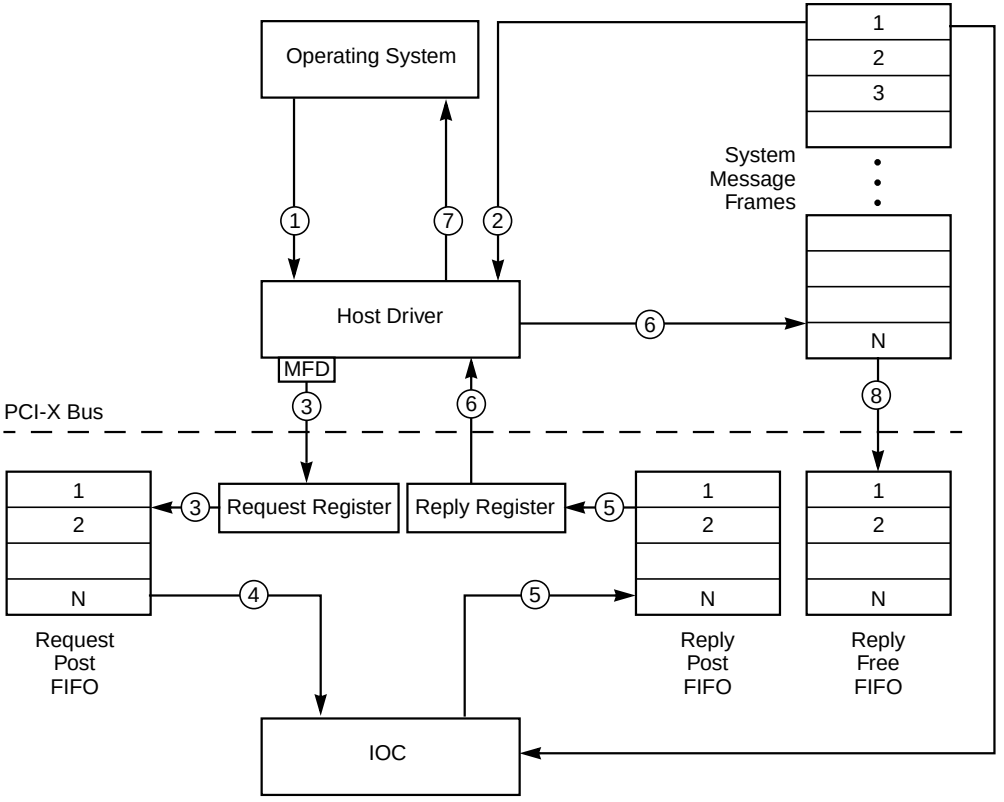
The LSIFC929XL uses two types of messages to communicate with the system. Request messages are created by the system to “request” an action by the LSIFC929XL. Reply messages are used by the LSIFC929XL to send status information back to the system. Request message data structures are up to 128 bytes in 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*.

3.2.2 Message Flow

Before Requests can be posted to the LSIFC929XL, 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 LSIFC929XL, 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 LSIFC929XL 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 LSIFC929XL. 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 LSIFC929XL 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*.

3.4 Target Message

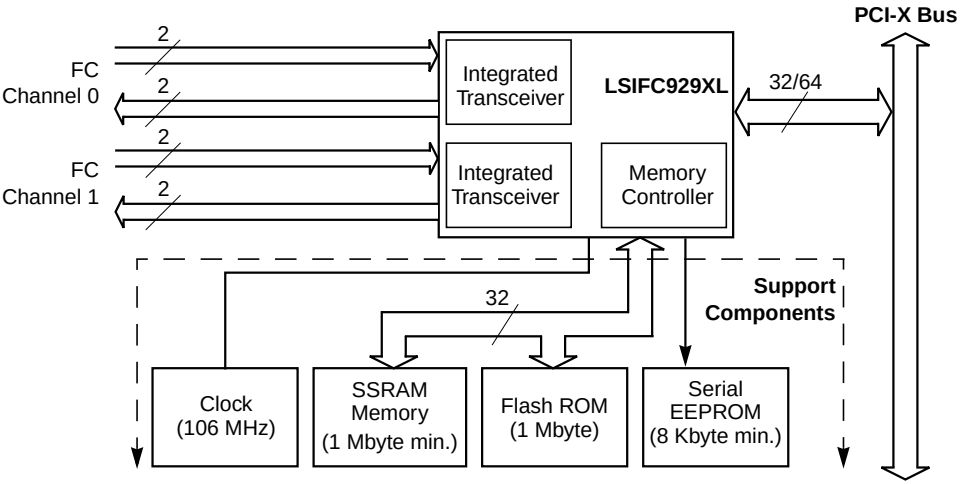
The Target Interface allows the LSIFC929XL to be used as the system interface for FC bridge controllers. The LSIFC929XL provides an FCP exchange level message interface that routes commands to the system. The system identifies the appropriate data, and passes a Scatter Gather List (SGL) to the LSIFC929XL describing the data to transfer. A single Target message directs the LSIFC929XL 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* for details on the Target Message Class.

3.5 Support Components

The memory controller block within the LSIFC929XL 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 LSIFC929XL. Figure 3.3 shows an LSIFC929XL typical implementation diagram.

Figure 3.3 LSIFC929XL Typical Implementation



3.5.1 SSRAM Memory

The primary function of this memory is to store data structures used by the LSIFC929XL to manage exchanges and transmit and receive queues. The SSRAM memory also stores part of the run time image of the LSIFC929XL firmware, such as initialization and error recovery code. The mainline code is stored within the internal LRAM for performance reasons.

The LSIFC929XL 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 LSIFC929XL 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 LSIFC929XL.

3.5.2 Flash ROM

The memory controller in the LSIFC929XL also manages an optional Flash ROM. If present, the Flash ROM stores the firmware for the LSIFC929XL, 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 LSIFC929XL through the PCI-X interface. The LSIFC929XL supports a simple register handshake interface for firmware download. Firmware may be directly written to the LSIFC929XL internal memory and external SSRAM through this interface. Details of this implementation are available in the *Fusion-MPT Message Passing Interface Specification*. Flash ROM is optional for the LSIFC929XL, 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* for details on the programming of the Flash ROM.

3.5.3 Serial EEPROM

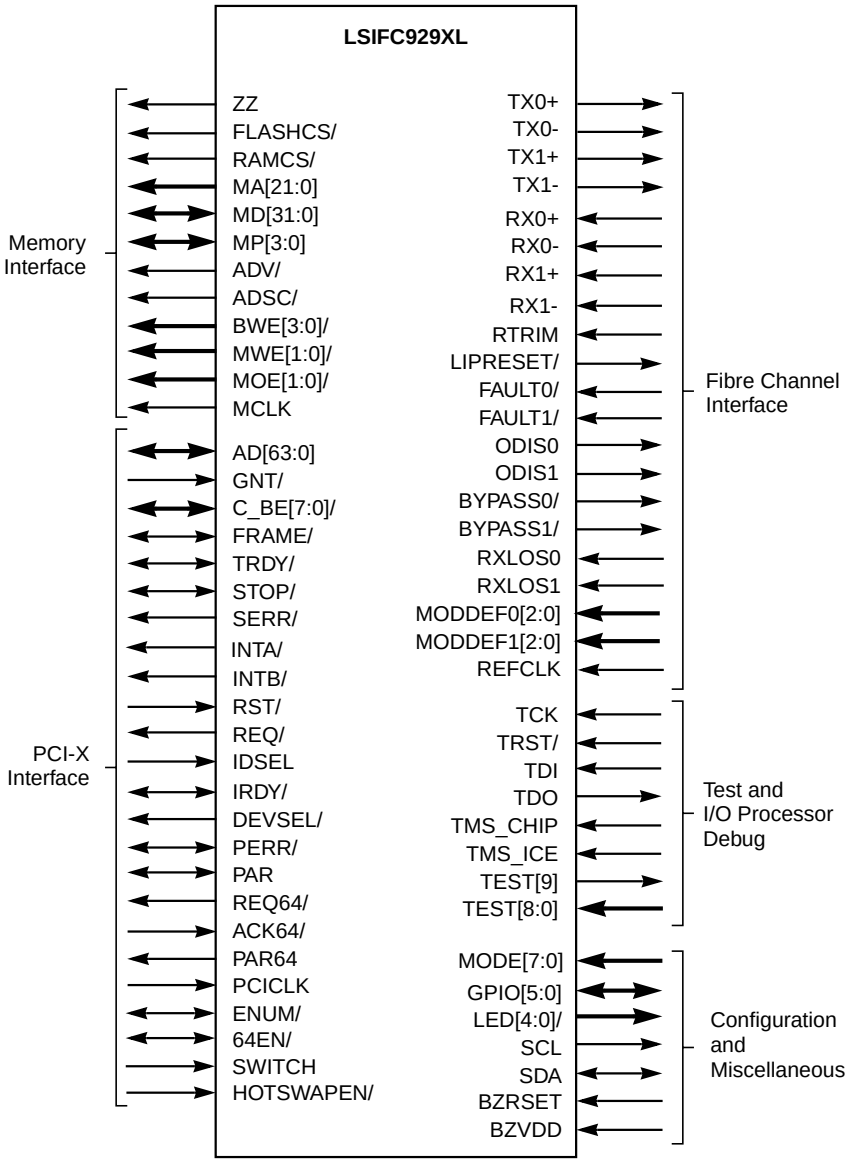
The serial EEPROM stores nonvolatile data for the LSIFC929XL, such as the World Wide Name, VPD, and other vendor-specific information. The SEEPROM data is programmed by the firmware, so the firmware must be downloaded and running before the SEEPROM is programmed. The required minimum size of the SEEPROM is 8 Kbytes.

Chapter 4
Signal Descriptions

This chapter contains signal descriptions for the LSIFC929XL. 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 LSIFC929XL 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	V1	5 V Tol In	**Clock**. Refer to the *PCI Local Bus Specification, Version 2.2*, and the *PCI-X Addendum to the PCI Local Bus Specification, Version 1.0a*, for this signal description.
RST/	I	U3	5 V Tol In	**Reset**. Refer to the *PCI Local Bus Specification, Version 2.2,* and the *PCI-X Addendum to the PCI Local Bus Specification, Version 1.0a*, for this signal description.
GNT/	I/O	U4	5 V Tol BiDir PCI	**Grant**. Refer to the *PCI Local Bus Specification, Version 2.2,* and the *PCI-X Addendum to the PCI Local Bus Specification, Version 1.0a*, for this signal description.
REQ/	I/O	V2	5 V Tol BiDir PCI	**Request**. Refer to the *PCI Local Bus Specification, Version 2.2,* and the *PCI-X Addendum to the PCI Local Bus Specification, Version 1.0a*, for this signal description.
REQ64/	I/O	AD14	5 V Tol BiDir PCI	**Request64**. Refer to the *PCI Local Bus Specification, Version 2.2,* and the *PCI-X Addendum to the PCI Local Bus Specification, Version 1.0a*, for this signal description.
ACK64/	S/T/S	AE14	5 V Tol BiDir PCI	**Acknowledge64**. Refer to the *PCI Local Bus Specification, Version 2.2,* and the *PCI-X Addendum to the PCI Local Bus Specification, Version 1.0a*, for this signal description.

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

Signal	I/O	BGA Pad No.	Pad Type	Description
AD[63:0]	T/S	AE16, AE17, AD17, AC17, AF18, AE18, AD18, AF19, AC19, AE19, AF20, AE20, AC20, AF21, AE21, AD21, AF22, AC21, AE22, AD22, AF23, AE23, AE24, AE25, AD25, AD26, AD24, AC25, AA26, AB24, AA23, Y23, V3, W1, W2, W5, Y1, Y2, Y4, AB1, AA4, W4, AB2, AC1, AC3, AD2, AF6, AE3, AC8, AE8, AF8, AD9, AE9, AF10, AE10, AC10, AC11, AE12, AF13, AC12, AE13, AD13, AC15, AC13	5 V Tol BiDir PCI	**Address and Data**. Refer to the *PCI Local Bus Specification, Version 2.2,* and the *PCI-X Addendum to the PCI Local Bus Specification, Version 1.0a*, for this signal description.
C_BE[7:0]/	T/S	AC14, AF15, AE15, AF14, AA2, AE4, AF7, AD10	5 V Tol BiDir PCI	**Command and Byte Enables**. Refer to the *PCI Local Bus Specification, Version 2.2*, and the *PCI-X Addendum to the PCI Local Bus Specification, Version 1.0a*, for this signal description.
IDSEL	I/O	AA3	5 V Tol BiDir PCI	**Initialization Device Select**. Refer to the *PCI Local Bus Specification, Version 2.2,* and the *PCI-X Addendum to the PCI Local Bus Specification, Version 1.0a*, for this signal description.
FRAME/	S/T/S	AF3	5 V Tol BiDir PCI	**Cycle Frame**. Refer to the *PCI Local Bus Specification, Version 2.2,* and the *PCI-X Addendum to the PCI Local Bus Specification, Version 1.0a*, for this signal description.

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

Signal	I/O	BGA Pad No.	Pad Type	Description
IRDY/	S/T/S	AE5	5 V Tol BiDir PCI	**Initiator Ready**. Refer to the *PCI Local Bus Specification, Version 2.2,* and the *PCI-X Addendum to the PCI Local Bus Specification, Version 1.0a*, for this signal description.
TRDY/	S/T/S	AF4	5 V Tol BiDir PCI	**Target Ready**. Refer to the *PCI Local Bus Specification, Version 2.2*, and the *PCI-X Addendum to the PCI Local Bus Specification, Version 1.0a*, for this signal description.
DEVSEL/	I/O	AC6	5 V Tol BiDir PCI	**Device Select**. Refer to the *PCI Local Bus Specification, Version 2.2,* and the *PCI-X Addendum to the PCI Local Bus Specification, Version 1.0a*, for this signal description.
STOP/	S/T/S	AD6	5 V Tol BiDir PCI	**Stop**. Refer to the *PCI Local Bus Specification, Version 2.2,* and the *PCI-X Addendum to the PCI Local Bus Specification, Version 1.0a*, for this signal description.
PERR/	S/T/S	AE6	5 V Tol BiDir PCI	**Parity Error**. Refer to the *PCI Local Bus Specification, Version 2.2,* and the *PCI-X Addendum to the PCI Local Bus Specification, Version 1.0a*, for this signal description.
SERR/	I/O	AF5	5 V Tol BiDir PCI	**System Error**. Refer to the *PCI Local Bus Specification, Version 2.2,* and the *PCI-X Addendum to the PCI Local Bus Specification, Version 1.0a*, for this signal description.
PAR	T/S	AE7	5 V Tol BiDir PCI	**Parity**. Refer to the *PCI Local Bus Specification, Version 2.2,* and the *PCI-X Addendum to the PCI Local Bus Specification, Version 1.0a*, for this signal description.
PAR64	I/O	AF16	5 V Tol BiDir PCI	**Parity64**. Refer to the *PCI Local Bus Specification, Version 2.2*, and the *PCI-X Addendum to the PCI Local Bus Specification, Version 1.0a*, for this signal description.
INTA/	I/O	T4	5 V Tol BiDir PCI	**Interrupt A**. Refer to the *PCI Local Bus Specification, Version 2.2*, and the *PCI-X Addendum to the PCI Local Bus Specification, Version 1.0a*, for this signal description.

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

Signal	I/O	BGA Pad No.	Pad Type	Description
INTB/	I/O	T2	5 V Tol BiDir PCI	**Interrupt B**. Refer to the *PCI Local Bus Specification, Version 2.2*, and the *PCI-X Addendum to the PCI Local Bus Specification, Version 1.0a*, for this signal description.
ENUM/	I/O	P4	5 V Tol 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	P1	5 V Tol BiDir PCI	**PCI Bus Width Enable**. This signal indicates the width of the bus when hot swap capability is enabled.
SWITCH	I	R1	5 V Tol In	**Insertion/Deassertion Indicator**. This signal is an input to the LSIFC929XL 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.
HOTSWAPEN/	I/O	T1	5 V Tol In	**Hot Swap Enable**. When this signal is LOW, the LSIFC929XL 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.
GPIO[2] (BLUELED/)	I/O	K1	3.3 V BiDir 8 mA with pull-up	**GPIO[2] (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. Refer also to the GPIO[2] pin description in Table 4.4 on page 4-14 for details on other operational capabilities of this signal.

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	B10	Diff Tx	Transmit differential data (Channel0).
TX1+	O	B15	Diff Tx	Transmit differential data (Channel1).
TX0–	O	A10	Diff Tx	Transmit differential data (Channel0).
TX1–	O	A15	Diff Tx	Transmit differential data (Channel1).
RX0+	I	B12	Diff Rx	Receive differential data (Channel0).
RX1+	I	B17	Diff Rx	Receive differential data (Channel1).
RX0–	I	A12	Diff Rx	Receive differential data (Channel0).
RX1–	I	A17	Diff Rx	Receive differential data (Channel1).
RTRIM	I	C13		**Trim Resistor**. This pin is the analog current reference for the integrated transceiver core. A 2.74 kΩ ± 1% resistor should be tied from the RTRIM pad to either the RXVDD0 or the RXVDD1 pin.
LIPRESET/	O	C14	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	B8	3.3 V TTL Input with pull-up	**Electrical Fault**. This active-LOW pin indicates that an electrical fault has been detected by the channel0 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.

Table 4.2 Fibre Channel Interface (Cont.)

Signal	I/O	BGA Pad No.	Pad Type	Description
FAULT1/	I	B19	3.3 V TTL Input with pull-up	**Electrical Fault**. This active-LOW pin indicates that an electrical fault has been detected by the channel1 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	B7	3.3 V BiDir 4 mA	**Output Disable, Channel0.** This output, when asserted, disables an external GBIC or MIA transmitter for channel0. This output also clears a module fault.
ODIS1	O	B20	3.3 V BiDir 4 mA	**Output Disable, Channel1.** This output when asserted disables an external GBIC or MIA transmitter for channel1. This output also clears a module fault.
BYPASS0/	O	D8	3.3 V BiDir 4 mA	**Bypass**. This line is driven LOW when the LSIFC929XL Link Controller block determines that channel0 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	D19	3.3 V BiDir 4 mA	**Bypass**. This line is driven LOW when the LSIFC929XL Link Controller block determines that channel1 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	A8	3.3 V 4 mA BiDir with pull-down	**Received Signal Loss**. This line is driven HIGH, disabling the on-chip receiver, when the GBIC for channel0 of the LSIFC929XL detects a loss of signal. If enabled through the Link Control register, this signal becomes an output test strobe.

Table 4.2 Fibre Channel Interface (Cont.)

Signal	I/O	BGA Pad No.	Pad Type	Description
RXLOS1	I	A19	3.3 V 4 mA BiDir with pull-down	**Received Signal Loss**. This line is driven HIGH, disabling the on-chip receiver, when the GBIC for channel1 of the LSIFC929XL detects a loss of signal. If enabled through the Link Control register, this signal becomes an output test strobe.
MODDEF0[2:0]	I	D9, A7, E11	3.3 V BiDir 8 mA with pull-up	**Module Identifiers**. GBIC and pluggable small form factor (SFF) optical module Identifiers (channel0).
MODDEF1[2:0]	I	E16, A20, D18	3.3 V BiDir 8 mA with pull-up	**Module Identifiers**. GBIC and pluggable SFF optical module Identifiers (channel1).
REFCLK	I	C1	3.3 V Schmitt Input	**FC Reference Clock**. FC reference clock (106.25 MHz ± 100 ppm).

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] *MD[31:24]*	I/O	P24, P23, N26, N25, N24, N23, M26, M25, M23, L26, L25, L23, K26, K25, K24, K23, D25, C26, C25, B25, A24, B24, C24, A23, B23, D23, A22, B22, C21, D21, D20, E20	3.3 V BiDir 4 mA	**SSRAM Read/Write Data**. *MD[31:24] are used for the FLASH ROM Read/Write Data.*
MP[3:0]	I/O	P25, J24, D26, E19	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 LSIFC929XL Programming Model.

Table 4.3 Memory Interface (Cont.)

Signal	I/O	BGA Pad No.	Pad Type	Description
MA[21:0]	I/O	AC26, AB26, AA25, Y26, Y25, W26, W25, W23, V26, V25, V24, U26, U25, U24, U23, T26, T25, T23, R26, R25, R23, R22	3.3 V BiDir 4 mA	**SSRAM/FLASH ROM Address**. These pins are also used at power-on to provide configuration information to the LSIFC929XL. Following are the power-on sense functions of each of the pins: Note: "1" means the pin is pulled up on reset "0" means the pin is pulled down on reset (these pins have internal put-downs) MA[21:18] **Not Applicable** for power-on sense MA[17] 1 = Keep the IOP from booting 0 = Enable IOP booting following reset MA[16] 1 = Channel 1 RXLOS1 signal polarity is active LOW 0 = Channel 1 RXLOS1 signal polarity is active HIGH MA[15] 1 = Channel 0 RXLOS0 signal polarity is active LOW 0 = Channel 0 RXLOS0 signal polarity is active HIGH MA[14] 1 = FAULT1/ signal polarity is active HIGH 0 = FAULT1/ signal polarity is active LOW MA[13] 1 = FAULT0/ signal polarity is active HIGH 0 = FAULT0/ signal polarity is active LOW MA[12:11] 00 = PCI ROM size is 256 Kbytes 01 = PCI ROM size is 512 Kbytes 10 = PCI ROM size is 1 Mbyte 01 = No ROM is present MA[10] 1 = Do not report bus as 66 MHz capable 0 = Report device as 66 MHz capable MA[9] 1 = Report device as not supporting PCI 64-bit mode 0 = Report device as supporting PCI 64-bit mode MA[8] 1 = Report device as supporting 66 MHz PCI-X 0 = Report device as supporting 133 MHz PCI-X

Table 4.3 Memory Interface (Cont.)

Signal	I/O	BGA Pad No.	Pad Type	Description
MA[21:0]	I/O			MA[7] 1 = Report device as not PCI-X compatible 0 = Report device as PCI-X compatible MA[6] 1 = PciFunction1 SubSysCntlId LSB will be a “1” 0 = PciFunction1 SubSysCntlId LSB will be a “0” MA[5] 1 = PciFunction0 SubSysCntlId LSB will be a “1” 0 = PciFunction0 SubSysCntlId LSB will be a “0” MA[4] 1 = PciFunction1 DeviceId LSB will be a “1” 0 = PciFunction1 DeviceId LSB will be a “0” MA[3] 1 = PciFunction0 DeviceId LSB will be a “1” 0 = PciFunction0 DeviceId LSB will be a “0” MA[2] 1 = Force single function device (EEPROM override) 0 = Do not force single function device (EEPROM override) MA[1] **Not Applicable** for power-on sense MA[0] 1 = Set to this if EEPROM size is 1 Kbyte, 2 Kbytes, 4 Kbytes, 8 Kbytes, or 16 Kbytes 0 = Set to this if EEPROM size is 32 Kbytes or 64 Kbytes
MOE[1:0]/	O	F26, E26	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	G23, F25	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	E24	3.3 V BiDir 4 mA	**FLASH Chip Select**. This active-LOW chip select allows connection of a single, 8-bit FLASH ROM device.

Table 4.3 Memory Interface (Cont.)

Signal	I/O	BGA Pad No.	Pad Type	Description
MCLK	B	J26	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. Exceptions are MOE/ and ZZ, which are typically asynchronous inputs to SRAM and/or FLASH devices.
ADSC/	B	J25	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/	B	H26	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	H25, H23, G26, G25	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	F24	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).
ZZ	O	F23	3.3 V BiDir 4 mA	**Snooze Control**. Asserting this output HIGH causes a synchronous SRAM to enter its lowest power state (not all RAMs support this function).

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	J2, K3, H2, K1, J3, J1	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 and may be configured as inputs or outputs. GPIO[3] may be optionally enabled as an external interrupt source to the ARM RISC processor core. Refer also to the GPIO[2] pin description in Table 4.1 on page 4-3 for details on other operational capabilities of this signal.
LED[4:0]/	O	B3, E7, D6, B4, D7	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 LED[3]: Channel 1 – Active LED[2]: Channel 0 – Fault LED[1]: Channel 0 – Active LED[0]: Firmware Controlled (Heartbeat)
SCL	O	B21	3.3 V 4 mA BiDir with pull-up	**Serial EEPROM clock**.
SDA	I/O	A21	3.3 V 4 mA BiDir with pull-up	**Serial EEPROM data**.
MODE[7:0]	I	A4, C3, A5, D4, B2, E1, C2, E3	3.3 V TTL Input with pull-up	**Mode Select.** This 8-bit bus defines operational and test modes for the chip. Valid mode encodings are as follows: Mode[7:0] = 00111111 — Interleaved PBSRAM Mode[7:0] = 00011111 — Noninterleaved PBSRAM
BZRSET		P3		**Reference resistor node** for the PCI-X impedance controller.
BZVDD		R4		**Reference resistor node** for the PCI-X impedance controller.

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	L2	3.3 V Schmitt with pull-up	JTAG/CtxMgr Debug **Test Clock**.
TRST/	I	M2	3.3 V Schmitt with pull-up	JTAG/Debug **Test Reset**. Asynchronous active LOW.
TDI	I	N4	3.3 V Schmitt with pull-up	JTAG/CtxMgr Debug **Test Data In**.
TDO	B	M4	3.3 V 4 mA T/S Output with pull-up	JTAG/CtxMgr Debug **Test Data Out**.
TMS_CHIP	I	L4	3.3 V Schmitt with pull-up	JTAG **Test Mode Select**.
TMS_ICE	I	N1	3.3 V Schmitt with pull-up	CtxMgr Debug **Test Mode Select**.
TEST[9]	O	N3		**Factory Test Pin**. Not used during normal operation, and must be left unconnected.
TEST[8:0]	I	C6, H5, F3, F1, N2, L1, E8, B5, A6		**Factory Test Pins**. Not used during normal operation, and must be tied to VDDIO (3.3 V) or VSSIO (0 V) depending on pin, as follows: TEST[8] Use 4.7 KΩ pull-up to VDDIO (3.3 V). TEST[7] Use 4.7 KΩ pull-up to VDDIO (3.3 V). TEST[6] Use 4.7 KΩ pull-up to VDDIO (3.3 V). TEST[5] Use 4.7 KΩ pull-up to VDDIO (3.3 V). TEST[4] Use 330 Ω pull-down to VSSIO (0 V). TEST[3] Use 4.7 KΩ pull-up to VDDIO (3.3 V). TEST[2] Use 330 Ω pull-down to VSSIO (0 V). TEST[1] Use 330 Ω pull-down to VSSIO (0 V). TEST[0] Use 330 Ω pull-down to VSSIO (0 V).

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
VDDC[1]	C9, C18, D5, D22, E15, G5, G22, J4, J23, M5, M22, R5, T22, W22, AA1, AB4, AB7, AB11, AB16, AB20, AB23, AC9, AC18	Core power.	1.8 V
VSSC	C5, C10, C17, C22, E4, E12, E23, H4, H22, L5, L22, P26, T5, V4, V23, Y5, Y22, AB8, AB12, AB15, AB19, AC5, AC22	Core ground.	0 V
VDDIO	A2, A26, B1, C7, C11, C15, C19, C23, D3, E6, E10, E14, E18, E22, E25, F5, G24, H3, J22, K5, L24, M3, N22, P5, R24, T3, U22, V5, W24, Y3, AA22, AB3, AB5, AB9, AB13, AB17, AB21, AC24, AD5, AD8, AD12, AD16, AD20, AE26, AF1, AF25	I/O power.	3.3 V
VSSIO	A1, A25, B26, C4, C8, C12, C16, C20, D24, E2, E5, E9, E13, E17, E21, F22, G3, H24, J5, K22, L3, L11, L12, L13, L14, L15, L16, M11, M12, M13, M14, M15, M16, M24, N5, N11, N12, N13, N14, N15, N16, P11, P12, P13, P14, P15, P16, P22, R3, R11, R12, R13, R14, R15, R16, T11, T12, T13, T14, T15, T16, T24, U5, V22, W3, Y24, AA5, AB6, AB10, AB14, AB18, AB22, AB25, AC4, AD7, AD11, AD15, AD19, AD23, AE1, AF2, AF26	I/O ground.	0 V

Table 4.6 Power and Ground (Cont.)

Signal	BGA Pad No.	Description	Voltage
V5PCIX	U1, U2, AC2, AC7, AC16, AC23, AD1, AE2, AF9, AF11, AF12, AF17, AF24	PCI-X 5 V reference power supply.	5 V
REFPLLVDD	D2	**Analog power** for PCI FSN cell.	1.8 V
REFPLLVSS	G4	Analog ground for PCI FSN cell.	0 V
PCIPLLVDD	AD3	Analog power for ARM clock generation.	1.8 V
PCIPLLVSS	AD4	Analog ground for ARM clock generation.	0 V
RXBVDD0	B13	Analog power for integrated transceiver core.	1.8 V
RXBVSS0	A13	Analog ground for integrated transceiver core.	0 V
RXBVDD1	D16	Analog power for integrated transceiver core.	1.8 V
RXBVSS1	D17	Analog ground for integrated transceiver core.	0 V
RXVDD0	D13	Analog power for integrated transceiver core.	1.8 V
RXVSS0	D12	Analog ground for integrated transceiver core.	0 V
RXVDD1	A18	Analog power for integrated transceiver core.	1.8 V
RXVSS1	B18	Analog ground for integrated transceiver core.	0 V
TXBVDD0	D11	Analog power for integrated transceiver core.	1.8 V
TXBVSS0	D10	Analog ground for integrated transceiver core.	0 V
TXBVDD1	B14	Analog power for integrated transceiver core.	1.8 V
TXBVSS1	A14	Analog ground for integrated transceiver core.	0 V
TXVDD0	A9	Analog power for integrated transceiver core.	1.8 V
TXVSS0	B9	Analog ground for integrated transceiver core.	0 V
TXVDD1	D14	Analog power for integrated transceiver core.	1.8 V
TXVSS1	D15	Analog ground for integrated transceiver core.	0 V

1. The required core voltage on the LSIFC929XL is 1.8 V. The I/O pads are 5 V tolerant. The PCI I/O voltage requires 3.3 V, and the GigaBlaze Fibre Channel transceiver interface requires 1.8 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 PCI system with PCI buffers, the recommended power sequence is 1.8 V, then 5 V, and then 3.3 V; or make certain that the following conditions are met during the power cycling:
   (VDD1.8 > 1 V) before (VDD3.3 > 1 V)
   VDD5 > (VDD3.3 -0.3 V)

Chapter 5
PCI-X Functional Description

This chapter provides a general description of the PCI-X features contained in the LSIFC929XL 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 2.2,* and the *PCI-X Addendum to the PCI Local Bus Specification, Revision 1.0a*. The LSIFC929XL supports up to a 133 MHz, 64-bit PCI-X bus. The LSIFC929XL supports 64-bit addressing with Dual Address Cycle (DAC).

The LSIFC929XL is a true multifunction PCI-X device that presents a single electrical load to the PCI-X bus. The LSIFC929XL 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 LSIFC929XL 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 LSIFC929XL 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 LSIFC929XL ignores configuration read/write cycles when IDSEL is not asserted.

Because the LSIFC929XL 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 LSIFC929XL does not respond to any other encodings of AD[10:8]. Bits AD[7:2] select one of the 64 Dword registers in the LSIFC929XL 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 LSIFC929XL is not a PCI Bridge device, all PCI Configuration Commands designated for the LSIFC929XL 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 LSIFC929XL. 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 LSIFC929XL 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 LSIFC929XL 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
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]

1. The LSIFC929XL ignores reserved commands as a slave and never generates them as a master.
2. When acting as a slave in the PCI mode, the LSIFC929XL supports this command as the PCI Memory Read command.
3. When acting as a slave in the PCI mode, the LSIFC929XL supports this command as the PCI Memory Write command.

The following sections describe how the LSIFC929XL implements these commands.

5.3.1 Interrupt Acknowledge Command

The LSIFC929XL ignores this command as a slave and never generates it as a master.

5.3.2 Special Cycle Command

The LSIFC929XL 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 LSIFC929XL decodes the lower 32 address bits and ignores the upper 32 address bits. The LSIFC929XL 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 LSIFC929XL decodes the lower 32 address bits and ignores the upper 32 address bits. The LSIFC929XL supports this command when operating in either the PCI or PCI-X bus mode.

5.3.5 Memory Read Command

The LSIFC929XL 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 LSIFC929XL 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 LSIFC929XL 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 LSIFC929XL 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 LSIFC929XL never generates this command as a master. When a slave, the LSIFC929XL 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 LSIFC929XL never generates this command as a master. When a slave, the LSIFC929XL 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 LSIFC929XL never generates this command as a master, but does respond to it as a slave. A device on the PCI bus selects the LSIFC929XL 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 LSIFC929XL 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 LSIFC929XL never generates this command as a master, but does respond to it as a slave. A device on the PCI bus selects the LSIFC929XL 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 LSIFC929XL 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 LSIFC929XL 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 LSIFC929XL reads is a multiple of the cache line size, which the *PCI Local Bus Specification, Revision 2.2,* provides. The LSIFC929XL 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 LSIFC929XL 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 1.0a,* 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 LSIFC929XL supports the Split Completion command for all of these commands except the Interrupt Acknowledge command, which the LSIFC929XL neither responds to nor generates.

5.3.14 Dual Address Cycles (DAC) Command

The LSIFC929XL performs Dual Address Cycles (DAC), according to the *PCI Local Bus Specification, Revision 2.2*. The LSIFC929XL 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 LSIFC929XL supports this command when operating in the PCI mode.

5.3.16 Memory Read Block Command

The LSIFC929XL uses this command to read from memory. The LSIFC929XL 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 LSIFC929XL 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 LSIFC929XL 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 LSIFC929XL bursts data using a noncache command. If the starting address aligns, the LSIFC929XL 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 LSIFC929XL 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 LSIFC929XL 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 LSIFC929XL issues a Memory Write command on the next cache boundary to complete the data transfer.

5.3.18 Memory Write Block Command

The LSIFC929XL uses this command to burst data to memory. The LSIFC929XL supports this command when operating in the PCI-X bus mode.

5.4 PCI Arbitration

The LSIFC929XL 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 LSIFC929XL 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 LSIFC929XL 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 LSIFC929XL 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”
- Section 6.3, “Shared Memory”

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 LSIFC929XL. 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 LSIFC929XL 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.

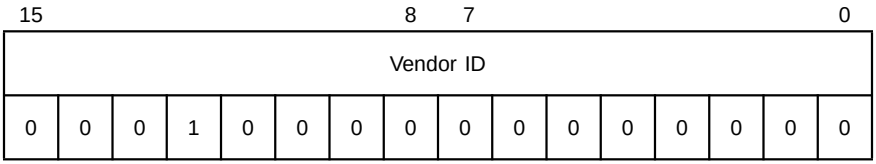
The LSIFC929XL enables, orders, and locates the PCI-extended capability register structures (Power Management, Messaged Signaled Interrupts, and PCI-X) to optimize device performance. The LSIFC929XL

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.

Table 6.1 LSIFC929XL PCI-X Configuration Space Address Map

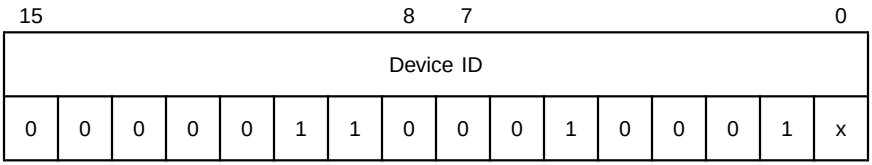
31	24	23	16	15	8	7	0	Offset	Page
Device ID				Vendor ID				0x00	6-3
Status				Command				0x04	6-4
Class Code						Revision ID		0x08	6-8
Reserved		Header Type		Latency Timer		Cache Line Size		0x0C	6-9
I/O Base Address								0x10	6-11
Memory [0] Base Address Low								0x14	6-11
Memory [0] Base Address High								0x18	6-12
Memory [1] Base Address Low								0x1C	6-12
Memory [1] Base Address High								0x20	6-13
Reserved								0x24	–
								0x28	–
Subsystem ID				Subsystem Vendor ID				0x2C	6-14
Expansion ROM Base Address								0x30	6-15
Reserved						Capabilities Pointer		0x34	6-16
								0x38	–
Maximum Latency		Minimum Grant		Interrupt Pin		Interrupt Line		0x3C	6-17
Reserved								0x40–0x7F	–
Power Management Capabilities				PM Next Pointer		PM Capability ID			6-19
PM Data		PM BSE		Power Management Control/Status					6-21
Reserved									–
Message Control				MSI Next Pointer		MSI Capability ID			6-23
Message Address									6-25
Message Upper Address									6-25
				Message Data					6-26
Reserved									–
PCI-X Command				PCI-X Next Pointer		PCI-X Capability ID			6-27
PCI-X Status									6-29
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



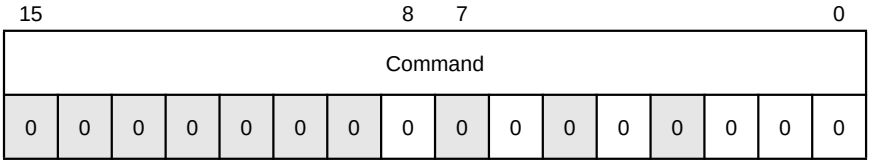
Device ID **[15:0]**
This register identifies the particular device. The most significant 12 bits are hardcoded to a constant of 0x062. The LSB is dependent upon the power-on-sense functions corresponding to the states of pins MA[4] and MA[3] as decoded in Table 6.2.

Table 6.2 Device ID Values

Single/Dual Channel	State of MA[4:3]	Device ID
Function 1		
Dual Channel	MA[4] = 0	0x0626
Dual Channel	MA[4] = 1	0x0627
Single Channel	MA[4] = 0	0x0628
Single Channel	MA[4] = 1	0x0629
Function 0		
Dual Channel	MA[3] = 0	0x0626
Dual Channel	MA[3] = 1	0x0627
Single Channel	MA[3] = 0	0x0628
Single Channel	MA[3] = 1	0x0629

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 LSIFC929XL 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 LSIFC929XL 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 LSIFC929XL PCI function to detect parity errors on the PCI bus and report these errors to the system. Clearing this bit causes the LSIFC929XL 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 LSIFC929XL 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 LSIFC929XL 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

15								8	7						0
Status															
0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0

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 2.2*, and the *PCI-X Addendum to the PCI Local Bus Specification, Revision 1.0a*.

Signaled System Error **14**
The LSIFC929XL 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 LSIFC929XL 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 2.2*, and the *PCI-X Addendum to the PCI Local Bus Specification, Revision 1.0a*. Refer to bit 0 of the PCI-X Command register for details.

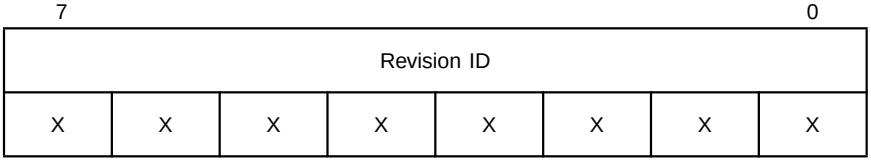
Reserved **[7:6]**
This field is reserved.

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 LSIFC929XL PCI function is capable of operating at 66 MHz. Pulling MA[10] HIGH clears this bit and indicates to the host system that the LSIFC929XL PCI function is not capable of operating at 66 MHz. Refer to Table 4.3 on page 4-10 for details.

New Capabilities **4**
The LSIFC929XL 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.

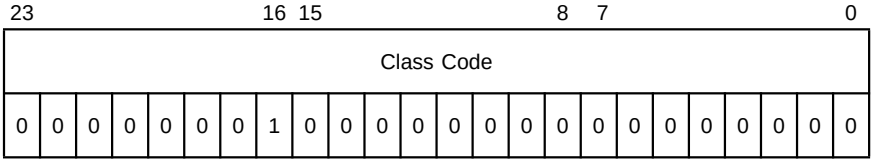
Reserved **[3:0]**
This field is reserved.

Register: 0x08
Revision ID
Read/Write



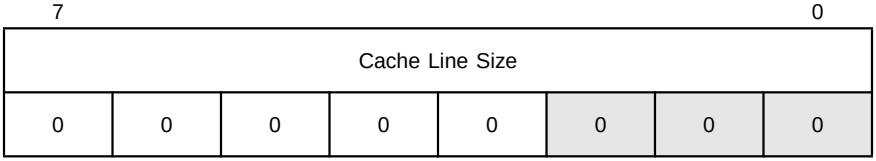
Revision ID **[7:0]**
This register indicates the current revision level of the device.

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.

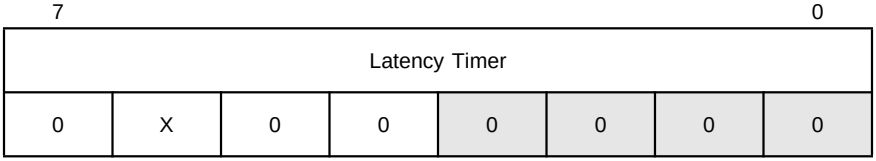
Register: 0x0C
Cache Line Size
Read/Write



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 LSIFC929XL 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 LSIFC929XL PCI function ignores this register when operating in the PCI-X mode.

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

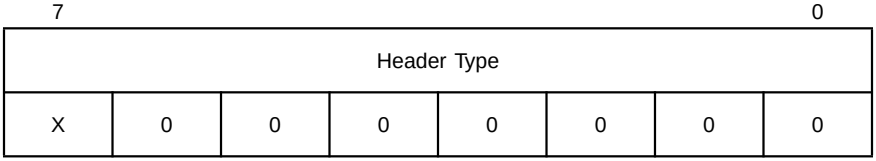
Register: 0x0D
Latency Timer
Read/Write



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

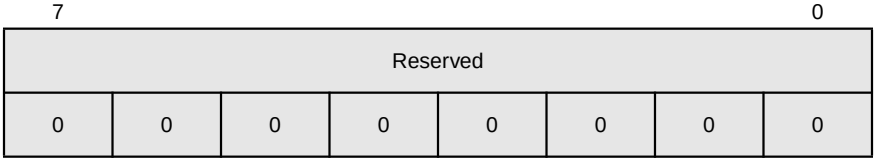
Reserved **[3:0]**
This field is reserved.

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.

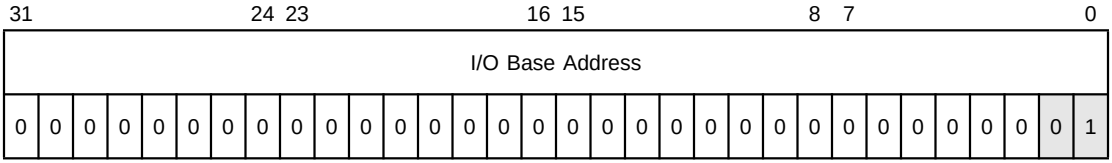
Register: 0x0F
Reserved



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

Register: 0x10–0x13

I/O Base Address
Read/Write



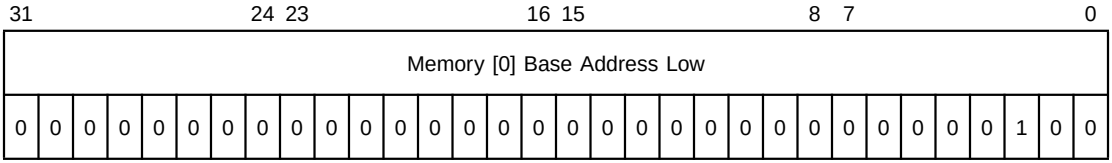
This register maps the operating register set into I/O space. The LSIFC929XL requires 256 bytes of I/O space for this register. Hardware sets bit 0 to 0b1. Bit 1 is reserved and returns 0b0 on all reads.

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

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

Register: 0x14–0x17

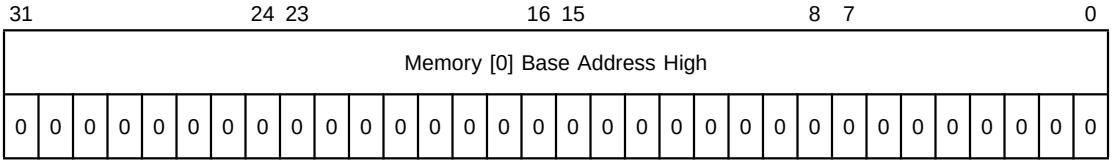
Memory [0] Base Address Low
Read/Write



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 LSIFC929XL requires 1024 bytes of memory space.

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

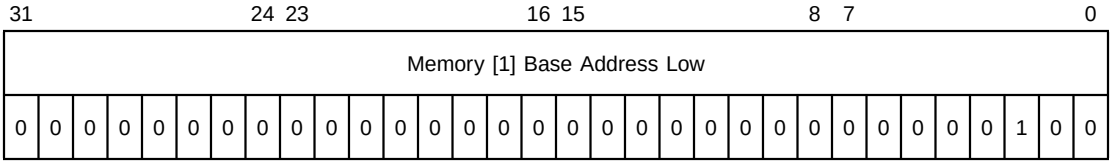
Register: 0x18–0x1B
Memory [0] Base Address High
Read/Write



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 LSIFC929XL requires 1024 bytes of memory space.

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

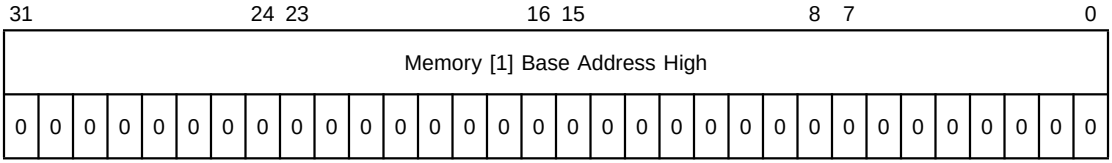
Register: 0x1C–0x1F
Memory [1] Base Address Low
Read/Write



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 LSIFC929XL 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.

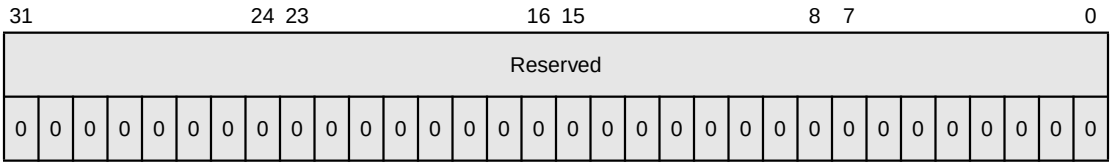
Register: 0x20–0x23
Memory [1] Base Address High
Read/Write



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 LSIFC929XL requires 64 Kbytes of memory for Memory Space [1].

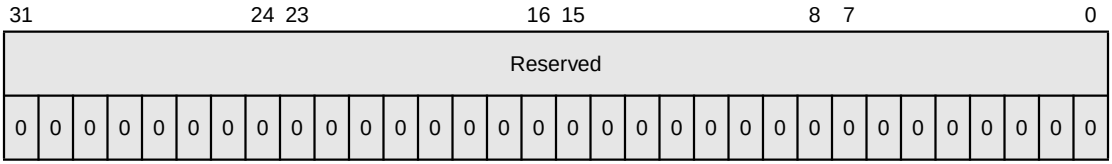
Memory [1] Base Address High **[31:0]**
This field contains the Memory [1] Base Address High address.

Register: 0x24–0x27
Reserved



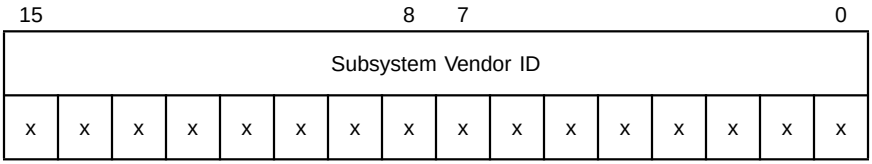
Reserved **[31:0]**
This register is reserved.

Register: 0x28–0x2B
Reserved



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

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

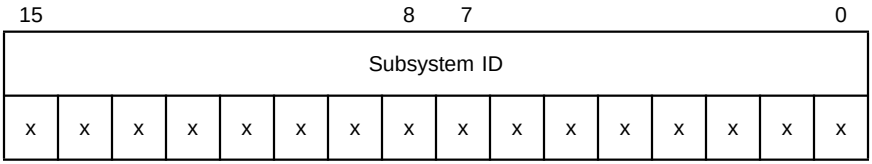


SVID **Subsystem Vendor ID** **[15:0]**

This 16-bit register uniquely identifies the vendor that manufactures the add-in board or subsystem where the LSIFC929XL 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

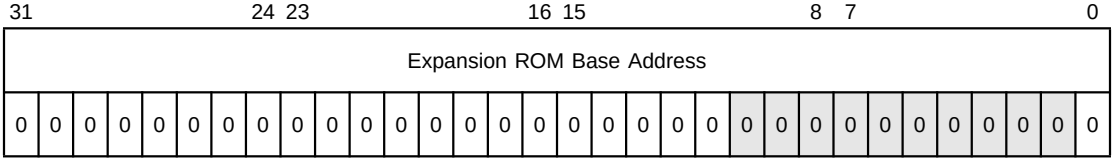


SID **Subsystem ID** **[15:0]**

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



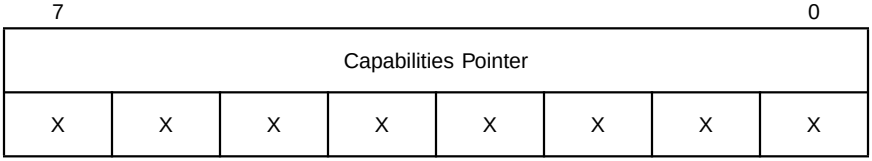
This 32-bit register contains the base address and size information for the expansion ROM.

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 LSIFC929XL 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.

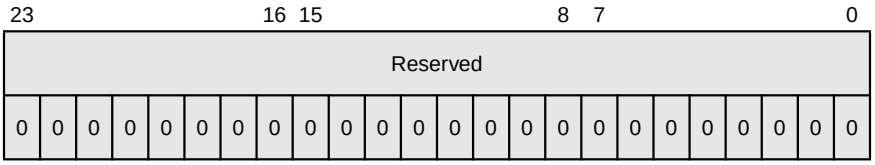
Expansion ROM Enable **0**
This bit controls whether the device accepts accesses to its expansion ROM. Setting this bit enables address decoding. Depending on the system configuration, the device can optionally use an expansion ROM. Note that to access the expansion ROM, the user must also set bit 1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

Register: 0x34
Capabilities Pointer
Read Only



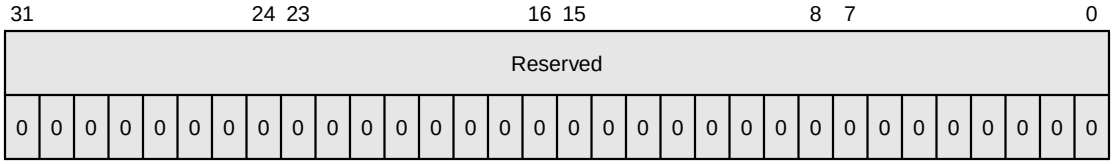
Capabilities Pointer **[7:0]**
This register indicates the location of the first extended capabilities register in PCI Configuration Space. The value of this register varies according to system configuration.

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

7							0
Interrupt Line							
0	0	0	0	0	0	0	0

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

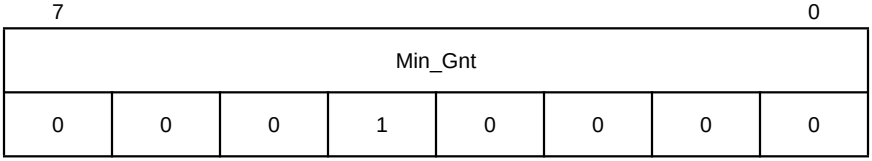
Interrupt Pin
Read Only

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]**

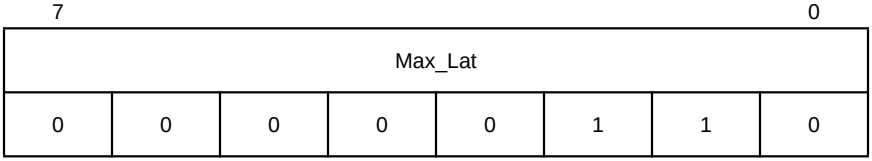
The encoding of this read-only register is unique to each function on the LSIFC929XL. It indicates which interrupt pin the function uses. The value for Function [0] is 0x01, which indicates that Function [0] presents interrupts on the INTA/ or ALT_INTA pins. The value for Function [1] is 0x02, which indicates that Function [1] presents interrupts on the INTB/ or ALT_INTB/ pins.

Register: 0x3E
Minimum Grant
Read Only



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 LSIFC929XL sets this register to 0x10, indicating a burst period of 4.0 µs.

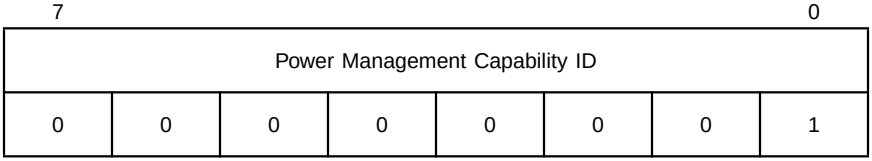
Register: 0x3F
Maximum Latency
Read Only



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 LSIFC929XL sets this register to 0x06, indicating a burst period of 1.5 µs.

Register: 0xXX

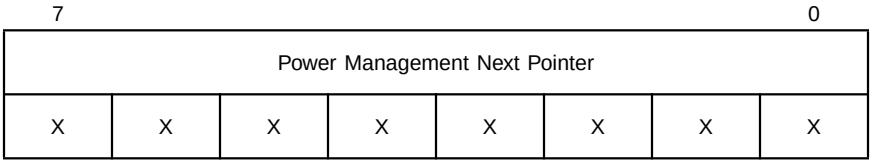
Power Management Capability ID
Read Only



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

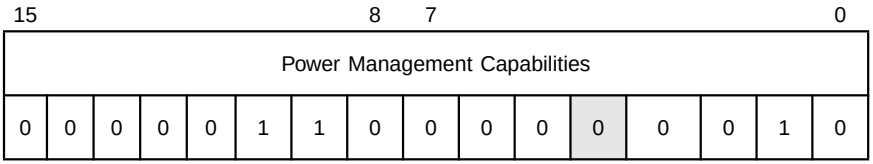
Power Management Next Pointer
Read Only



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



PME_Support **[15:11]**
These bits define the power management states in which the device asserts the Power Management Event (PME) pin. The LSIFC929XL clears these bits because the LSIFC929XL does not provide a PME signal.

D2_Support **10**
The PCI function sets this bit since the LSIFC929XL supports power management state D2.

D1_Support **9**
The PCI function sets this bit because the LSIFC929XL supports power management state D1.

Aux_Current **[8:6]**
The PCI function clears this field because the LSIFC929XL does not support Aux_Current.

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

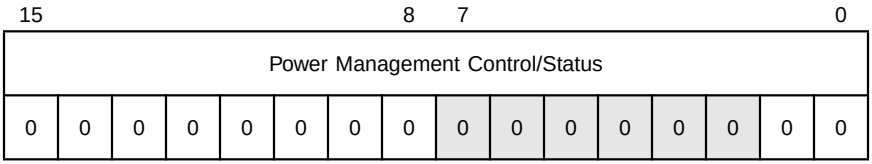
Reserved **4**
This bit is reserved.

PME Clock **3**
The LSIFC929XL clears this bit because the chip does not provide a PME pin.

Version **[2:0]**
The PCI function programs these bits to 0b010 to indicate that the LSIFC929XL complies with the *PCI Power Management Interface Specification, Revision 1.1*.

Register: 0xXX

Power Management Control/Status
Read/Write



PME_Status **15**
The PCI function clears this bit because the LSIFC929XL does not support PME signal generation from $D3_{cold}$.

Data_Scale **[14:13]**
The PCI function clears this bit because the LSIFC929XL does not support the Power Management Data register.

Data_Select **[12:9]**
The PCI function clears these bits because the LSIFC929XL does not support the Power Management Data register.

PME_Enable **8**
The PCI function clears this bit because the LSIFC929XL does not provide a PME signal and disables PME assertion.

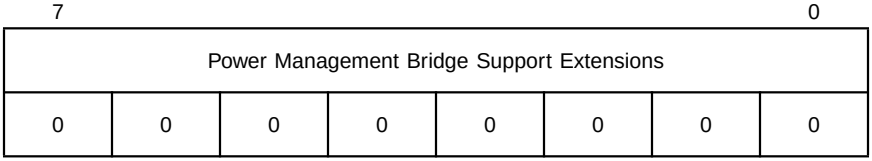
Reserved **[7:2]**
This field is reserved.

Power State **[1:0]**
These bits determine the current power state of the LSIFC929XL. Power states are as follows:

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

Register: 0xXX

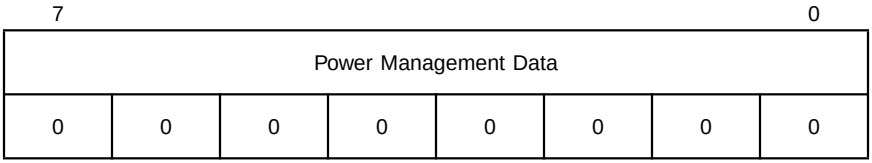
Power Management Bridge Support Extensions
Read Only



Power Management Bridge Support Extensions [7:0]
This register indicates PCI Bridge specific functionality. The LSIFC929XL always returns 0x00 in this register.

Register: 0xXX

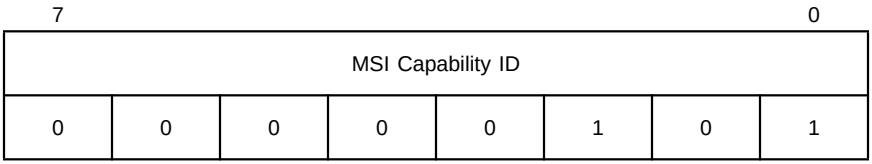
Power Management Data
Read Only



Power Management Data [7:0]
This register provides an optional mechanism for the PCI function to report state-dependent operating data. The LSIFC929XL always returns 0x00 in this register.

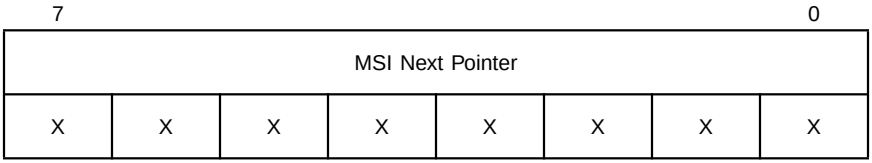
Register: 0xXX

MSI Capability ID
Read Only



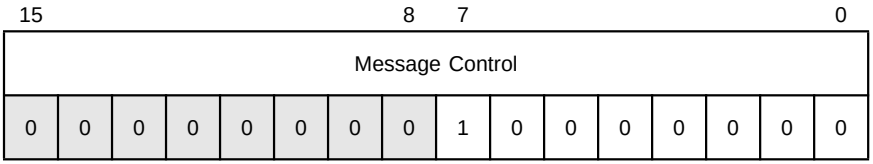
MSI Capability ID [7:0]
This register indicates the type of the current data structure. This register always returns 0x05, indicating a Message Signaled Interrupt (MSI).

Register: 0xXX
MSI Next Pointer
Read Only



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

Register: 0xXX
Message Control
Read/Write



Reserved **[15:8]**
This field is reserved.

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 LSIFC929XL. 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.

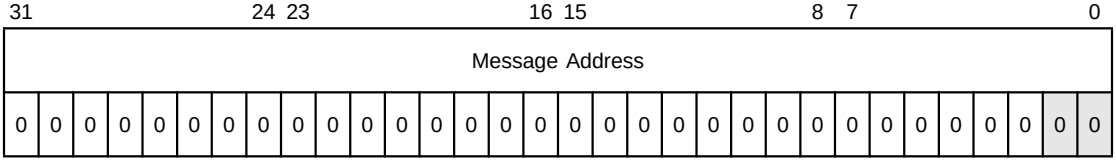
Table 6.3 Multiple Message Enable Field Bit Encoding

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 LSIFC929XL 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 LSIFC929XL 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. Setting this bit enables the device to use MSI to interrupt the host and request service.

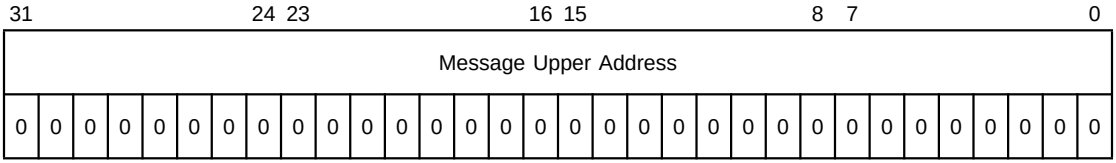
Register: 0xXX
Message Address
Read/Write



Message Address **[31:2]**
This field contains message address bits [31:2] for the MSI memory write transaction. The host system specifies and aligns the message address to a Dword. During the address phase, the LSIFC929XL drives Message Address[1:0] to 0b00.

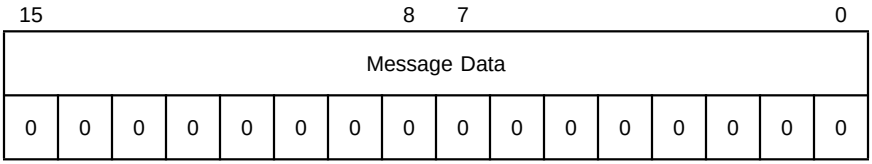
Reserved **[1:0]**
This field is reserved.

Register: 0xXX
Message Upper Address
Read/Write



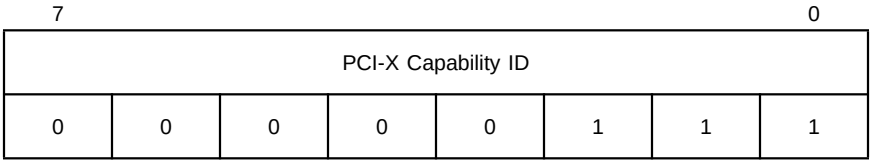
Message Upper Address **[31:0]**
The LSIFC929XL supports 64-bit MSI. This register contains the upper 32 bits of the 64-bit message address, which the system specifies. The host system software can program this register to 0x0000 to force the PCI function to generate 32-bit message addresses.

Register: 0xXX
Message Data
Read/Write



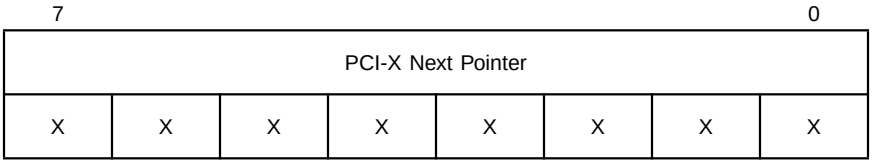
Message Data **[15:0]**
System software initializes this register by writing to it. The LSIFC929XL sends an interrupt message by writing a Dword to the address held in the Message Address and Message Upper Address registers. This register forms bits [15:0] of the Dword message that the PCI function passes to the host. The PCI function drives bits [31:16] of this message to 0x0000.

Register: 0xXX
PCI-X Capability ID
Read Only



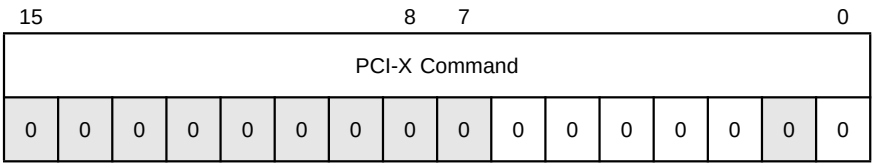
PCI-X Capability ID **[7:0]**
This register indicates the type of the current data structure. This register returns 0x07, indicating the PCI-X Data Structure.

Register: 0xXX
PCI-X Next Pointer
Read Only



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

Register: 0xXX
PCI-X Command
Read/Write



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

Maximum Outstanding Split Transactions **[6:4]**
These bits indicate the maximum number of split transactions the LSIFC929XL can have outstanding at one time. The LSIFC929XL uses the most recent value of this register each time it prepares a new sequence. Note that if the LSIFC929XL prepares a sequence before the setting of this field changes, the PCI function initiates the prepared sequence with the previous setting. Table 6.4 provides the bit encodings for this field.

Table 6.4 Maximum Outstanding Split Transactions

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

Maximum Memory Read Byte Count **[3:2]**
These bits indicate the maximum byte count the LSIFC929XL uses when initiating a sequence with one of the burst memory read commands. Table 6.5 provides the bit encodings for this field.

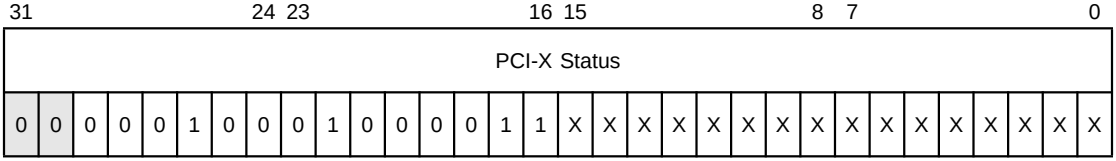
Table 6.5 Maximum Memory Read Byte Count

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 LSIFC929XL to attempt to recover from data parity errors. If the user clears this bit and the LSIFC929XL is operating in the PCI-X mode, the LSIFC929XL asserts SERR/ whenever the Data Parity Error Reported 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 LSIFC929XL 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 LSIFC929XL PCI device. The PCI function must report the smallest value that correctly indicates its capability. The LSIFC929XL reports 0b001 in this field to indicate a designed maximum cumulative read size of 2 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 LSIFC929XL PCI device. The PCI function must report the smallest value that correctly indicates its capability. The LSIFC929XL reports 0b000 in this field to indicate that the designed maximum number of outstanding split transactions is one.

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 LSIFC929XL PCI 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 LSIFC929XL 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 LSIFC929XL is a simple device.

Unexpected Split Completion **19**
The PCI function sets this read-only bit when it receives an unexpected split completion. After this bit is 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. After this bit is 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 Table 4.3 on page 4-10 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 LSIFC929XL on an add-in card, this bit must indicate the size of the PCI Address/Data bus on the card. Refer to Table 4.3 for details on the power-on sense pins.

Bus Number **[15:8]**
These read-only bits indicate the number of the LSIFC929XL bus segment. This 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 LSIFC929XL. This 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 either PCI I/O Space or PCI Memory Space [0]. Access to the Diagnostic Read/Write Data and Diagnostic Read/Write Address registers is available only through PCI I/O Space.

When the LSIFC929XL 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 LSIFC929XL operates as a single function PCI device, only PCI Function [0] register sets are accessible.

Table 6.6 defines the PCI I/O Space address map.

Table 6.6 PCI I/O Space Address Map

31 16 15 0	Offset	Page
System Doorbell	0x00	6-33
Write Sequence	0x04	6-34
Host Diagnostic	0x08	6-35
Test Base Address	0x0C	6-37
Diagnostic Read/Write Data	0x10	6-38
Diagnostic Read/Write Address	0x14	6-39
Reserved	0x18–0x2F	–
Host Interrupt Status	0x30	6-40
Host Interrupt Mask	0x34	6-41
Reserved	0x38–0x3F	–
Request FIFO	0x40	6-42
Reply FIFO	0x44	6-43
Reserved	0x48–0x4C	–
Host Index Register	0x50	6-43
Reserved	0x54–0x7F	–

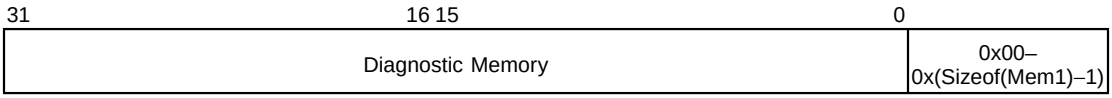
Table 6.7 defines the PCI Memory Space [0] address map.

Table 6.7 PCI Memory [0] Address Map

31 16 15 0	Offset	Page
System Doorbell	0x00	6-33
Write Sequence	0x04	6-34
Host Diagnostic	0x08	6-35
Test Base Address	0x0C	6-37
Reserved	0x10–0x2F	–
Host Interrupt Status	0x30	6-40
Host Interrupt Mask	0x34	6-41
Reserved	0x38–0x3F	–
Request FIFO	0x40	6-42
Reply FIFO	0x44	6-43
Reserved	0x48–0x7F	–
Shared Memory	0x80– 0x(Sizeof(Mem0)–1)	–

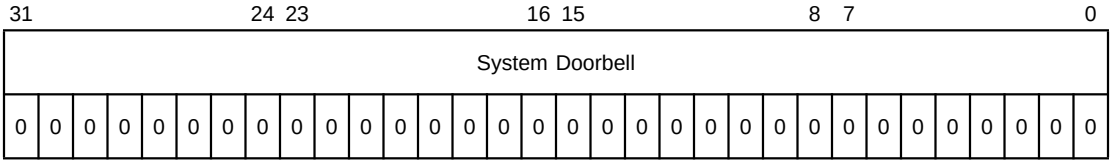
Table 6.8 defines the PCI Memory Space [1] address map.

Table 6.8 PCI Memory [1] Address Map



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

Register: 0x00
System Doorbell
Read/Write



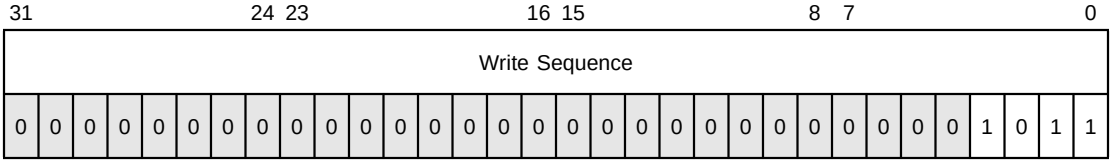
This register is a simple message passing mechanism that allows the system to pass single word messages to the embedded IOP processor and vice versa. There is a unique system doorbell for each PCI function.

When a host system PCI master writes to the Host Registers → Doorbell register, the LSIFC929XL 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 LSIFC929XL 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

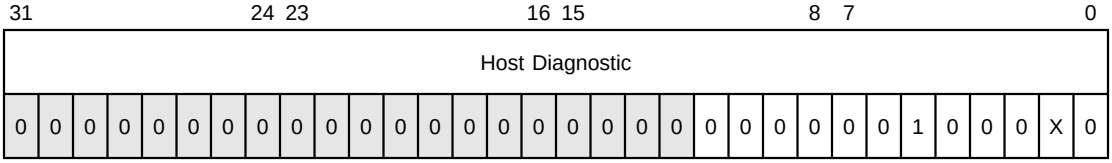


This register provides a protection mechanism against inadvertent writes to the Host Diagnostic register. There is one Write I/O register that is visible to both PCI functions. The two PCI functions physically share this register.

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 LSIFC929XL sets this read-only bit when the host writes the correct Write I/O Key to the Write Sequence register. The LSIFC929XL 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 LSIFC929XL sets this bit if the IOP ARM966E-S™ processor encounters a bad Flash signature when booting from Flash ROM. The LSIFC929XL also sets the DisARM bit (bit 1 in this register) to hold the IOP ARM

processor in a reset state. The LSIFC929XL maintains this state until the PCI host clears both the Flash Bad Signature and DisARM bits.

Reset History **5**
The LSIFC929XL 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.

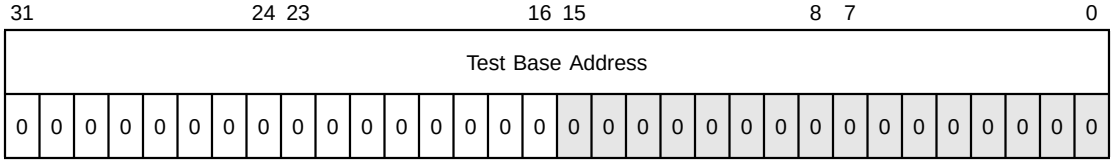
Reset Adapter **2**
Setting this write-only bit causes a hard reset within the LSIFC929XL. The bit self-clears after eight PCI clock periods. After deasserting this bit, the IOP ARM processor executes from its default reset vector.

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



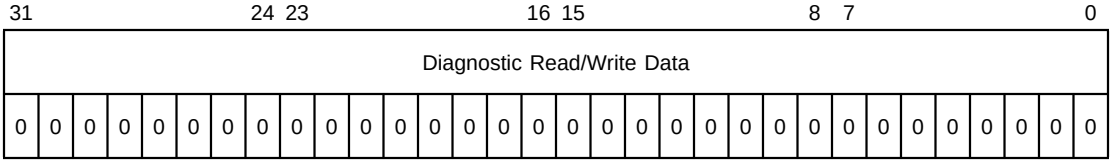
This register specifies the base address for Memory Space [1] accesses. There is one Test Base Address register that is visible to both PCI functions. The two PCI functions physically share this register. Because Diagnostic Memory is visible only to PCI Function [0], PCI Function [1] cannot write to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

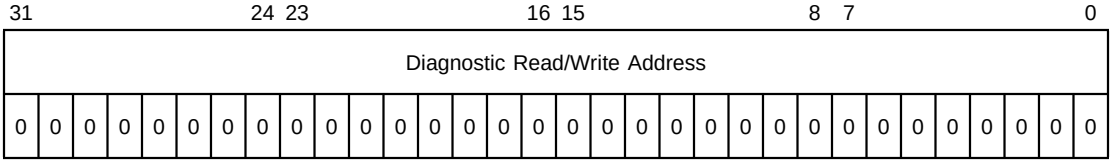


This register reads or writes Dword locations on the LSIFC929XL internal bus. This register is only accessible through PCI I/O space and returns 0xFFFFFFFF if read through PCI Memory Space. The host can enable write access to this register by writing the correct Write I/O Key to the Write Sequence register 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 LSIFC929XL reads/writes data at the address that the Diagnostic Read/Write Address register specifies.

Register: 0x14

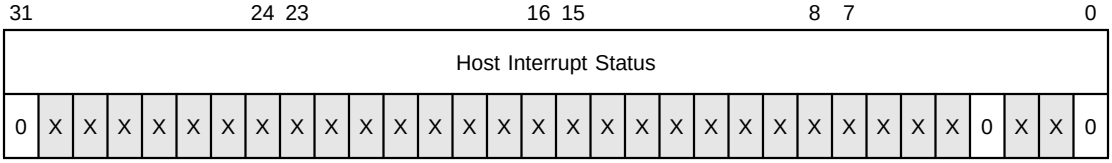
Diagnostic Read/Write Address
Read/Write



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 LSIFC929XL 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.

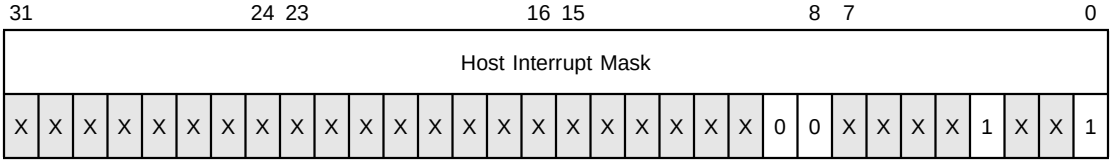
Reserved **[30:4]**
This field is reserved.

Reply Interrupt **3**
The LSIFC929XL sets this bit when the Reply Post FIFO is not empty. The LSIFC929XL 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 LSIFC929XL 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 LSIFC929XL 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 6.9. If the host system enables MSI, the LSIFC929XL does not signal PCI interrupts on the INTx/ pins.

Table 6.9 Interrupt Signal Routing

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

Note: The LSIFC929XL 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.

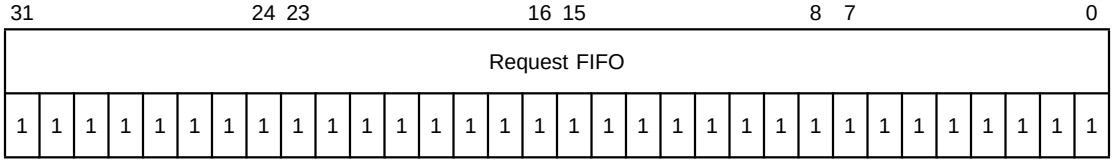
Reserved **[7:4]**
This field is reserved.

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.

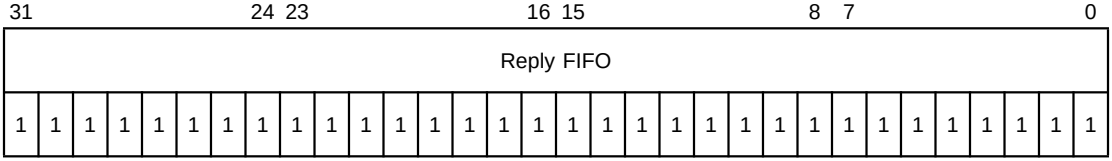
Register: 0x40
Request FIFO
Read/Write



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.

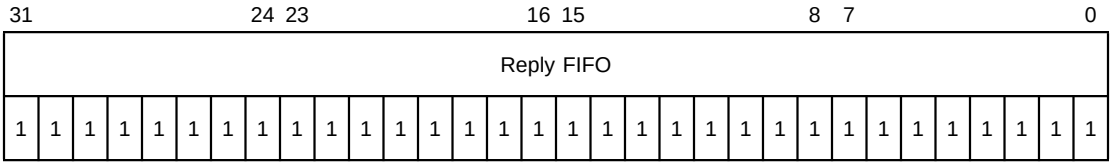
Register: 0x44
Reply FIFO
Read/Write



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: 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.

6.3 Shared Memory

A region of Shared Memory (LSIFC929XL local memory mapped to System Addresses) is provided to allow the host to write Request Message Frames. This is the default method (PUSH model) for Request Message Frame transport, where the Host itself copies the Request Message Frame into the LSIFC929XL local memory. The total size of Shared Memory is configured by the I/O Processor (IOP) on reset. Supported values are 32 Kbytes, 64 Kbytes, 128 Kbytes (default), 256 Kbytes, and 512 Kbytes. Shared memory is accessible only through Mem0 space starting at address 0x080.

Chapter 7
Specifications

This chapter provides a description of the DC and AC electrical characteristics of the LSIFC929XL Dual Channel Fibre Channel I/O processor chip, and the available packaging. The chapter contains the following sections:

7.1 Electrical Requirements

Table 7.1 provides absolute maximum stress ratings for the LSIFC929XL, while Table 7.2 specifies the normal operating conditions. Table 7.3 through Table 7.9 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	–
ESD	Electrostatic discharge	–	1.5 K	V	MIL-STD 883C, Method 3015.7

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. –3 V < V_{PIN} < 6.6 V.

Table 7.2 Operating Conditions[1]

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{DDC}[2]	Core supply voltage	1.71	1.89	V	–
V_{DDIO}	I/O supply voltage	3.0	3.6	V	–
PCI5VREF	PCI 5 V reference voltage	4.75 –	5.25 VDDIO	V V	5 V PCI System 3.3 V PCI System
T_A[3]	Operating free air	0	70	˚C	–
θ_{JMA}[4]	Thermal resistance (junction to moving air)	–	15.3	˚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 4.7 (page 4-16) for instructions on power sequencing for the LSIFC929XL.
3. Requires proper heatsink and airflow. See Section 7.5, “Package Thermal Considerations,” on page 7-17 for details.
4. θ_{JMAmax} (junction-to-moving air thermal resistance) assumes a 4-layer package substrate, a 456-pad PBGA, and a 4-layer PCB design (10–12 watt/meter ˚K). This maximum number is the worst-case θ_{JMA} for the LSIFC929XL with no heat sink and no air flow. Refer to Section 7.5, “Package Thermal Considerations,” on page 7-17 for details.

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, HOTSWAPEN/)

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[21:0], MWE[1:0]/, FLASHCS/, BWE[3:0]/, RAMCS/, ZZ, 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	2.0	V_{DD} + 0.5	V	5 V PCI System
		0.5 V_{DD}	5.5	V	3.3 V PCI System
V_{IL}	Input low voltage	–0.5	0.8	V	5 V PCI System
		–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	2.0	V_{DD} + 0.5	V	5 V PCI System
		0.5 V_{DD}	5.5	V	3.3 V PCI System
V_{IL}	Input low voltage	–0.5	0.8	V	5 V PCI System
		–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 (3.3 V PCI)
V_{OH}	Output high voltage	2.4	–	V	–2 mA (5 V PCI)
V_{OL}	Output low voltage	V_{SS}	0.1 V_{DD}	V	1.5 mA (3.3 V PCI)
V_{OL}	Output low voltage	–	0.55	V	3 mA, 6 mA (5 V PCI)[1]
I_{OZ}	3-state leakage	–10	10	µA	–

1. Signals without pull-up resistors meet a 3 mA output current load. Signals requiring pull-ups meet a 6 mA output current load. The latter include **FRAME/**, **TRDY/**, **IRDY/**, **STOP/**, **PERR/**, and, when used, **AD[63:32]**, **C_BE[7:4]**, and **ACK64/**.

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 (3.3 V PCI)
V_{OH}	Output high voltage	2.4	–	V	–2 mA (5 V PCI)
V_{OL}	Output low voltage	V_{SS}	0.1 V_{DD}	V	1.5 mA (3.3 V PCI)
V_{OL}	Output low voltage	–	0.55	V	3 mA, 6 mA (5 V PCI)[1]
I_{OZ}	3-state leakage	–10	10	µA	–

1. Signals without pull-up resistors meet a 3 mA output current load. Signals requiring pull-ups meet a 6 mA output current load. The latter include **DEVSEL/**, **SERR/**, **INTA/**, **INTB/**, and, when used, **PAR64**, and **REQ64/**.

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 LSIFC929XL PCI/PCI-X signals conform to the electrical and timing standards as shown in the *PCI Local Bus Specification, Version 2.2*, and the *PCI-X Addendum to the PCI Local Bus Specification, Version 1.0a*. All hardware validation testing performed by LSI Logic guarantees that the LSIFC929XL meets or exceeds the specifications contained in those documents.

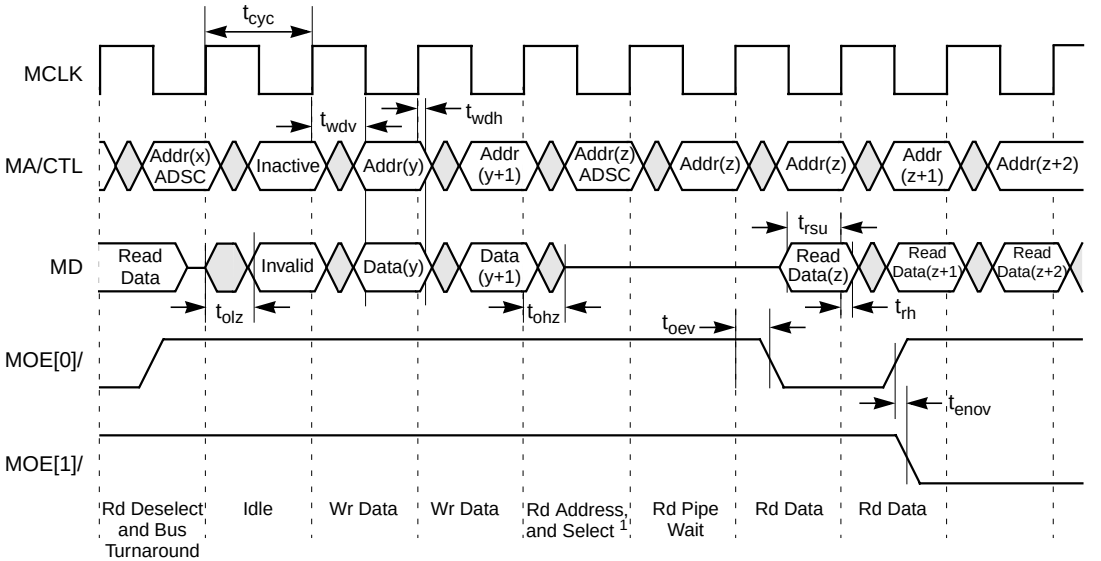
7.2.2 Fibre Channel Interface Timings

The LSIFC929XL 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 LSIFC929XL meets or exceeds the specifications contained in that document.

7.2.3 Memory Interface Timings

7.2.3.1 SSRAM Timings

Figure 7.1 SSRAM Read/Write/Read Timing Waveforms



Note 1: To avoid device contention, at least one idle cycle exists between the end of a write burst and a read cycle.

Table 7.11 SSRAM Read/Write/Read Timings

Symbol	Parameter		Min	Max	Unit
t_{cyc}	MCLK cycle time		12.548	12.550	ns
t_{rsu}	Read setup time		5	–	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]	0.2	–	ns
		MA[21:0]	0.4	–	ns
		Control Signals[2]	0.6	–	ns
t_{oev}	Output enable valid		–	2.5	ns
t_{olz}	Data low impedance		1.75	6.5	ns
t_{ohz}	Data high impedance		0.25	3.5	ns
t_{enov}	Output enable overlap		–	0	ns

1. Refer to SEN #11082, “LSIFC929XL 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 Timings

Figure 7.2 Flash ROM Read Timing Waveforms

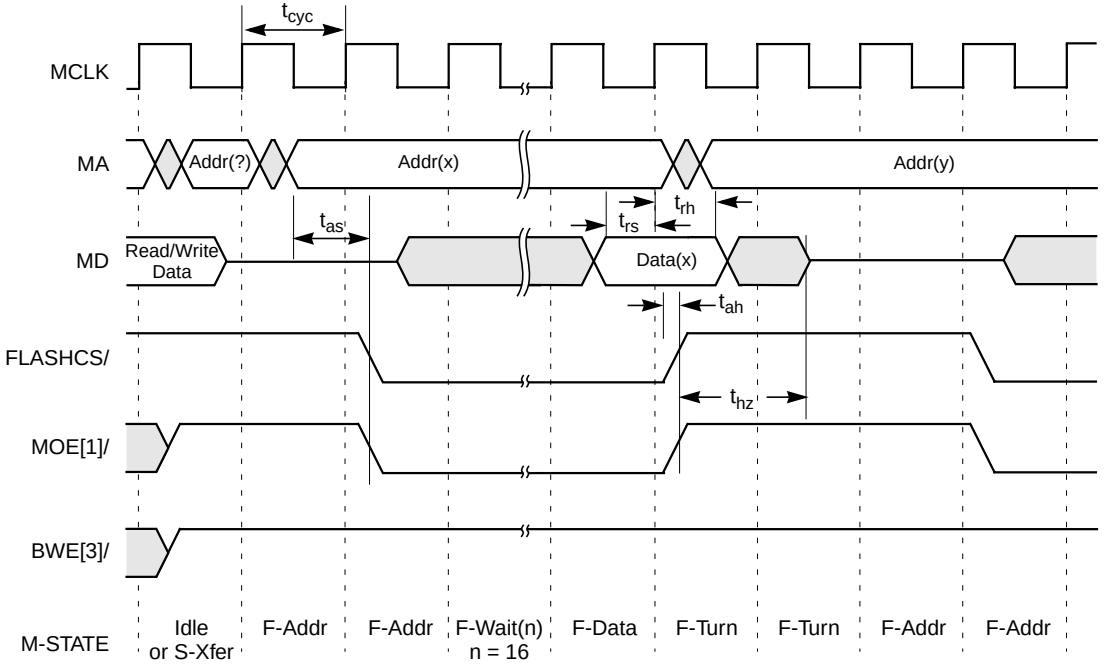


Table 7.12 FLASH ROM Read Timings

Symbol	Parameter	Min	Max	Unit
t_{cyc}	MCLK cycle time	12.548	12.550	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.

Figure 7.3 Flash ROM Write Timing Waveforms

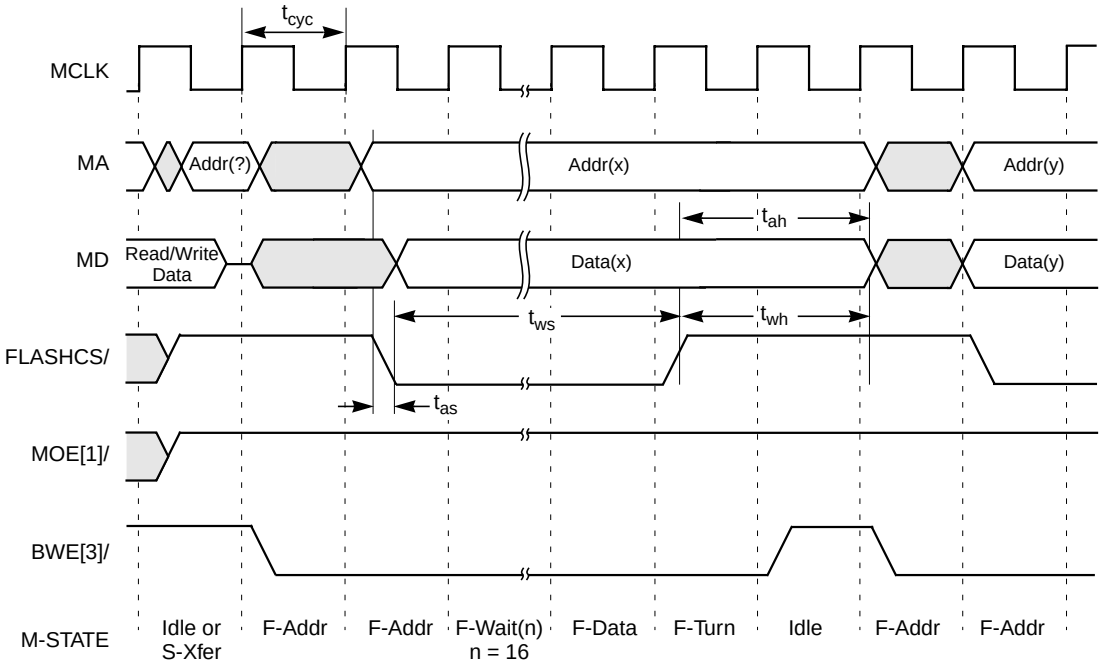


Table 7.13 Flash ROM Write Timings

Symbol	Parameter	Min	Max	Unit
t_{cyc}	MCLK cycle time	12.548	12.550	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 456 Plastic Ball Grid Array (PBGA). Table 7.14 on page 7-12 lists the LSIFC929XL signals in alphanumeric order by PBGA position. Table 7.15 on page 7-14 lists the LSIFC929XL signals alphanumerically by signal name. Figure 7.5 on page 7-16 is the mechanical drawing of the package for the LSIFC929XL.

Figure 7.4 LSIFC929XL 456-Pin PBGA Top View

A1 VSSIO	A2 VDDIO	A3 NC	A4 MODE[7]	A5 MODE[5]	A6 TEST[0]	A7 MODDEF0[1]	A8 RXLOS0	A9 TXVDD0	A10 TX0-	A11 NC	A12 RX0–	A13 RXBVSS0
B1 VDDIO	B2 MODE[3]	B3 LED[4]/	B4 LED[1]/	B5 TEST[1]	B6 NC	B7 ODIS0	B8 FAULT0/	B9 TXVSS0	B10 TX0+	B11 NC	B12 RX0+	B13 RXBVDD0
C1 REFCLK	C2 MODE[1]	C3 MODE[6]	C4 VSSIO	C5 VSSC	C6 TEST[8]	C7 VDDIO	C8 VSSIO	C9 VDDC	C10 VSSC	C11 VDDIO	C12 VSSIO	C13 RTRIM
D1 NC	D2 REF-PLLVDD	D3 VDDIO	D4 MODE[4]	D5 VDDC	D6 LED[2]/	D7 LED[0]/	D8 BYPASS0/	D9 MODDEF0[2]	D10 TXBVSS0	D11 TXBVDD0	D12 RXVSS0	D13 RXVDD0
E1 MODE[2]	E2 VSSIO	E3 MODE[0]	E4 VSSC	E5 VSSIO	E6 VDDIO	E7 LED[3]/	E8 TEST[2]	E9 VSSIO	E10 VDDIO	E11 MODDEF0[0]	E12 VSSC	E13 VSSIO
F1 TEST[5]	F2 NC	F3 TEST[6]	F4 NC	F5 VDDIO								
G1 NC	G2 NC	G3 VSSIO	G4 REFPLLVSS	G5 VDDC								
H1 NC	H2 GPIO[3]	H3 VDDIO	H4 VSSC	H5 TEST[7]								
J1 GPIO[0]	J2 GPIO[5]	J3 GPIO[1]	J4 VDDC	J5 VSSIO								
K1 GPIO[2] (BLUELED/)	K2 NC	K3 GPIO[4]	K4 NC	K5 VDDIO								
L1 TEST[3]	L2 TCK	L3 VSSIO	L4 TMS_CHIP	L5 VSSC						L11 VSSIO	L12 VSSIO	L13 VSSIO
M1 NC	M2 TRST/	M3 VDDIO	M4 TDO	M5 VDDC						M11 VSSIO	M12 VSSIO	M13 VSSIO
N1 TMS_ICE	N2 TEST[4]	N3 TEST[9]	N4 TDI	N5 VSSIO						N11 VSSIO	N12 VSSIO	N13 VSSIO
P1 64EN/	P2 NC	P3 BZRSET	P4 ENUM/	P5 VDDIO						P11 VSSIO	P12 VSSIO	P13 VSSIO
R1 SWITCH	R2 NC	R3 VSSIO	R4 BZVDD	R5 VDDC						R11 VSSIO	R12 VSSIO	R13 VSSIO
T1 HOT-SWAPEN/	T2 INTB/	T3 VDDIO	T4 INTA/	T5 VSSC						T11 VSSIO	T12 VSSIO	T13 VSSIO
U1 V5PCIX	U2 V5PCIX	U3 RST/	U4 GNT/	U5 VSSIO								
V1 PCICLK	V2 REQ/	V3 AD[31]	V4 VSSC	V5 VDDIO								
W1 AD[30]	W2 AD[29]	W3 VSSIO	W4 AD[22]	W5 AD[28]								
Y1 AD[27]	Y2 AD[26]	Y3 VDDIO	Y4 AD[25]	Y5 VSSC								
AA1 VDDC	AA2 C_BE[3]/	AA3 IDSEL	AA4 AD[23]	AA5 VSSIO								
AB1 AD[24]	AB2 AD[21]	AB3 VDDIO	AB4 VDDC	AB5 VDDIO	AB6 VSSIO	AB7 VDDC	AB8 VSSC	AB9 VDDIO	AB10 VSSIO	AB11 VDDC	AB12 VSSC	AB13 VDDIO
AC1 AD[20]	AC2 V5PCIX	AC3 AD[19]	AC4 VSSIO	AC5 VSSC	AC6 DEVSEL/	AC7 V5PCIX	AC8 AD[15]	AC9 VDDC	AC10 AD[8]	AC11 AD[7]	AC12 AD[4]	AC13 AD[0]
AD1 V5PCIX	AD2 AD[18]	AD3 PCIPLLVDD	AD4 PCIPLLVSS	AD5 VDDIO	AD6 STOP/	AD7 VSSIO	AD8 VDDIO	AD9 AD[12]	AD10 C_BE[0]/	AD11 VSSIO	AD12 VDDIO	AD13 AD[2]
AE1 VSSIO	AE2 V5PCIX	AE3 AD[16]	AE4 C_BE[2]/	AE5 IRDY/	AE6 PERR/	AE7 PAR	AE8 AD[14]	AE9 AD[11]	AE10 AD[9]	AE11 NC	AE12 AD[6]	AE13 AD[3]
AF1 VDDIO	AF2 VSSIO	AF3 FRAME/	AF4 TRDY/	AF5 SERR/	AF6 AD[17]	AF7 C_BE[1]/	AF8 AD[13]	AF9 V5PCIX	AF10 AD[10]	AF11 V5PCIX	AF12 V5PCIX	AF13 AD[5]

Figure 7.4 LSIFC929XL 456-Pin PBGA Top View (Cont.)

A14 TXBVSS1	A15 TX1–	A16 NC	A17 RX1–	A18 RXVDD1	A19 RXLOS1	A20 MOD-DEF1[1]	A21 SDA	A22 MD[5]	A23 MD[8]	A24 MD[11]	A25 VSSIO	A26 VDDIO
B14 TXBVDD1	B15 TX1+	B16 NC	B17 RX1+	B18 RXVSS1	B19 FAULT1/	B20 ODIS1	B21 SCL	B22 MD[4]	B23 MD[7]	B24 MD[10]	B25 MD[12]	B26 VSSIO
C14 LIPRESET/	C15 VDDIO	C16 VSSIO	C17 VSSC	C18 VDDC	C19 VDDIO	C20 VSSIO	C21 MD[3]	C22 VSSC	C23 VDDIO	C24 MD[9]	C25 MD[13]	C26 MD[14]
D14 TXVDD1	D15 TXVSS1	D16 RXBVDD1	D17 RXBVSS1	D18 MOD-DEF1[0]	D19 BYPASS1/	D20 MD[1]	D21 MD[2]	D22 VDDC	D23 MD[6]	D24 VSSIO	D25 MD[15]	D26 MP[1]
E14 VDDIO	E15 VDDC	E16 MOD-DEF1[2]	E17 VSSIO	E18 VDDIO	E19 MP[0]	E20 MD[0]	E21 VSSIO	E22 VDDIO	E23 VSSC	E24 FLASHCS/	E25 VDDIO	E26 MOE[0]/
								F22 VSSIO	F23 ZZ	F24 RAMCS/	F25 MWE[0]/	F26 MOE[1]/
								G22 VDDC	G23 MWE[1]/	G24 VDDIO	G25 BWE[0]/	G26 BWE[1]/
								H22 VSSC	H23 BWE[2]/	H24 VSSIO	H25 BWE[3]/	H26 ADV/
								J22 VDDIO	J23 VDDC	J24 MP[2]	J25 ADSC/	J26 MCLK
								K22 VSSIO	K23 MD[16]	K24 MD[17]	K25 MD[18]	K26 MD[19]
L14 VSSIO	L15 VSSIO	L16 VSSIO						L22 VSSC	L23 MD[20]	L24 VDDIO	L25 MD[21]	L26 MD[22]
M14 VSSIO	M15 VSSIO	M16 VSSIO						M22 VDDC	M23 MD[23]	M24 VSSIO	M25 MD[24]	M26 MD[25]
N14 VSSIO	N15 VSSIO	N16 VSSIO						N22 VDDIO	N23 MD[26]	N24 MD[27]	N25 MD[28]	N26 MD[29]
P14 VSSIO	P15 VSSIO	P16 VSSIO						P22 VSSIO	P23 MD[30]	P24 MD[31]	P25 MP[3]	P26 VSSC
R14 VSSIO	R15 VSSIO	R16 VSSIO						R22 MA[0]	R23 MA[1]	R24 VDDIO	R25 MA[2]	R26 MA[3]
T14 VSSIO	T15 VSSIO	T16 VSSIO						T22 VDDC	T23 MA[4]	T24 VSSIO	T25 MA[5]	T26 MA[6]
								U22 VDDIO	U23 MA[7]	U24 MA[8]	U25 MA[9]	U26 MA[10]
								V22 VSSIO	V23 VSSC	V24 MA[11]	V25 MA[12]	V26 MA[13]
								W22 VDDC	W23 MA[14]	W24 VDDIO	W25 MA[15]	W26 MA[16]
								Y22 VSSC	Y23 AD[32]	Y24 VSSIO	Y25 MA[17]	Y26 MA[18]
								AA22 VDDIO	AA23 AD[33]	AA24 NC	AA25 MA[19]	AA26 AD[35]
AB14 VSSIO	AB15 VSSC	AB16 VDDC	AB17 VDDIO	AB18 VSSIO	AB19 VSSC	AB20 VDDC	AB21 VDDIO	AB22 VSSIO	AB23 VDDC	AB24 AD[34]	AB25 VSSIO	AB26 MA[20]
AC14 C_BE[7]/	AC15 AD[1]	AC16 V5PCIX	AC17 AD[60]	AC18 VDDC	AC19 AD[55]	AC20 AD[51]	AC21 AD[46]	AC22 VSSC	AC23 V5PCIX	AC24 VDDIO	AC25 AD[36]	AC26 MA[21]
AD14 REQ64/	AD15 VSSIO	AD16 VDDIO	AD17 AD[61]	AD18 AD[57]	AD19 VSSIO	AD20 VDDIO	AD21 AD[48]	AD22 AD[44]	AD23 VSSIO	AD24 AD[37]	AD25 AD[39]	AD26 AD[38]
AE14 ACK64/	AE15 C_BE[5]/	AE16 AD[63]	AE17 AD[62]	AE18 AD[58]	AE19 AD[54]	AE20 AD[52]	AE21 AD[49]	AE22 AD[45]	AE23 AD[42]	AE24 AD[41]	AE25 AD[40]	AE26 VDDIO
AF14 C_BE[4]/	AF15 C_BE[6]/	AF16 PAR64	AF17 V5PCIX	AF18 AD[59]	AF19 AD[56]	AF20 AD[53]	AF21 AD[50]	AF22 AD[47]	AF23 AD[43]	AF24 V5PCIX	AF25 VDDIO	AF26 VSSIO

Table 7.14 Alphanumeric Pad Listing by PBGA Position

Ball #	Signal	Ball #	Signal	Ball #	Signal	Ball #	Signal	Ball #	Signal
A1	VSSIO	C9	VDDC	E17	VSSIO	L1	TEST[3]	P23	MD[30]
A2	VDDIO	C10	VSSC	E18	VDDIO	L2	TCK	P24	MD[31]
A3	NC[1]	C11	VDDIO	E19	MP[0]	L3	VSSIO	P25	MP[3]
A4	MODE[7]	C12	VSSIO	E20	MD[0]	L4	TMS_CHIP	P26	VSSC
A5	MODE[5]	C13	RTRIM	E21	VSSIO	L5	VSSC	R1	SWITCH
A6	TEST[0]	C14	LIPRESET/	E22	VDDIO	L11	VSSIO	R2	NC
A7	MODDEF0[1]	C15	VDDIO	E23	VSSC	L12	VSSIO	R3	VSSIO
A8	RXLOS0	C16	VSSIO	E24	FLASHCS/	L13	VSSIO	R4	BZVDD
A9	TXVDD0	C17	VSSC	E25	VDDIO	L14	VSSIO	R5	VDDC
A10	TX0–	C18	VDDC	E26	MOE[0]/	L15	VSSIO	R11	VSSIO
A11	NC	C19	VDDIO	F1	TEST[5]	L16	VSSIO	R12	VSSIO
A12	RX0-	C20	VSSIO	F2	NC	L22	VSSC	R13	VSSIO
A13	RXBVSS0	C21	MD[3]	F3	TEST[6]	L23	MD[20]	R14	VSSIO
A14	TXBVSS1	C22	VSSC	F4	NC	L24	VDDIO	R15	VSSIO
A15	TX1–	C23	VDDIO	F5	VDDIO	L25	MD[21]	R16	VSSIO
A16	NC	C24	MD[9]	F22	VSSIO	L26	MD[22]	R22	MA[0]
A17	RX1–	C25	MD[13]	F23	ZZ	M1	NC	R23	MA[1]
A18	RXVDD1	C26	MD[14]	F24	RAMCS/	M2	TRST/	R24	VDDIO
A19	RXLOS1	D1	NC	F25	MWE[0]/	M3	VDDIO	R25	MA[2]
A20	MODDEF1[1]	D2	REFPLLVDD	F26	MOE[1]/	M4	TDO	R26	MA[3]
A21	SDA	D3	VDDIO	G1	NC	M5	VDDC	T1	HOTSWAPEN/
A22	MD[5]	D4	MODE[4]	G2	NC	M11	VSSIO	T2	INTB/
A23	MD[8]	D5	VDDC	G3	VSSIO	M12	VSSIO	T3	VDDIO
A24	MD[11]	D6	LED[2]/	G4	REFPLLVSS	M13	VSSIO	T4	INTA/
A25	VSSIO	D7	LED[0]/	G5	VDDC	M14	VSSIO	T5	VSSC
A26	VDDIO	D8	BYPASS0/	G22	VDDC	M15	VSSIO	T11	VSSIO
B1	VDDIO	D9	MODDEF0[2]	G23	MWE[1]/	M16	VSSIO	T12	VSSIO
B2	MODE[3]	D10	TXBVSS0	G24	VDDIO	M22	VDDC	T13	VSSIO
B3	LED[4]/	D11	TXBVDD0	G25	BWE[0]/	M23	MD[23]	T14	VSSIO
B4	LED[1]/	D12	RXVSS0	G26	BWE[1]/	M24	VSSIO	T15	VSSIO
B5	TEST[1]	D13	RXVDD0	H1	NC	M25	MD[24]	T16	VSSIO
B6	NC	D14	TXVDD1	H2	GPIO[3]	M26	MD[25]	T22	VDDC
B7	ODIS0	D15	TXVSS1	H3	VDDIO	N1	TMS_ICE	T23	MA[4]
B8	FAULT0/	D16	RXBVDD1	H4	VSSC	N2	TEST[4]	T24	VSSIO
B9	TXVSS0	D17	RXBVSS1	H5	TEST[7]	N3	TEST[9]	T25	MA[5]
B10	TX0+	D18	MODDEF1[0]	H22	VSSC	N4	TDI	T26	MA[6]
B11	NC	D19	BYPASS1/	H23	BWE[2]/	N5	VSSIO	U1	V5PCIX
B12	RX0+	D20	MD[1]	H24	VSSIO	N11	VSSIO	U2	V5PCIX
B13	RXBVDD0	D21	MD[2]	H25	BWE[3]/	N12	VSSIO	U3	RST/
B14	TXBVDD1	D22	VDDC	H26	ADV/	N13	VSSIO	U4	GNT/
B15	TX1+	D23	MD[6]	J1	GPIO[0]	N14	VSSIO	U5	VSSIO
B16	NC	D24	VSSIO	J2	GPIO[5]	N15	VSSIO	U22	VDDIO
B17	RX1+	D25	MD[15]	J3	GPIO[1]	N16	VSSIO	U23	MA[7]
B18	RXVSS1	D26	MP[1]	J4	VDDC	N22	VDDIO	U24	MA[8]
B19	FAULT1/	E1	MODE[2]	J5	VSSIO	N23	MD[26]	U25	MA[9]
B20	ODIS1	E2	VSSIO	J22	VDDIO	N24	MD[27]	U26	MA[10]
B21	SCL	E3	MODE[0]	J23	VDDC	N25	MD[28]	V1	PCICLK
B22	MD[4]	E4	VSSC	J24	MP[2]	N26	MD[29]	V2	REQ/
B23	MD[7]	E5	VSSIO	J25	ADSC/	P1	64EN/	V3	AD[31]
B24	MD[10]	E6	VDDIO	J26	MCLK	P2	NC	V4	VSSC
B25	MD[12]	E7	LED[3]/	K1	GPIO[2](BLUELED/)	P3	BZRSET	V5	VDDIO
B26	VSSIO	E8	TEST[2]	K2	NC	P4	ENUM/	V22	VSSIO
C1	REFCLK	E9	VSSIO	K3	GPIO[4]	P5	VDDIO	V23	VSSC
C2	MODE[1]	E10	VDDIO	K4	NC	P11	VSSIO	V24	MA[11]
C3	MODE[6]	E11	MODDEF0[0]	K5	VDDIO	P12	VSSIO	V25	MA[12]
C4	VSSIO	E12	VSSC	K22	VSSIO	P13	VSSIO	V26	MA[13]
C5	VSSC	E13	VSSIO	K23	MD[16]	P14	VSSIO	W1	AD[30]
C6	TEST[8]	E14	VDDIO	K24	MD[17]	P15	VSSIO	W2	AD[29]
C7	VDDIO	E15	VDDC	K25	MD[18]	P16	VSSIO	W3	VSSIO
C8	VSSIO	E16	MODDEF1[2]	K26	MD[19]	P22	VSSIO		

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
W4	AD[22]	AB6	VSSIO	AC12	AD[4]	AD18	AD[57]	AE24	AD[41]
W5	AD[28]	AB7	VDDC	AC13	AD[0]	AD19	VSSIO	AE25	AD[40]
W22	VDDC	AB8	VSSC	AC14	C_BE[7]/	AD20	VDDIO	AE26	VDDIO
W23	MA[14]	AB9	VDDIO	AC15	AD[1]	AD21	AD[48]	AF1	VDDIO
W24	VDDIO	AB10	VSSIO	AC16	V5PCIX	AD22	AD[44]	AF2	VSSIO
W25	MA[15]	AB11	VDDC	AC17	AD[60]	AD23	VSSIO	AF3	FRAME/
W26	MA[16]	AB12	VSSC	AC18	VDDC	AD24	AD[37]	AF4	TRDY/
Y1	AD[27]	AB13	VDDIO	AC19	AD[55]	AD25	AD[39]	AF5	SERR/
Y2	AD[26]	AB14	VSSIO	AC20	AD[51]	AD26	AD[38]	AF6	AD[17]
Y3	VDDIO	AB15	VSSC	AC21	AD[46]	AE1	VSSIO	AF7	C_BE[1]/
Y4	AD[25]	AB16	VDDC	AC22	VSSC	AE2	V5PCIX	AF8	AD[13]
Y5	VSSC	AB17	VDDIO	AC23	V5PCIX	AE3	AD[16]	AF9	V5PCIX
Y22	VSSC	AB18	VSSIO	AC24	VDDIO	AE4	C_BE[2]/	AF10	AD[10]
Y23	AD[32]	AB19	VSSC	AC25	AD[36]	AE5	IRDY/	AF11	V5PCIX
Y24	VSSIO	AB20	VDDC	AC26	MA[21]	AE6	PERR/	AF12	V5PCIX
Y25	MA[17]	AB21	VDDIO	AD1	V5PCIX	AE7	PAR	AF13	AD[5]
Y26	MA[18]	AB22	VSSIO	AD2	AD[18]	AE8	AD[14]	AF14	C_BE[4]/
AA1	VDDC	AB23	VDDC	AD3	PCIPLLVDD	AE9	AD[11]	AF15	C_BE[6]/
AA2	C_BE[3]/	AB24	AD[34]	AD4	PCIPLLVSS	AE10	AD[9]	AF16	PAR64
AA3	IDSEL	AB25	VSSIO	AD5	VDDIO	AE11	NC	AF17	V5PCIX
AA4	AD[23]	AB26	MA[20]	AD6	STOP/	AE12	AD[6]	AF18	AD[59]
AA5	VSSIO	AC1	AD[20]	AD7	VSSIO	AE13	AD[3]	AF19	AD[56]
AA22	VDDIO	AC2	V5PCIX	AD8	VDDIO	AE14	ACK64/	AF20	AD[53]
AA23	AD[33]	AC3	AD[19]	AD9	AD[12]	AE15	C_BE[5]/	AF21	AD[50]
AA24	NC[1]	AC4	VSSIO	AD10	C_BE[0]/	AE16	AD[63]	AF22	AD[47]
AA25	MA[19]	AC5	VSSC	AD11	VSSIO	AE17	AD[62]	AF23	AD[43]
AA26	AD[35]	AC6	DEVSEL/	AD12	VDDIO	AE18	AD[58]	AF24	V5PCIX
AB1	AD[24]	AC7	V5PCIX	AD13	AD[2]	AE19	AD[54]	AF25	VDDIO
AB2	AD[21]	AC8	AD[15]	AD14	REQ64/	AE20	AD[52]	AF26	VSSIO
AB3	VDDIO	AC9	VDDC	AD15	VSSIO	AE21	AD[49]		
AB4	VDDC	AC10	AD[8]	AD16	VDDIO	AE22	AD[45]		
AB5	VDDIO	AC11	AD[7]	AD17	AD[61]	AE23	AD[42]		

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
64EN/	P1	AD[58]	AE18	MA[12]	V25	MP[1]	D26	TDI	N4
ACK64/	AE14	AD[59]	AF18	MA[13]	V26	MP[2]	J24	TDO	M4
AD[0]	AC13	AD[60]	AC17	MA[14]	W23	MP[3]	P25	TEST[0]	A6
AD[1]	AC15	AD[61]	AD17	MA[15]	W25	MWE[1]/	G23	TEST[1]	B5
AD[2]	AD13	AD[62]	AE17	MA[16]	W26	MWE[0]/	F25	TEST[2]	E8
AD[3]	AE13	AD[63]	AE16	MA[17]	Y25	NC[1]	A3	TEST[3]	L1
AD[4]	AC12	ADSC/	J25	MA[18]	Y26	NC	A11	TEST[4]	N2
AD[5]	AF13	ADV/	H26	MA[19]	AA25	NC	A16	TEST[5]	F1
AD[6]	AE12	BWE[0]/	G25	MA[20]	AB26	NC	B6	TEST[6]	F3
AD[7]	AC11	BWE[1]/	G26	MA[21]	AC26	NC	B11	TEST[7]	H5
AD[8]	AC10	BWE[2]/	H23	MCLK	J26	NC	B16	TEST[8]	C6
AD[9]	AE10	BWE[3]/	H25	MD[0]	E20	NC	D1	TEST[9]	N3
AD[10]	AF10	BYPASS0/	D8	MD[1]	D20	NC	F2	TMS_CHIP	L4
AD[11]	AE9	BYPASS1/	D19	MD[2]	D21	NC	F4	TMS_ICE	N1
AD[12]	AD9	BZRSET	P3	MD[3]	C21	NC	G1	TRDY/	AF4
AD[13]	AF8	BZVDD	R4	MD[4]	B22	NC	G2	TRST/	M2
AD[14]	AE8	C_BE[0]/	AD10	MD[5]	A22	NC	H1	TX0-	A10
AD[15]	AC8	C_BE[1]/	AF7	MD[6]	D23	NC	K2	TX0+	B10
AD[16]	AE3	C_BE[2]/	AE4	MD[7]	B23	NC	K4	TX1-	A15
AD[17]	AF6	C_BE[3]/	AA2	MD[8]	A23	NC	M1	TX1+	B15
AD[18]	AD2	C_BE[4]/	AF14	MD[9]	C24	NC	P2	TXBVDD0	D11
AD[19]	AC3	C_BE[5]/	AE15	MD[10]	B24	NC	R2	TXBVDD1	B14
AD[20]	AC1	C_BE[6]/	AF15	MD[11]	A24	NC	AA24	TXBVSS0	D10
AD[21]	AB2	C_BE[7]/	AC14	MD[12]	B25	NC	AE11	TXBVSS1	A14
AD[22]	W4	DEVSEL/	AC6	MD[13]	C25	ODIS0	B7	TXVDD0	A9
AD[23]	AA4	ENUM/	P4	MD[14]	C26	ODIS1	B20	TXVDD1	D14
AD[24]	AB1	FAULT0/	B8	MD[15]	D25	PAR	AE7	TXVSS0	B9
AD[25]	Y4	FAULT1/	B19	MD[16]	K23	PAR64	AF16	TXVSS1	D15
AD[26]	Y2	FLASHCS/	E24	MD[17]	K24	PCICLK	V1	V5PCIX	U1
AD[27]	Y1	FRAME/	AF3	MD[18]	K25	PCIPLLVDD	AD3	V5PCIX	U2
AD[28]	W5	GNT/	U4	MD[19]	K26	PCIPLLVSS	AD4	V5PCIX	AC2
AD[29]	W2	GPIO[0]	J1	MD[20]	L23	PERR/	AE6	V5PCIX	AC7
AD[30]	W1	GPIO[1]	J3	MD[21]	L25	RAMCS/	F24	V5PCIX	AC16
AD[31]	V3	GPIO[2](BLUELED/)	K1	MD[22]	L26	REFCLK	C1	V5PCIX	AC23
AD[32]	Y23	GPIO[3]	H2	MD[23]	M23	REFPLLVDD	D2	V5PCIX	AD1
AD[33]	AA23	GPIO[4]	K3	MD[24]	M25	REFPLLVSS	G4	V5PCIX	AE2
AD[34]	AB24	GPIO[5]	J2	MD[25]	M26	REQ/	V2	V5PCIX	AF9
AD[35]	AA26	HOTSWAPEN/	T1	MD[26]	N23	REQ64/	AD14	V5PCIX	AF11
AD[36]	AC25	IDSEL	AA3	MD[27]	N24	RST/	U3	V5PCIX	AF12
AD[37]	AD24	INTA/	T4	MD[28]	N25	RTRIM	C13	V5PCIX	AF17
AD[38]	AD26	INTB/	T2	MD[29]	N26	RX0–	A12	V5PCIX	AF24
AD[39]	AD25	IRDY/	AE5	MD[30]	P23	RX0+	B12	VDDC	C9
AD[40]	AE25	LED[0]/	D7	MD[31]	P24	RX1–	A17	VDDC	C18
AD[41]	AE24	LED[1]/	B4	MODDEF0[0]	E11	RX1+	B17	VDDC	D5
AD[42]	AE23	LED[2]/	D6	MODDEF0[1]	A7	RXBVDD0	B13	VDDC	D22
AD[43]	AF23	LED[3]/	E7	MODDEF0[2]	D9	RXBVDD1	D16	VDDC	E15
AD[44]	AD22	LED[4]/	B3	MODDEF1[0]	D18	RXBVSS0	A13	VDDC	G5
AD[45]	AE22	LIPRESET/	C14	MODDEF1[1]	A20	RXBVSS1	D17	VDDC	G22
AD[46]	AC21	MA[0]	R22	MODDEF1[2]	E16	RXLOS0	A8	VDDC	J4
AD[47]	AF22	MA[1]	R23	MODE[0]	E3	RXLOS1	A19	VDDC	J23
AD[48]	AD21	MA[2]	R25	MODE[1]	C2	RXVDD0	D13	VDDC	M5
AD[49]	AE21	MA[3]	R26	MODE[2]	E1	RXVDD1	A18	VDDC	M22
AD[50]	AF21	MA[4]	T23	MODE[3]	B2	RXVSS0	D12	VDDC	R5
AD[51]	AC20	MA[5]	T25	MODE[4]	D4	RXVSS1	B18	VDDC	T22
AD[52]	AE20	MA[6]	T26	MODE[5]	A5	SCL	B21	VDDC	W22
AD[53]	AF20	MA[7]	U23	MODE[6]	C3	SDA	A21	VDDC	AA1
AD[54]	AE19	MA[8]	U24	MODE[7]	A4	SERR/	AF5	VDDC	AB4
AD[55]	AC19	MA[9]	U25	MOE[0]/	E26	STOP/	AD6	VDDC	AB7
AD[56]	AF19	MA[10]	U26	MOE[1]/	F26	SWITCH	R1	VDDC	AB11
AD[57]	AD18	MA[11]	V24	MP[0]	E19	TCK	L2		

1. *NC* pins are not connected.

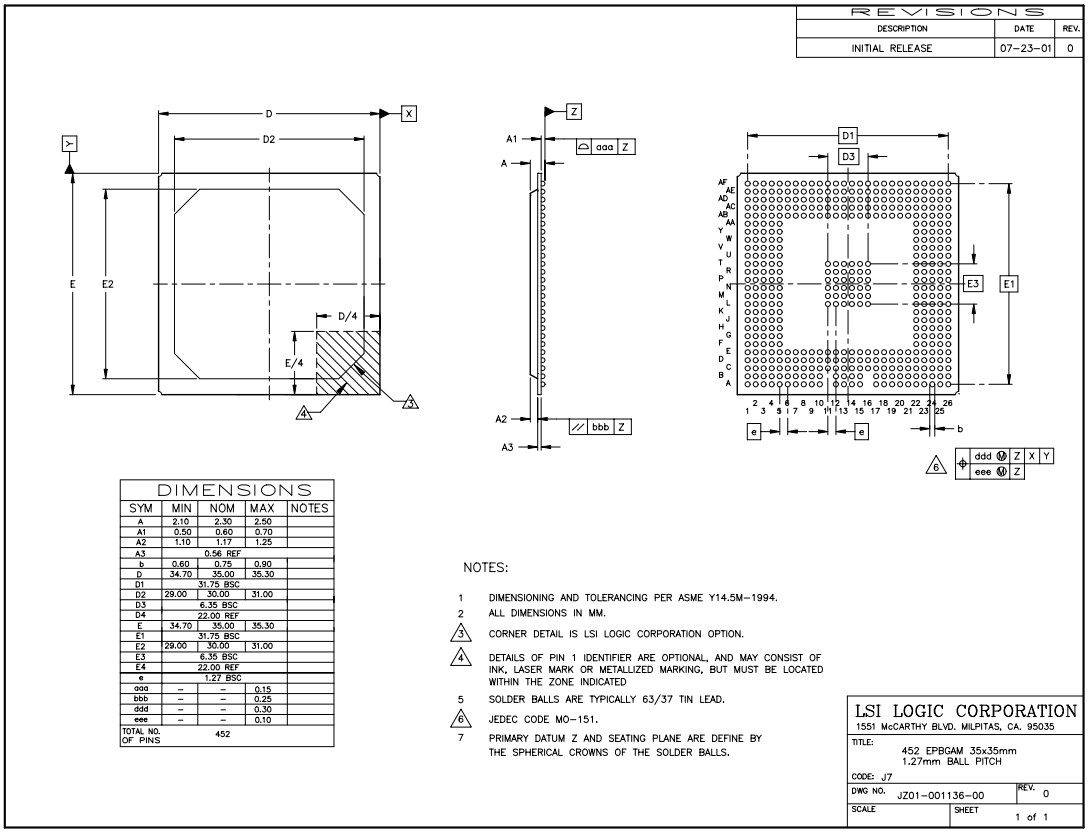
Table 7.15 Alphanumeric Pad Listing by Signal Name (Cont.)

Signal	Ball #	Signal	Ball #	Signal	Ball #	Signal	Ball #	Signal	Ball #
VDDC	AB16	VDDIO	V5	VSSC	V4	VSSIO	L12	VSSIO	R16
VDDC	AB20	VDDIO	W24	VSSC	V23	VSSIO	L13	VSSIO	T11
VDDC	AB23	VDDIO	Y3	VSSC	Y5	VSSIO	L14	VSSIO	T12
VDDC	AC9	VDDIO	AA22	VSSC	Y22	VSSIO	L15	VSSIO	T13
VDDC	AC18	VDDIO	AB3	VSSC	AB8	VSSIO	L16	VSSIO	T14
VDDIO	A2	VDDIO	AB5	VSSC	AB12	VSSIO	M11	VSSIO	T15
VDDIO	A26	VDDIO	AB9	VSSC	AB15	VSSIO	M12	VSSIO	T16
VDDIO	B1	VDDIO	AB13	VSSC	AB19	VSSIO	M13	VSSIO	T24
VDDIO	C7	VDDIO	AB17	VSSC	AC5	VSSIO	M14	VSSIO	U5
VDDIO	C11	VDDIO	AB21	VSSC	AC22	VSSIO	M15	VSSIO	V22
VDDIO	C15	VDDIO	AC24	VSSIO	A1	VSSIO	M16	VSSIO	W3
VDDIO	C19	VDDIO	AD5	VSSIO	A25	VSSIO	M24	VSSIO	Y24
VDDIO	C23	VDDIO	AD8	VSSIO	B26	VSSIO	N5	VSSIO	AA5
VDDIO	D3	VDDIO	AD12	VSSIO	C4	VSSIO	N11	VSSIO	AB6
VDDIO	E6	VDDIO	AD16	VSSIO	C8	VSSIO	N12	VSSIO	AB10
VDDIO	E10	VDDIO	AD20	VSSIO	C12	VSSIO	N13	VSSIO	AB14
VDDIO	E14	VDDIO	AE26	VSSIO	C16	VSSIO	N14	VSSIO	AB18
VDDIO	E18	VDDIO	AF1	VSSIO	C20	VSSIO	N15	VSSIO	AB22
VDDIO	E22	VDDIO	AF25	VSSIO	D24	VSSIO	N16	VSSIO	AB25
VDDIO	E25	VSSC	C5	VSSIO	E2	VSSIO	P11	VSSIO	AC4
VDDIO	F5	VSSC	C10	VSSIO	E5	VSSIO	P12	VSSIO	AD7
VDDIO	G24	VSSC	C17	VSSIO	E9	VSSIO	P13	VSSIO	AD11
VDDIO	H3	VSSC	C22	VSSIO	E13	VSSIO	P14	VSSIO	AD15
VDDIO	J22	VSSC	E4	VSSIO	E17	VSSIO	P15	VSSIO	AD19
VDDIO	K5	VSSC	E12	VSSIO	E21	VSSIO	P16	VSSIO	AD23
VDDIO	L24	VSSC	E23	VSSIO	F22	VSSIO	P22	VSSIO	AE1
VDDIO	M3	VSSC	H4	VSSIO	G3	VSSIO	R3	VSSIO	AF2
VDDIO	N22	VSSC	H22	VSSIO	H24	VSSIO	R11	VSSIO	AF26
VDDIO	P5	VSSC	L5	VSSIO	J5	VSSIO	R12	ZZ	F23
VDDIO	R24	VSSC	L22	VSSIO	K22	VSSIO	R13		
VDDIO	T3	VSSC	P26	VSSIO	L3	VSSIO	R14		
VDDIO	U22	VSSC	T5	VSSIO	L11	VSSIO	R15		

7.4 Mechanical Drawing

Figure 7.5 shows the mechanical drawing for the 456-pad PBGA. This package actually contains only 452 pads, as the pads for A11, B11, A16, and B16 have been removed from this package to improve device performance.

Figure 7.5 456-Pad Plastic Ball Grid Array



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 J7.**

7.5 Package Thermal Considerations

Package thermal management is an important element of electronic product design. In any system environment, it is important not to exceed the maximum recommended semiconductor junction temperature. The maximum recommended junction temperature for the LSIFC929XL is 115 ˚C.

To that end, LSI Logic recommends using an appropriate heat sink for the LSIFC929XL and maintaining an adequate airflow throughout the system.

Allowing for a maximum dynamic power consumption of 4 W, Table 7.16 shows some examples of the maximum allowable junction temperature to maintain less than a 70 ˚C ambient for the given airflow and heat sink conditions.

Table 7.16 Maximum Allowable Ambient Temperature vs. Airflow

	Airflow (m/s)	θ_{JMAmax} (˚C/W)	Junction Temperature at 70 ˚C Ambient (˚C)	Maximum Allowable Ambient Temperature (˚C)
With Heat Sink	0	12.2	118.8	66.2
	0.5	10.1	110.4	74.6
	1.0	8.7	104.8	80.2
	2.0	6.9	97.6	87.4
	3.0	6.2	94.8	90.2
Without Heat Sink	0	15.3	131.2	53.8
	0.5	13.4	123.6	61.4
	1.0	12.8	121.2	63.8
	2.0	11.9	117.6	67.4
	3.0	11.1	114.4	70.6

Appendix A
Register Summary

Table A.1 and Table A.2 list the register summary for the LSIFC929XL.

Table A.1 LSIFC929XL 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	6-11
Memory[0] Base Address Low	0x14	Read/Write	6-11
Memory[0] Base Address High	0x18	Read/Write	6-12
Memory[1] Base Address Low	0x1C	Read/Write	6-12
Memory[1] Base Address High	0x20	Read/Write	6-13
Reserved	0x24–0x28	Read Only	6-13
Subsystem Vendor ID	0x2C	Read Only	6-14
Subsystem ID	0x2E	Read Only	6-14
Expansion ROM Base Address	0x30	Read/Write	6-15
Capabilities Pointer	0x34	Read Only	6-16
Reserved	0x38	Read Only	6-17
Interrupt Line	0x3C	Read/Write	6-17
Interrupt Pin	0x3D	Read Only	6-17

Table A.1 LSIFC929XL Multifunction PCI Registers (Cont.)

Register Name	Address	Read/Write	Page
Minimum Grant	0x3E	Read Only	6-18
Minimum Latency	0x3F	Read Only	6-18
Power Management Capability ID	0xXX	Read Only	6-19
Power Management Next Pointer	0xXX	Read Only	6-19
Power Management Capabilities	0xXX	Read Only	6-20
Power Management Control/Status	0xXX	Read/Write	6-21
Power Management Bridge Support Extensions	0xXX	Read Only	6-22
Power Management Data	0xXX	Read Only	6-22
MSI Capability ID	0xXX	Read Only	6-22
MSI Next Pointer	0xXX	Read Only	6-23
Message Control	0xXX	Read/Write	6-23
Message Address	0xXX	Read/Write	6-25
Message Upper Address	0xXX	Read/Write	6-25
Message Data	0xXX	Read/Write	6-26
PCI-X Capability ID	0xXX	Read Only	6-26
PCI-X Next Pointer	0xXX	Read Only	6-27
PCI-X Command	0xXX	Read/Write	6-27
PCI-X Status	0xXX	Read/Write	6-29

Table A.2 LSIFC929XL Host Interface Registers

Register Name	Address	Read/Write	Page
System Doorbell	0x00	Read/Write	6-33
Write Sequence	0x04	Read/Write	6-34
Host Diagnostic	0x08	Read/Write	6-35
Test Base Address	0x0C	Read/Write	6-37
Diagnostic Read/Write Data	0x10	Read/Write	6-38
Diagnostic Read/Write Address	0x14	Read/Write	6-39
Host Interrupt Status	0x30	Read Only	6-40
Host Interrupt Mask	0x34	Read/Write	6-41

Table A.2 LSIFC929XL Host Interface Registers (Cont.)

Register Name	Address	Read/Write	Page
Request FIFO	0x40	Read/Write	6-42
Reply FIFO	0x44	Read/Write	6-43
Host Index Register	0x50	Read/Write	6-43

Appendix B Reference Specifications

The LSIFC929XL is compliant with the following specifications:

Table B.1 Reference Specifications

Specification	Revision
Fibre Channel Physical Interface (FC-PI)	11
Fibre Channel Physical and Signaling Interface (FC-PH)	4.3
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
GBIC	5.4
PCI Local Bus	2.2
PCI-X Addendum to the PCI Local Bus	1.0a

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 eight 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	“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	“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 LSIFC929XL 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

D

E

F

R

S

T

U

V

W

Z

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: *LSIFC929XL 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 _____