IDT®
Tsi620 RapidIO Switch / RapidIO-to-PCI Bridge

User Reference Manual

June 4, 2013

Formal Status

Contents

26. SREP Registers 717

Figures

Tables

About this Document

This section discusses the following topics:

Scope

The *Tsi620 Evaluation Board User Manual* discusses the features, capabilities, and configuration requirements for the Tsi620. It is intended for hardware and software engineers who are designing system interconnect applications with the device.

Organization

This document is organized into six parts:

Conventions

This document uses the following conventions.

Non-differential Signal Notation

Non-differential signals are either active-low or active-high. An active-low signal has an active state of logic 0 (or the lower voltage level), and is denoted by a lowercase “n”. An active-high signal has an active state of logic 1 (or the higher voltage level), and is not denoted by a special character. The following table shows the non-differential signal naming convention.

State	Single-line signal	Multi-line signal
Active low	NAMEn	NAMEn[3]
Active high	NAME	NAME[3]

Differential Signal Notation

Differential signals consist of pairs of complement positive and negative signals that are measured at the same time to determine a signal’s active or inactive state (they are denoted by “_p” and “_n”, respectively). The following table shows the differential signal naming convention.

State	Single-line signal	Multi-line signal
Inactive	NAME_p = 0 NAME_n = 1	NAME_p[3] = 0 NAME_n[3] =1
Active	NAME_p = 1 NAME_n = 0	NAME_p[3] is 1 NAME_n[3] is 0

Object Size Notation

- A *byte* is an 8-bit object.
- A *word* is a 16-bit object.
- A *doubleword* (Dword) is a 32-bit object.

Numeric Notation

- Hexadecimal numbers are denoted by the prefix *0x* (for example, 0x04).
- Binary numbers are denoted by the prefix *0b* (for example, 0b010).
- Registers that have multiple iterations are denoted by {x..y} in their names; where *x* is first register and address, and *y* is the last register and address. For example, REG{0..1} indicates there are two versions of the register at different addresses: REG0 and REG1.

Symbols

This symbol indicates a basic design concept or information considered helpful.

This symbol indicates important configuration information or suggestions.

This symbol indicates procedures or operating levels that may result in misuse or damage to the device.

Document Status Information

- Advance – Contains information that is subject to change, and is available once prototypes are released to customers.
- Preliminary – Contains information about a product that is near production-ready, and is revised as required.
- Formal – Contains information about a final, customer-ready product, and is available once the product is released to production.

Revision History

June 4, 2013, Formal

- Updated the “Recommended Termination” information for various pins in Table 100
- Added a note to SP_TX_SWAP/SP_RX_SWAP description in Table 108
- Changed the Bridge ISF port number reference in “Base Address Register (BARs) and Lookup Table (LUT) Operation” to 0b1 from 0xC
- Completed other minor improvements throughout the document

80D7000_MA001_04, Formal, August 2009

- Changed the branding of the document to indicate Integrated Device Technology
- Added a new section that discusses “Port-writes and Multicast”
- Completed numerous minor updates throughout the document

80D7000_MA001_03, Formal, November 2008

- Added more information about “Software Assisted Error Recovery”
- Added “Performance Information” about the Tsi620 Switch and Tsi620 Bridge
- Added recommended termination information for Tsi620’s signals (see “Signal Descriptions”)
- Added “Power Consumption” information

- Removed descriptions of the Time-to-Live (TTL) feature for the Tsi620 Switch. The TTL feature, however, is still applicable to the SREP module (see "Packet TTL Expired").
- Added a caution statement regarding writing to the "RapidIO Host Base Device ID Lock CSR"
- Completed numerous minor updates throughout the document

80D7000_MA001_02, Preliminary, October 2007

- Added "Electrical Characteristics" information
- Added "Packaging" information

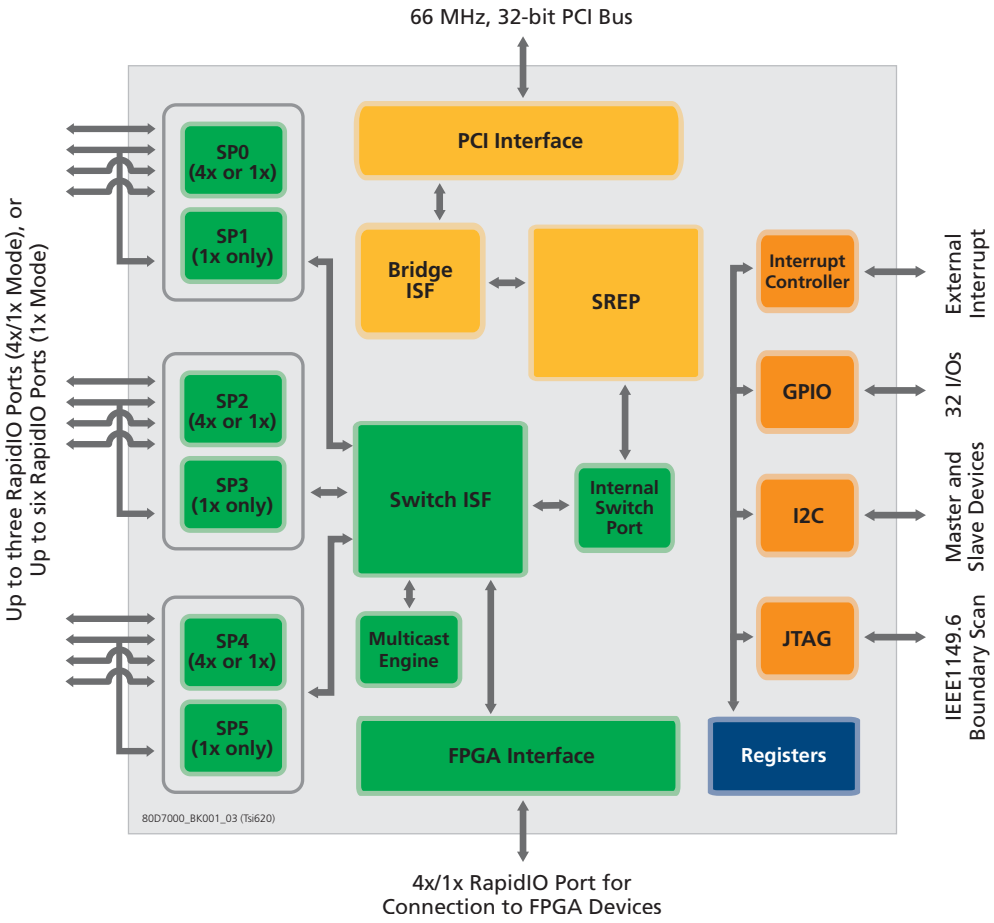
80D7000_MA001_01, Advance, July 2007

This is the first version of the *Tsi620 Evaluation Board User Manual*.

PART 1: OVERVIEW

The Tsi620 is a RapidIO Switch and a RapidIO-to-PCI Bridge. The switching component supports up to seven RapidIO devices, while the bridging functionality supports non-transparent transactions between a 32-bit PCI bus and a RapidIO fabric.

This part of the document provides an overview of the Tsi620's interfaces, features, and functions.



1. Functional Overview

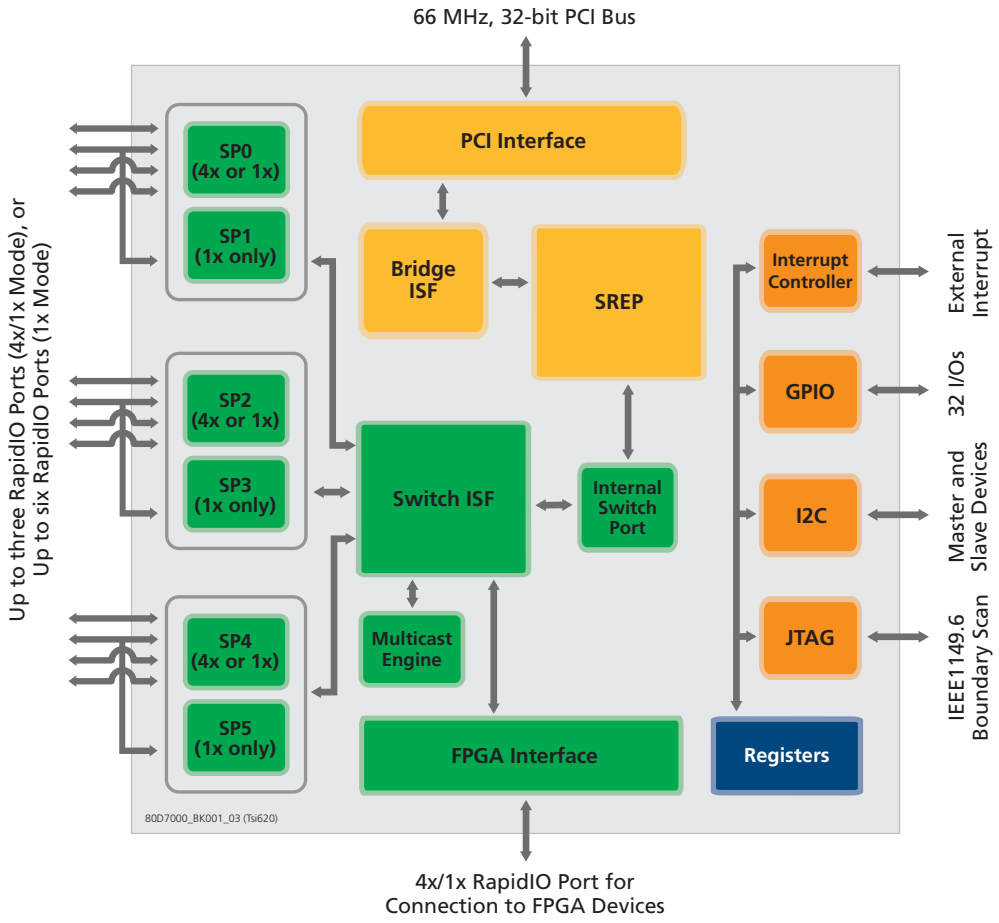
This chapter describes the main features and functions of the Tsi620. Topics discussed include the following:

1.1 Overview

The Tsi620 provides the functionality of both a serial RapidIO Switch and a non-transparent RapidIO-to-PCI bridge (see Figure 1). The RapidIO Switch offers 50 Gbps aggregate bandwidth, while the RapidIO-to-PCI bridge enables legacy systems to link to the high-bandwidth RapidIO interconnect. The Tsi620 contains all the benefits of IDT's family of "Tsi" RapidIO switches and adds interfaces to PCI-enabled processors, as well as an interface to low-cost FPGAs.

The Tsi620 allows system designers to develop applications with a variety of interfaces. The Tsi620 is optimized to reduce the overall Bill of Materials (BOM) for wireless, networking, storage, and military applications. It enables the use of low cost FPGAs and microprocessors that are typically used along with clusters of DSPs.

Figure 1: Tsi620 Block Diagram



1.2 RapidIO Switch (Tsi620 Switch)

- Software compatible with the IDT Tsi57x family of switches
- Full-duplex, line rate, non-blocking switching fabric (50 Gbps simplex bandwidth)
- Prevents head-of-line blocking on each port
- Eight packet buffers per ingress port
- Eight packet buffers per egress port
- One Multicast Engine provides dedicated multicast resources without impacting throughput on the ports.
  - Packets are replicated to each egress port in parallel.
  - The Multicast Engine can accept a bursts of traffic with different packet sizes.

— Arbitration at the egress port to allow management of resource contention between multicast or non-multicast traffic.

System behavior when multicasting of packets that require responses is not defined in the *RapidIO Interconnect Specification (Revision 1.3) - Part 11 Multicast Specification*.

1.3 RapidIO Interface

The Tsi620 contains a high-performance RapidIO Interface that provides connectivity for control plane and data plane applications. The device's RapidIO ports are compliant with the *RapidIO Interconnect Specification (Revision 1.3)*.

This document typically uses RapidIO Interface to refer to the Tsi620's 4x/1x RapidIO ports. This term, however, also applies to the device's FPGA Interface since this interface can be configured as a RapidIO port.

This section describes the transport layer features common to all Tsi620 RapidIO ports. For information on the electrical layer characteristics of the RapidIO ports, see "RapidIO Electrical Interface".

The RapidIO ports have the following capabilities:

- RapidIO packet and control symbol transmission
- RapidIO packet and control symbol reception
- Register access through RapidIO maintenance requests

1.3.1 Features

The following features are supported:

- Up to three 4x-mode or up to six 1x-mode RapidIO ports operating at up to 3.125 Gbps
- Per-port destination ID lookup table, used to direct packets through the switch

This is an IDT-specific design. The *RapidIO Interconnect Specification (Revision 1.3)* standard design of lookup tables is also supported.

- RapidIO error management extensions described in *RapidIO Interconnect Specification (Revision 1.3) Part 8*, including both hardware and software error recovery
- Low latency forwarding of the multicast control symbol
- Proprietary registers for performance monitoring and tuning
- Both cut-through and store-and-forward modes for performance tuning
- Debug packet generation and capture
- Multicast capability
- Head-of-line blocking avoidance

1.3.2 Transaction Flow Overview

Packets and control symbols are received by the Serial RapidIO Electrical Interface (Serial MAC) and forwarded to the RapidIO Interface (for more information on the Serial MAC, see "RapidIO Electrical Interface"). Received packets have their integrity verified by error checking defined by *RapidIO Interconnect Specification (Revision 1.3)*. Once the packet's integrity is verified, the destination ID of the packet is used to access the routing lookup table to determine which port the packet should be forwarded to and whether the packet is a multicast packet. The packet is then buffered by the Switch ISF for transmission to the port. After the packet is transferred to the egress port, the port transmits the packet. If a packet fails the CRC check, the packet is discarded and the transmitter is instructed to retransmit the packet through the use of control symbols.

The egress port receives packets to be transmitted by the Switch ISF. The integrity of packets forwarded through the Switch ISF is retained by sending the CRC code received with the packet. For more information on the input and output queues, see "Packet Queuing".

The packet transmitter and the packet receiver cooperate to ensure that packets are never dropped (lost). A transmitter must retain a packet in its buffers until the port receives a packet accepted control symbol from the other end of the link.

1.3.3 Maintenance Requests

A maintenance packet is the only packet type that is modified by the RapidIO Switch. If the hop count value of the maintenance request is 0, the maintenance request is forwarded to the register bus for processing. The register bus accesses the registers in the appropriate port. The response to the maintenance request is compiled into a maintenance response packet and queued by the port for transmission. Maintenance packets with a non-zero hop count value have their hop count decremented, CRC recomputed, and are then forwarded to the port selected by the destination ID value in the lookup table.

1.3.4 Control Symbols

Control symbols received by the Tsi620 have their CRC validated, and their field values checked. If either the CRC is incorrect or the control symbol field values are incorrect, a packet-not-accepted control symbol is sent back and the control symbol is discarded. Otherwise, the control symbol is used by the port for packet management in the transmit port or link maintenance.

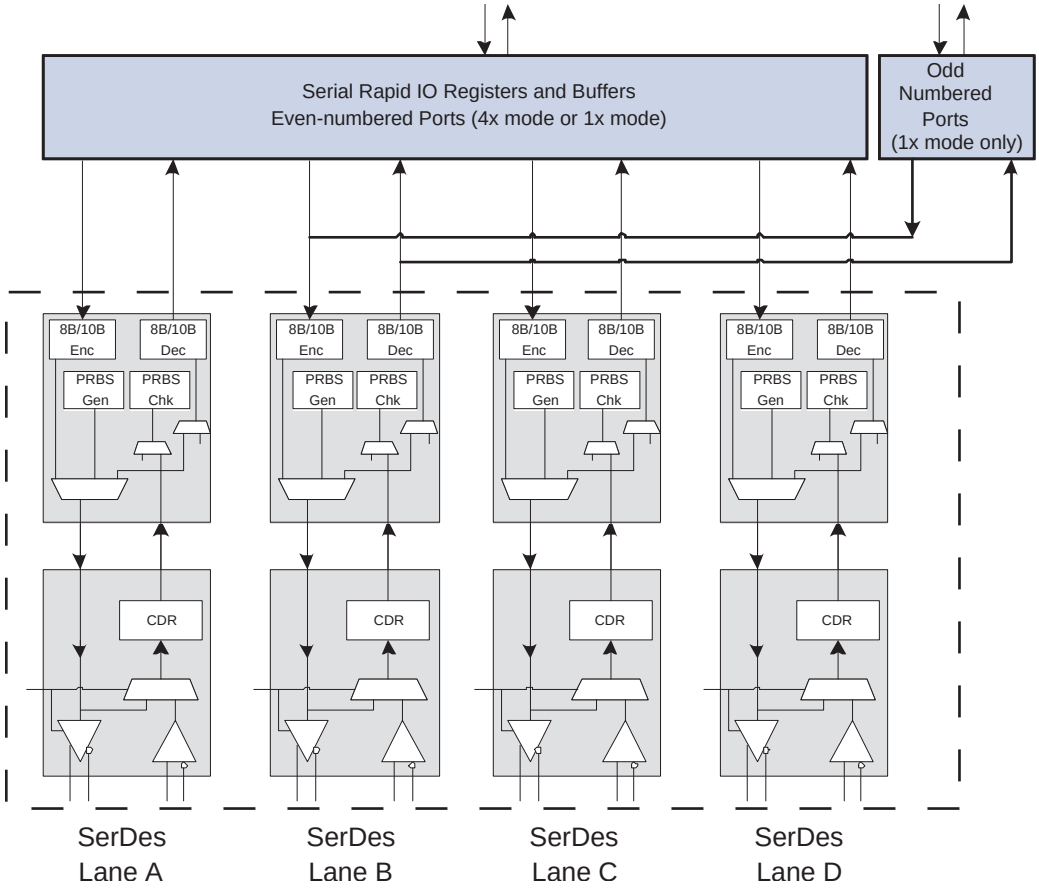
1.4 RapidIO Electrical Interfaces

The Tsi620 Media Access Controller (MAC) has three RapidIO ports. The three ports are grouped into pairs consisting of one even numbered port and one odd numbered port. Each pair of ports share four differential transmit lanes and four differential receive lanes.

Even and odd number ports have different capabilities. Even numbered ports can operate in either 4x or 1x mode, while odd numbered ports can only operate in 1x mode. When the even numbered port is operating in 4x mode, it has control over all four differential pairs. In 4x mode, the default state of the odd numbered port is powered on. All registers in the MAC are accessible but the MAC does not have access to the PHY. However, the odd numbered port can be powered down in this configuration. When the even numbered port is operating in 1x mode, the odd numbered port can also operate in 1x mode.

Each port has flexible testing features including multiple loopback modes, bit error rate test and signal scope support (for more information, see "RapidIO Electrical Interface"). The Tsi620 MAC block is shown in the following figure.

Figure 2: Tsi620 MAC Block Diagram



The RapidIO ports include the following features:

- Up to three ports in 4x LP-Serial mode
- Up to six ports in 1x LP-Serial mode (each 4x port can be configured as two 1x ports)
- RapidIO standard operating baud rate per data lane: 1.25 Gbps, 2.5 Gbps, or 3.125 Gbps
  - 12.5 Gbps inbound and 12.5 Gbps outbound bandwidth at 3.125 Gbps for a port configured for 4x mode
  - 3.125 Gbps inbound and 3.125 Gbps outbound bandwidth at 3.125 Gbps for a port configured for 1x mode
- Supports non-standard baud rates from 1 Gbps up to 3.2 Gbps
- Programmable serial transmit current with pre-emphasis equalization
- Serial loopback with a built-in testability

- Bit error rate testing (BERT)
- Hot-insertion capable I/Os and hardware support

1.5 FPGA Interface

The FPGA Interface supports a single, 4x/1x RapidIO port. It is a parallel interface that connects to low-cost FPGAs, and has flexible test features including multiple loopback modes (for more information, see “RapidIO Electrical Interface”).

The FPGA Interface includes the following features:

- Configurable as a single 4x/1x RapidIO port
- Operates at 1, 2, or 2.5 Gbps data rate per lane:
  - Maximum 10 Gbps for 4x mode
  - Maximum 2.5 Gbps for 1x mode
- Standards-based interface features:
  - Is a subset of the 10-Gb Ethernet XGMII Interface standard. XGMII specifies a standard connection to external 10-Gb Ethernet PHYs
  - Transmit Interface consists of the following:
    - Four data lanes, each 8 bits in size
    - Four control signals, one for each data lane, to distinguish RapidIO ‘K’ characters from data
    - One clock signal whose edges are center-aligned with the data and control signals
    - One TxDisable signal, indicating that the data being sent should be interpreted as errors/silence on the link
  - Receive Interface consists of the following:
    - Four data lanes, each 8 bits in size
    - Four control signals, one for each data lane, to distinguish Serial RapidIO ‘K’ characters from data
    - One clock signal whose edges are center-aligned with the data and control signals
    - One RxError signal, indicating that the data being received on the link should be interpreted as silence/error
  - Transmit clock signal supports frequency options of 156.25, 125, or 62.5 MHz
  - Receive clock signal must be frequency locked to the transmit clock signal. This is typically done by driving the link partners with the same clock source.
  - Uses Dual Data Rate (DDR) signaling - data is presented on both edges of the clock signal
  - Supports center-alignment relationship between clock and data/control signals
  - Supports HSTL1 (High Speed Transceiver Logic) 1.5V electrical standard.

- Serial loopback with a built-in testability
- Can be powered down to conserve power when not in use

1.6 Multicast Engine

The Switch ISF Multicast Engine is compliant with the *RapidIO Version 1.3 Part 11 Multicast Specification*.

A RapidIO multicast operation is a single transaction sent to more than one target RapidIO endpoint.

1.6.1 Multicast Operation

In a multicast operation, packets are received at the speed of any ingress port (up to 10 Gbaud) and broadcast at the speed of the egress ports (up to 10 Gbaud) to every port can accept. This results in a maximum aggregate multicast rate of up to 50 Gbaud out of the switch fabric. The maximum amount of data that can be transmitted is 40 Gbaud for four egress ports operating at maximum width and lane speed, based on the number of ports on the Switch.

Packets are routed to the multicast engine based on their destID and TT field value: however, if no match is detected for the destID and TT field then the lookup tables route the packet. A maximum of eight different destIDs/TT field combinations can be routed to the multicast engine. Each destID/TT set can be multicast to a different set of egress ports. A set of egress ports that packets are multicast to is called a multicast group and is represented by the multicast mask in the group table. A multicast packet is never sent out on the port that it was received on, so that a number of ports can share the same multicast group.

If no match is detected for the destID and TT field then the lookup tables route the packet.

Multicast packets are accepted by egress ports based on priority. If multicast and non-multicast traffic are competing for resources in the egress port, multicast specific egress arbitration can be used to favor multicast or non-multicast traffic.

1.6.2 Features

The supports multicast packet replication in accordance with *RapidIO Specification Version 1.3, Part 11 Multicast*.

The Switch ISF includes the following features:

- One Multicast Engine provides dedicated multicast resources without impacting throughput on the ports
- Eight multicast groups
- 40 Gbaud of sustained multicast output bandwidth, up to 10 Gbaud per egress port
- 10 Gbaud of instantaneous multicast input bandwidth[1]
- Packets are replicated to each egress port in parallel

- The Multicast Engine can accept bursts of traffic with different packet sizes
- Arbitration at the egress port to allow management of resource contention between multicast or non-multicast traffic.

System behavior when multicasting of packets which require responses is not defined in the *RapidIO Interconnect Specification (Revision 1.3) - Part 11 Multicast Specification*.

1.6.3 Terminology

Table 1 contains the terms that describe the multicast functionality.

Table 1: Terminology

Term	Definition
Multicast Group	A multicast group is a set of ports that must all receive a copy of a packet. A system can support multiple multicast groups, each of which is completely independent of the other (a group can have all, some, or no ports in common with another group). A multicast group is identified by the destID and TT fields of a packet. A packet is never multicast back out of the port that it is received on, regardless of whether or not this port is included in the multicast group.
Multicast Mask	The set of ports in a multicast group.
Multicast Vector	The set of ports in a multicast group that will receive the multicast packets.
Original Packet	A single multicast packet that arrives at a switch and gets replicated and sent to multiple egress ports according to the multicast mask.
Packet Copy	A copy of an original packet. The copies are sent out on the egress ports.
Multicast Traffic	Packets which are sent to the multicast engine from ingress ports, and packets which are received from the multicast engine by egress ports.
Unicast Traffic	All packets which are not sent to or received from the multicast engine.

1.7 Switch ISF

The Switch Internal Switching Fabric (Switch ISF) is the crossbar switching matrix at the core of the Switch. It transfers packets from ingress ports to egress ports and prioritizes traffic based on the RapidIO priority associated with a packet and port congestion.

The Switch ISF has the following features:

- Full-duplex, non-blocking, crossbar-based switch fabric
- 10 Gbps fabric ports allow up to 10x internal speedup
- Manages head-of-line blocking on each port
- Buffers hold eight packets per ingress RapidIO port

1. All bandwidths assume the Switch ISF is clocked at 156.25 MHz.

- Buffers hold eight packets per egress RapidIO port
- Cut-through and store-and-forward switching of variable-length packets

1.8 RapidIO-to-PCI Bridge (Tsi620 Bridge)

The RapidIO bridge component of the Tsi620 (also called the “Tsi620 Bridge”), includes the modules that perform the various RapidIO-to-PCI and PCI-to-RapidIO bridging functions of the device. The Tsi620 Bridge provides non-transparent bridging between up to four PCI devices and devices connected to the Tsi620’s RapidIO ports (see “PART 3: RAPIDIO-to-PCI BRIDGE (Tsi620 BRIDGE)”).

1.9 Serial RapidIO Endpoint (SREP)

The SREP comprises the following:

- A RapidIO physical layer interface that supports a single 4x/1x port
- A RapidIO transport layer that supports correct packet routing for packets received and transmitted by the SREP
- RapidIO Doorbell support
- RapidIO Logical I/O support for NREAD, NWRITE, NWRITE_R, and SWRITE packets.
- RapidIO Maintenance transaction support for reads, writes, and port-write packets.
- RapidIO support for Part 8: Error Management Extensions compliant to *RapidIO Interconnect Specification (Revision 1.3)*, including both port-write and interrupt based error notification functions.
- A Bridge ISF interface for passing requests and responses between the RapidIO functionality and the other functions within an application of the SREP
- An register bus master interface that supports register access from RapidIO transactions
- An register bus slave interface that supports access to SREP registers from other blocks in Tsi620.

The SREP has two kinds of information storage structures: Queues and Buffers. Buffers hold significant amounts of data, while Queues hold scheduling information about the transactions in the buffer. Queues also support arbitration for transactions entering and leaving the buffers.

1.9.1 RapidIO to Bridge ISF/Register Flow

RapidIO requests and responses are received from the RapidIO Interface and are routed to one of three destinations:

- Register requests (maintenance reads/writes, port-writes, and NREAD/NWRITE/NWRITE_R register accesses) are routed to the Register Request Data Buffers/Register Request Header Queue. These buffers/queues perform register accesses through the register bus Master Interface, and buffer the register responses in the Register Response Data Buffers/Header Queue. The register block also supports the reception and transmission of port-write transactions.

- Logical I/O requests and responses (NREAD, NWRITE, NWRITE_R, SWRITE requests and responses,) are routed to the R2I Data Buffers/R2I Header Queue. These buffers/queues manage requests and responses that must be bridged to the Bridge ISF. The R2I Data Buffers/R2I Header Queue's assemble multiple RapidIO response packets into a single Bridge ISF response. Information about the requests/responses is passed to the Bridge ISF R2I Request Queue. After the Bridge ISF has arbitrated for a packet to be transmitted, the request and any associated information is passed to the Bridge ISF.
- Doorbell requests are handled by the Doorbell Buffer.

The RapidIO to Bridge ISF/Register flow operates in store-and-forward mode. A RapidIO packet is completely received before it is forwarded to the Bridge ISF.

1.9.2 Bridge ISF to RapidIO Flow

Requests and responses are received from the Bridge ISF interface, and are routed to the I2R Data Buffers/I2R Header Queue. Requests are decomposed into valid RapidIO transactions and transmitted. Responses to Bridge ISF requests are received in the RapidIO to Bridge ISF flow, and assembled in the R2I Data Buffers/Header Queues. Responses received from the Bridge ISF are transmitted as RapidIO response packets.

Some Bridge ISF requests do not result in the transfer of RapidIO packets. The responses to these requests are place into the I2I Queue, where they are in turn accepted into the R2I Buffer and Queue for transfer to Bridge ISF.

Short response packets that do not require data, such as NWRITE_R responses, RapidIO responses for error conditions, and responses for RapidIO Doorbell packets, are tracked in the R2R Queue.

Round-robin arbitration selects packets from the four Bridge ISF-to-RapidIO flows (port-write, register response, R2R Queue, and I2R Request/Response). The packets selected are transmitted on the RapidIO Physical Layer interface.

The Bridge ISF to RapidIO flow operates in cut-through mode at all times. Transactions received from the Bridge ISF can begin transmission on RapidIO before the complete Bridge ISF transaction is received.

1.10 Bridge ISF

The Bridge ISF interconnects the Tsi620's PCI Interface and SREP by providing a low latency transaction flow. The Bridge ISF has the following features:

- Contains two 64-bit ports
- Operates at either 125 or 156.25 MHz
- Supports simultaneous block-to-block and non-blocking transactions
- Provides ECC generation and error correction/detection on transfers
- Provides a peak throughput of 10 Gbps from port to port
- Does not require configuration by the end user

1.11 PCI Interface

The PCI Interface transfers PCI data between the bus and the Bridge ISF, and vice versa. This design allows other interfaces that are connected to the Bridge ISF to initiate transactions on the PCI bus. This also allows PCI devices to initiate transactions that access Tsi620 internal registers as well as other devices connected to the Bridge ISF.

The PCI Interface has the following features:

- Compliant with the following specifications:
  - *PCI Local Bus Specification (Revision 2.3)*
  - *PCI Bus Power Management Interface Specification (Revision 1.1)*
- 32- or 64-bit addressing
- 32-bit data bus
- PCI operation from 25 to 66 MHz
- Provides bus arbitration for four external PCI devices
- Supports vital product data (VPD)
- Supports message signaled interrupts (MSIs)
  - Generates outgoing MSIs
  - Handles incoming MSIs as posted write operations
- PCI master capability
  - Generates all *PCI Local Bus Specification (Revision 2.3)* commands except IACK and Special cycle
  - Multi-threading for PCI delayed read cycles
- PCI target capability
  - Accepts *PCI Local Bus Specification (Revision 2.3)* commands except IACK, Special cycle, I/O read, and I/O write
  - No target wait states
  - Supports up to four concurrent reads
  - Supports posted writes, delayed reads, and config type 0 writes
- Supports direct address translation (that is, the PCI address is passed unmodified directly to the Bridge ISF) and indexed address translation with the Bridge ISF
- 3.3V compliant I/Os; 5V is not supported
- Register support
  - Accessed through multi-master internal register bus
  - Full runtime access of PCI configuration space through PCI or internal register bus

The PCI Interface does not support the following:

- PCI lock command. In a switch fabric architecture, this type of command can cause transaction congestion and deadlock scenarios.
- 64-bit data bus
- Compact PCI Hot Swap
- Less than 25-MHz minimum speed of operation, as defined in the *PCI Local Bus Specification (Revision 2.3)*

1.12 I2C Interface

The I2C Interface provides a master and slave serial interface that can be used for the following purposes:

- Initializing device registers from an EEPROM after reset
- Reading and writing external devices on the I2C bus
- Reading and writing Tsi620's internal registers for management purposes by an external I2C master

The I2C Interface has the following features:

- Operates as a master or slave on the I2C bus
  - Multi-master support
    - Arbitrates among multiple masters for ownership of the I2C bus
    - Automatically retries accesses if arbitration is lost
    - Provides timeout indication if the Tsi620 is unable to arbitrate for the I2C bus
  - I2C Interface: Master interface
    - Supports 7-bit device addressing
    - Supports 0, 1, or 2-byte peripheral addressing
    - Supports 0, 1, 2, 3, or 4-byte data transfers
    - Reverts to slave mode if arbitration is lost
    - Supports clock stretching by an external slave to limit bus speed to less than 100 kHz
    - Handles timeouts and reports them through interrupts

— I^2C Interface: Slave interface

– Slave address can be loaded from three sources: power-up signals,boot load from EEPROM, or by software configuration

– Provides read and write accesses that are 32 bits in size to all Tsi620 registers

– Ignores General-Call accesses

– Ignores Start-Byte protocol

– Provides a status register for determination of Tsi620's health

– Slave operation enabled/disabled through power-up signal,boot load from EEPROM, or by software configuration

– Provides mailbox registers for communicating between maintenance software operating on RapidIO based processors and external I^2C masters

- Supports I^2C operations up to 100 kHz
- Provides boot-time register initialization
  — Supports 1- and 2-byte addressing of the EEPROM selected by power-up signal
  — Verifies the number of registers to be loaded is legal before loading registers
  — Supports up to 2K byte address space and up to 255 address/data pairs for register configuration in 1-byte addressing mode, or up to 65Kbyte address space and up to 8K-1 address/data pairs in 2-byte addressing mode.
  — Supports chaining to a different EEPROM and/or EEPROM address during initialization.

The I^2C Interface does not support the following features:

- START Byte protocol
  — Tsi620 does not provide a START Byte in transactions it masters
  — Tsi620 does not respond to START Bytes in transactions initiated by other devices. The Tsi620 will respond to the repeated start following the start byte provided the 7-bit address provided matches the Tsi620 device address.
- CBUS compatibility
  — Tsi620 does not provide the DLEN signal
  — Tsi620 does not respond as a CBUS device when addressed with the CBUS address. The Tsi620 will interpret the CBUS address like any other 7-bit address and compare it to its device address without consideration for any other meaning.
- Fast Mode or High-Speed Mode (HS-MODE)
- Reserved 7-bit addresses should not be used as the Tsi620's 7-bit address. If a reserved address is programmed, the Tsi620 will respond to that address as though it were any other 7-bit address with no consideration of any other meaning.

- 10-bit addressing
  - Tsi620 must not have its device address programmed to the 10-bit address selection (11110XXb) in systems that use 10-bit addressing. The Tsi620 will interpret this address like any other 7-bit address and compare it to its device address without consideration for any other meaning.
- General Call. The general call address will be NACK'd and the remainder of the transaction ignored up to a subsequent Restart or Stop.

1.13 GPIO Interface

The Tsi620 supports 32 General Purpose Input/Output (GPIO) signals. These signals are all 3.3 V. They can be used as outputs, or as inputs. When used as inputs, the signals can be used as events to trigger event notification for PCI (interrupt assertion), the SREP, or the Tsi620 interrupt signal.

1.14 JTAG Interface

The JTAG Interface in Tsi620 is fully compliant with IEEE 1149.6 B*oundary Scan Testing of Advanced Digital Networks* as well as IEEE 1149.1 *Standard Test Access Port and Boundary Scan Architecture* standards. There are five standard pins associated with the interface (TMS, TCK, TDI, TDO and TRST_b) which allow full control of the internal TAP (Test Access Port) Controller.

The JTAG Interface has the following features:

- Contains a 5-pin Test Access Port (TAP) Controller, with support for the following registers:
  - Instruction register (IR)
  - Boundary scan register
  - Bypass register
  - Device ID register
  - User test data register (DR)
- Supports debug access of Tsi620's registers
- Supports the following instruction opcodes:
  - Sample/Preload
  - Extest
  - EXTEST_PULSE *(1149.6)*
  - EXTEST_TRAIN *(1149.6)*
  - Bypass
  - IDCODE
  - Clamp
  - User data select

1.15 Transaction Flow

RapidIO packets received by the RapidIO ports are routed through the Switch ISF to the appropriate destination: another RapidIO port, the Multicast Engine, or the SREP. Packets routed to a RapidIO port are then transmitted by the switch port. Packets sent to the Multicast Engine are replicated in parallel and sent to the appropriate RapidIO ports and/or the SREP. Packets sent to the SREP are bridged to the appropriate Bridge ISF transaction and routed to the PCI or SREP. If a SREP transaction is looped back to the SREP, this has the effect of using the Tsi620 as a non-transparent RapidIO bridge. The PCI and SREP interfaces translate the Bridge ISF transaction to the appropriate interface-specific transaction, and route a response back to the SREP. The SREP translates the Bridge ISF response transaction to the appropriate RapidIO response packet. The RapidIO response packet is then routed out the RapidIO port that the request was received on.

Maintenance packets with a hop count of 0 received by the RapidIO ports are processed by the RapidIO port against the Tsi620 Switch registers, and an appropriate response is then sent. Maintenance packets with a non-zero hop count received by the RapidIO ports have their hop count decremented, and are then routed out a RapidIO port or to the SREP.

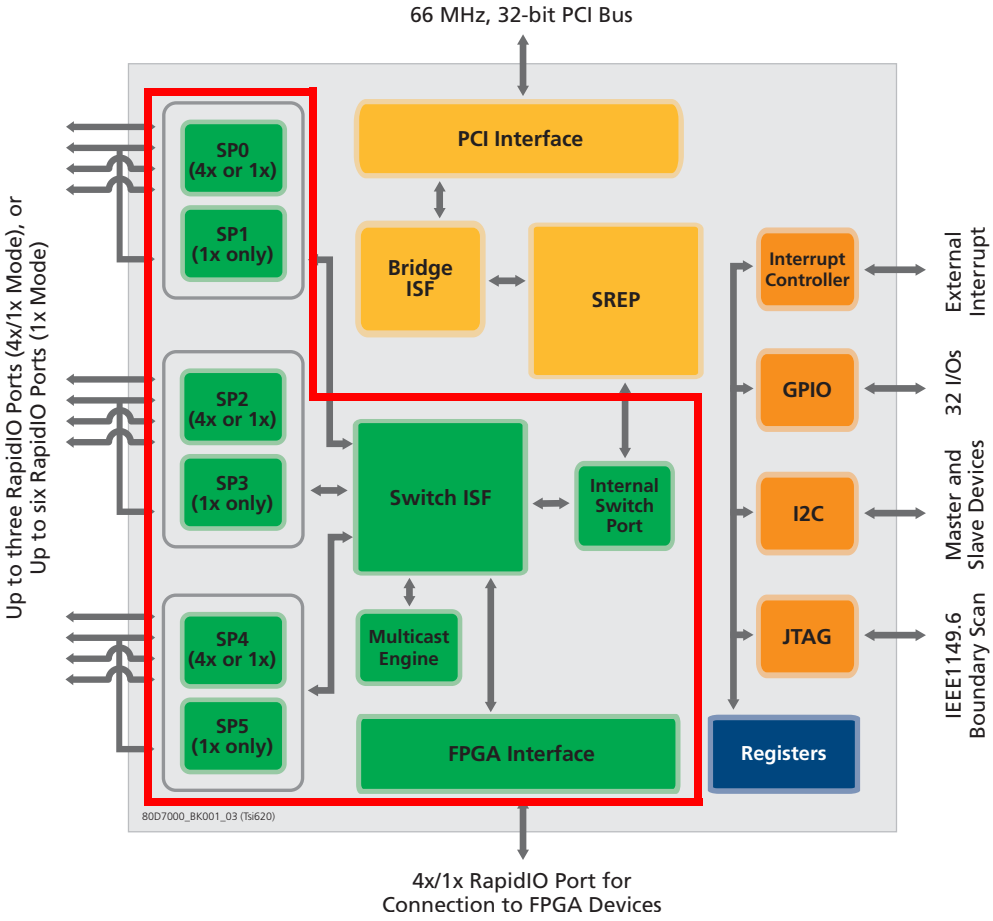
The SREP processes all maintenance packets received by it against the Tsi620 registers, and sends an appropriate response. The SREP can receive RapidIO Doorbell and Port-Write packets, which are queued in the SREP for processing by software. Maintenance, Doorbell, and Port-Write packets are not bridged to the Bridge ISF.

The PCI Interface can receive configuration requests, which are processed against the PCI Interface's configuration space registers. The PCI Interface can also receive memory requests, which may be processed against the Tsi620 registers, or may be bridged to the Bridge ISF. Memory requests bridged to the Bridge ISF are routed to the SREP. The SREP translates it to the appropriate RapidIO packet and sends it to the RapidIO port or the Multicast Engine according to values programmed in the SREP. RapidIO responses for the request are received by a RapidIO port and routed back to the SREP, where they are translated to Bridge ISF response transactions and sent to the PCI Interface.

PART 2: RAPIDIO SWITCH (Tsi620 SWITCH)

The RapidIO switch component of the Tsi620 (also called the “Tsi620 Switch”), includes the modules that perform various RapidIO-specific tasks, such as transferring and receiving RapidIO packets, address translation, packet routing, and multicasting. The Tsi620 Switch provides switching functionality for up to seven RapidIO devices; six using the 4x/1x ports, and another through the FPGA Interface.

The Tsi620 Switch topics that are discussed in this part of the document are highlighted in the following figure.



2. RapidIO Interface

Topics discussed include the following:

2.1 Overview

The Tsi620 contains a high-performance RapidIO Interface that provides connectivity for control plane and data plane applications. The device’s RapidIO ports are compliant with the *RapidIO Interconnect Specification (Revision 1.3)*.

This document typically uses RapidIO Interface to refer to the Tsi620’s 4x/1x RapidIO ports. This term, however, also applies to the device’s FPGA Interface since this interface can be configured as a RapidIO port.

This section describes the transport layer features common to all Tsi620 RapidIO ports. For information on the electrical layer characteristics of the RapidIO ports, see “RapidIO Electrical Interface”.

The RapidIO ports have the following capabilities:

- RapidIO packet and control symbol transmission
- RapidIO packet and control symbol reception
- Register access through RapidIO maintenance requests

2.1.1 Features

The following features are supported:

- Up to three 4x-mode or up to six 1x-mode RapidIO ports operating at up to 3.125 Gbps
- Per-port destination ID lookup table, used to direct packets through the switch

This is an IDT-specific design. The *RapidIO Interconnect Specification (Revision 1.3)* standard design of lookup tables is also supported.

- RapidIO error management extensions described in *RapidIO Interconnect Specification (Revision 1.3) Part 8*, including both hardware and software error recovery
- Low latency forwarding of the multicast control symbol
- Proprietary registers for performance monitoring and tuning
- Both cut-through and store-and-forward modes for performance tuning
- Debug packet generation and capture
- Multicast capability
- Head-of-line blocking avoidance

2.1.2 Transaction Flow Overview

Packets and control symbols are received by the Serial RapidIO Electrical Interface (Serial MAC) and forwarded to the RapidIO Interface (for more information on the Serial MAC, see "RapidIO Electrical Interface"). Received packets have their integrity verified by error checking defined by *RapidIO Interconnect Specification (Revision 1.3)*. Once the packet's integrity is verified, the destination ID of the packet is used to access the routing lookup table to determine which port the packet should be forwarded to and whether the packet is a multicast packet. The packet is then buffered by the Switch ISF for transmission to the port. After the packet is transferred to the egress port, the port transmits the packet. If a packet fails the CRC check, the packet is discarded and the transmitter is instructed to retransmit the packet through the use of control symbols.

The egress port receives packets to be transmitted by the Switch ISF. The integrity of packets forwarded through the Switch ISF is retained by sending the CRC code received with the packet. For more information on the input and output queues, see "Packet Queuing".

The packet transmitter and the packet receiver cooperate to ensure that packets are never dropped (lost). A transmitter must retain a packet in its buffers until the port receives a packet accepted control symbol from the other end of the link.

2.1.3 Maintenance Requests

A maintenance packet is the only packet type that is modified by the RapidIO Switch. If the hop count value of the maintenance request is 0, the maintenance request is forwarded to the register bus for processing. The register bus accesses the registers in the appropriate port. The response to the maintenance request is compiled into a maintenance response packet and queued by the port for transmission. Maintenance packets with a non-zero hop count value have their hop count decremented, CRC recomputed, and are then forwarded to the port selected by the destination ID value in the lookup table.

2.1.4 Control Symbols

Control symbols received by the Tsi620 have their CRC validated, and their field values checked. If either the CRC is incorrect or the control symbol field values are incorrect, a packet-not-accepted control symbol is sent back and the control symbol is discarded. Otherwise, the control symbol is used by the port for packet management in the transmit port or link maintenance.

2.2 Transaction Flow

The Tsi620 receives a RapidIO packet on one of its RapidIO ports. After performing integrity checks, such as validating a CRC, the interface logic locates the destination ID in the packet. The Tsi620 uses this information to determine to which egress port the packet must be sent and whether it is a multicast packet. It consults a user-configurable lookup table, which maps destination ID into egress port numbers.

The RapidIO port transfers the packet to the Switch ISF where it is buffered and transferred to an egress port or to the Multicast Engine. The Switch ISF is non-blocking, which means that all ports can switch data at the same time as long as they are not switching data from multiple ports to a single port. The Switch ISF manages head-of-line blocking, which means that when a packet cannot be moved to an egress port (for example, because multiple ingress ports are trying to send to the same egress port), the Switch ISF selects another packet to service from the same ingress port.

The ingress queue of the Tsi620 can operate in two modes: store-and-forward and cut-through (see TRANS_MODE in “RapidIO Port x Control Independent Register”). In store-and-forward mode, the ingress port of the device waits for the arrival of the whole packet before sending it to the Switch ISF. In cut-through mode, the ingress port transmits the packet as soon as the Switch ISF grants access (when the routing information is received). However, in both modes the egress port always operates in cut-through mode: the packet is immediately forwarded. A copy of the packet is saved at the egress port so that it can be retransmitted should an error occur.

RapidIO provides a *stomp* function to abort partially transmitted packets that are later determined to have data integrity errors or similar errors. This means if the Tsi620 finds that a packet that is being cut-through has an error, it may send a stomp control symbol to notify the receiver that the packet was in error and all received data of the erred packet should be dropped.

Packets delivered to the Multicast Engine (MCE) are replicated, based on user-configured “multicast groups”. The MCE sends copies of the original packet to the egress ports in a parallel fashion

Packets can cut-through from the ingress port to the Multicast Work Queue and from the Multicast Work Queue to the Broadcast Buffers. A complete packet copy must be received by a Broadcast Buffer before it attempts to forward the packet copy to the egress port.

2.3 Lookup Tables

Lookup tables (LUTs) direct incoming packets to output ports. An ingress port performs this routing operation by mapping the destination ID field of an incoming packet to an egress port number on the Tsi620 Switch. The ingress port does this by using the destination ID as an index to a lookup table containing user-defined egress port numbers.

Each RapidIO port has its own, uniquely configurable lookup table[1]. Configuration and maintenance of the LUTs is compliant with the *RapidIO Interconnect Specification (Revision 1.3)*. All LUTs are written simultaneously by these registers. Additionally, the LUT of each port can be accessed using device-specific registers.

The Internal Switch Port has its own, uniquely configurable lookup table. This module’s lookup table is identical to that of the other RapidIO ports.

The LUTs support two modes of operation, selectable on a per-port basis: “Flat Mode” and “Hierarchical Mode”. Flat mode is the default mode and it supports destination IDs in the range of 0 to 511, with a default port for destination IDs outside this range. The hierarchical model covers the large system range of 64-KB destination IDs, with some limitations.

To ensure high system reliability, the lookup tables are parity protected. System software must intervene when a parity error is detected. The Tsi620 guarantees that packets are not incorrectly delivered when the lookup table incurs single bit errors.

When a packet arrives at the ingress port, the destination ID of the packet is examined against the Multicast Group Table to determine if the packet is a *multicast* packet (see “Multicast Engine”).

2.3.1 Filling the Lookup Tables

The process of filling in the LUT is composed of the following series of register writes:

- The “RapidIO Route Configuration DestID CSR” is loaded with the destination ID value to be routed
- The “RapidIO Route Configuration Output Port CSR” is written with the desired egress port number

1. Routing packets to RapidIO port 7 will result in undefined behavior.

If there is an attempt to write a destination ID with a value of greater than 511 into the "RapidIO Route Configuration DestID CSR" using the LRG_CFG_DESTID and CFG_DESTID fields, the upper seven bits of the destination ID in the LRG_CFG_DESTID field is truncated.

"RapidIO Route LUT Size CAR" only advertises the Tsi620 Switch can map 512 destination IDs. This is due to the fact that this register is global in scope, whereas the ports can be independently configured for either flat mode or hierarchical mode lookup tables.

The LUT of all the ports can be loaded simultaneously if the same routing entries in all of the ports is required. The loading process is similar to loading an individual port's LUT, however alternative registers are used. The register addresses are:

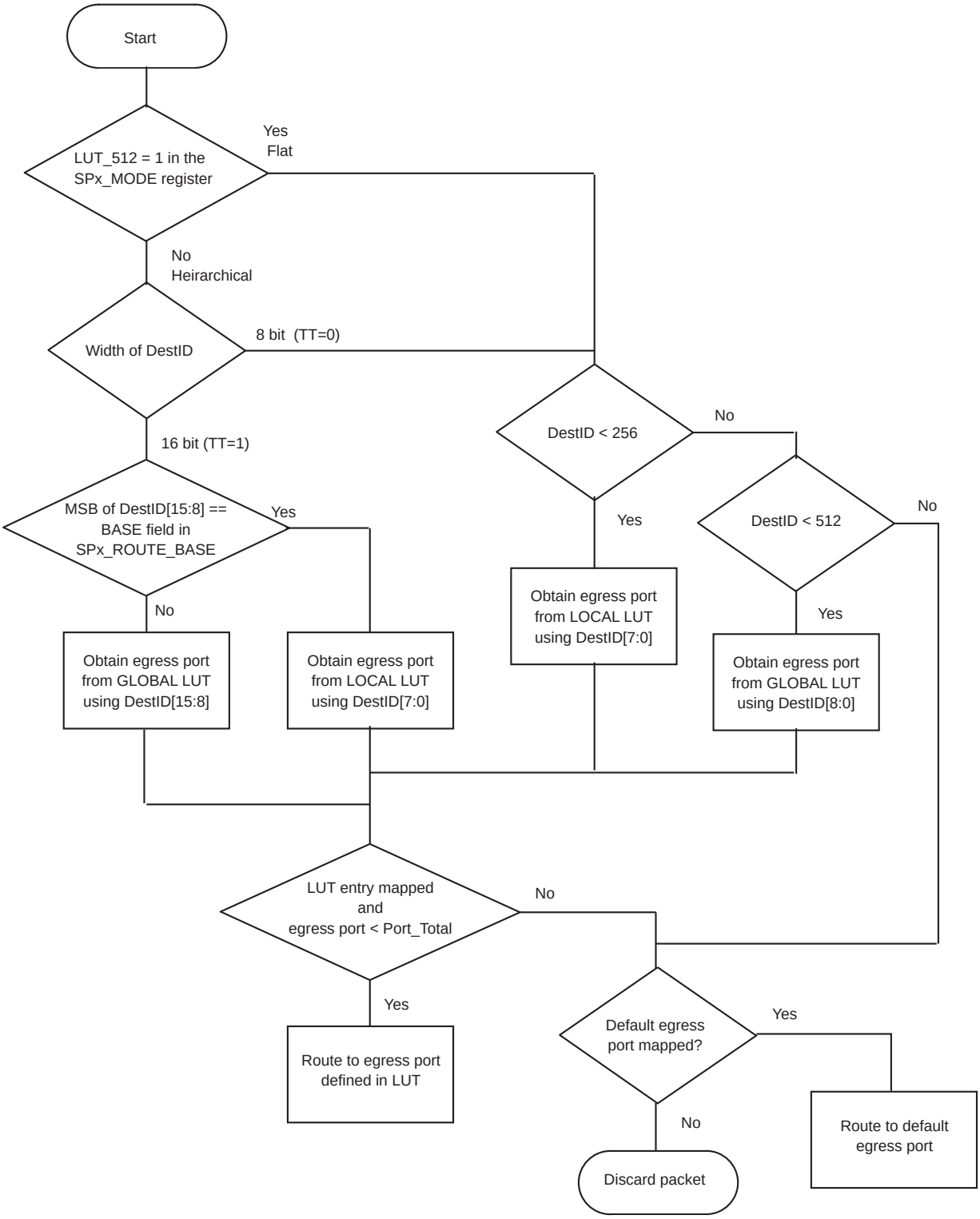
- "RapidIO Route Configuration DestID CSR" or
- "RapidIO Port x Route Configuration DestID CSR"

and

- "RapidIO Route Configuration Output Port CSR" or
- "RapidIO Port x Route Configuration Output Port CSR"

The register sets are identical except that "RapidIO Route Configuration Output Port CSR" are per-port configuration registers and include an auto-increment bit to increment the contents of "RapidIO Route Configuration DestID CSR" after a read or write operation[1].

Figure 3: LUT Mode of Operation



2.3.2 LUT Modes

The LUT mode, Flat or Hierarchical, is selected on a per-port basis through the LUT_512 field in the "RapidIO Port x Mode CSR".

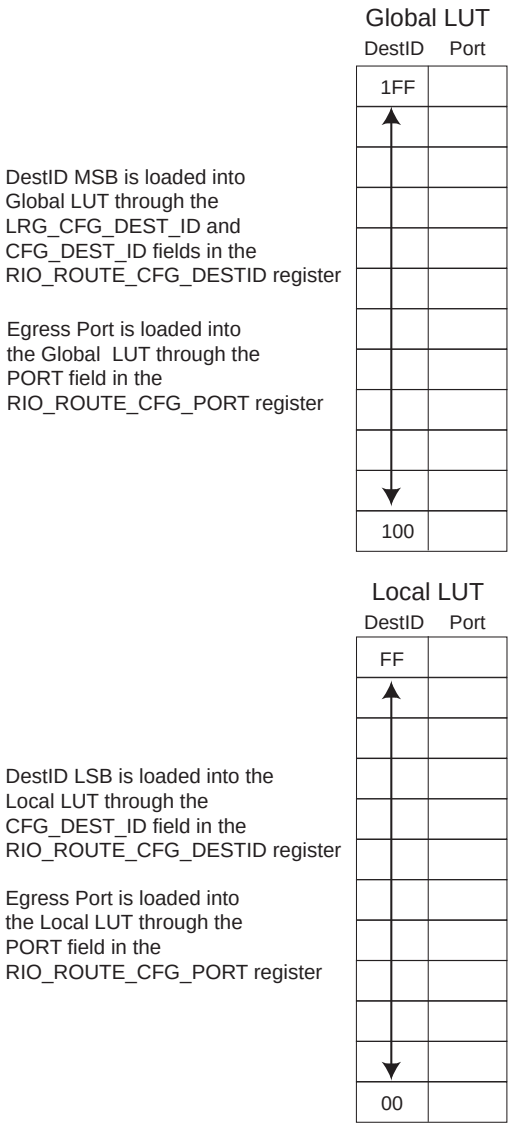
2.3.2.1 Flat Mode

A Flat mode LUT is a table that maps destination IDs 0 to 511 to user selectable egress ports. Destination IDs that fall outside this range are sent to the egress port identified in the "RapidIO Route LUT Attributes (Default Port) CSR".

Flat mode is the default mode of operation of the LUT.

Figure 4 shows the configuration of the Local and Global Lookup tables (LUT) in Flat mode[1].

Figure 4: Flat Mode Routing



An incoming packet's destination ID is examined following the process in the flowchart in Figure 3. The egress port number is obtained from the LUT if there is a match between the destination ID in the packet header and the table. If there is no match, the packet is routed based on the default egress port programmed into "RapidIO Route LUT Attributes (Default Port) CSR". If the default port is unmapped, the packet is discarded and the Tsi620 sets the IMP_SPEC_ERR bit in the "RapidIO Port x Error Detect CSR".

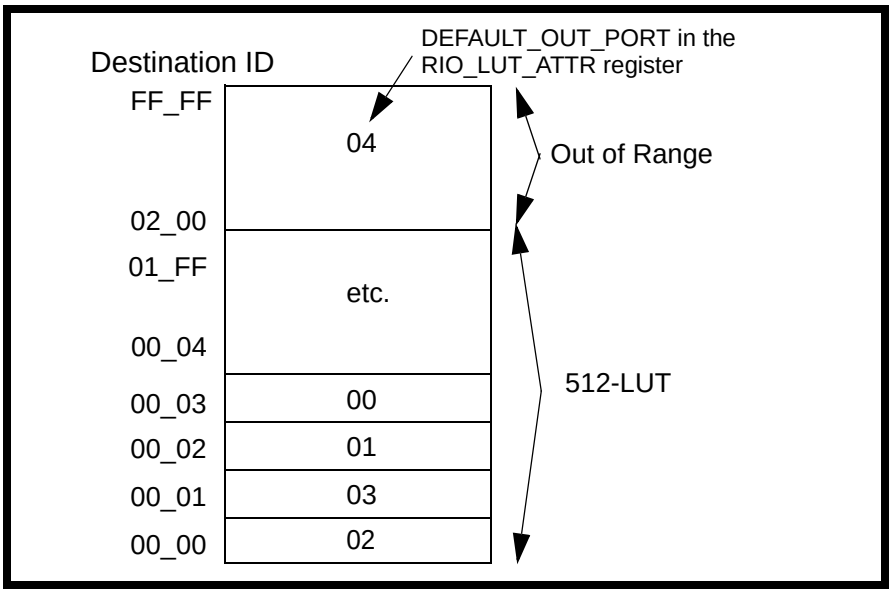
2.3.2.2 Flat LUT Programming

Each of the ports on the Tsi572 has its own lookup table. Each lookup table can be programmed with different values which allows each port to route packets differently. The lookup table maps the packet to the correct output port based on the destination ID. The capability of each port having their own LUT is functionality that is not required in the RapidIO Interconnect Specification (Revision 1.3).

LUT entries can be reprogrammed at any time during normal system operation. However, software must ensure transactions have completed before reprogramming the LUTs.

Figure 5 shows an example of a LUT in flat mode. In this example, a destination ID of 0x0002, or 0x02, is routed by the RapidIO Switch to output port 1. A destination ID of 0x0003, or 0x03, is routed out port 0 and destination IDs greater than 0x1FF are routed out port 4.

Figure 5: Flat Mode LUT Configuration Example



Registers Used in Lookup Table Configuration

The Tsi620's RapidIO Interface is compliant with the *RapidIO Interconnect Specification (Revision 1.3)*. The following RapidIO standard registers are used by the Tsi620 for programming the lookup tables:

- "RapidIO Route Configuration DestID CSR"
- "RapidIO Route Configuration Output Port CSR"
- "RapidIO Route LUT Size CAR"
- "RapidIO Route LUT Attributes (Default Port) CSR"
- "RapidIO Port x Local Routing LUT Base CSR"

Other related registers are IDT specific and include the following:

- "RapidIO Port x Route Configuration DestID CSR"
- "RapidIO Port x Route Configuration Output Port CSR"
- "RapidIO Port x Local Routing LUT Base CSR"

Other indirectly related (Multicast) registers include:

- "RapidIO Multicast Mask Configuration Register"
- "RapidIO Multicast DestID Configuration Register"

All lookup table entries are in an unknown state after power-up. All entries should be programmed to a mapped or unmapped state to ensure predictable operation. It is strongly recommended that 0xFF be used as the port value for writing unmapped lookup table entries. An unmapped lookup table entry returns the value of 0xFF as the port value when read.

Lookup Table Configuration Examples

The Tsi620 lookup tables can be configured through an external EEPROM or through software maintenance writes to the Tsi620 registers.

It is strongly recommended that the entire lookup table be configured on each port to avoid undefined lookup table entries that may cause non-deterministic behavior.

The following sequence programs the lookup tables using the broadcast (BC) offset:

1. Write the destination ID to be configured or queried using the broadcast (BC) offset in the "RapidIO Route Configuration DestID CSR".
2. Read the "RapidIO Route Configuration Output Port CSR" to determine the current egress port for the destination ID, or write this register to change the configuration of the destination ID.

Example One: Adding a Lookup Table Entry

In the following example, routing is added for all ports to route destination ID 0x98 to output port 0x4. To add a lookup table, complete the following steps:

1. Write to the "RapidIO Route Configuration DestID CSR" with a value of 0x00000098. This makes the Destination ID 0x98.
2. Write to the "RapidIO Route Configuration Output Port CSR" a value of 0x00000004. This makes the Port 0x4.

Example Two: Adding a Lookup Table Entry

In the following example, routing is added for port 0x5 to route destination ID 0x20 to output port 0x3. To add a lookup table, complete the following steps:

1. Write to the "RapidIO Route Configuration DestID CSR" with a value of 0x80000020. This makes the Destination ID 0x20 and the Auto-increment 0x1.
2. Write to the "RapidIO Route Configuration Output Port CSR" a value of 0x00000003. This programs the Port 0x3.

In this example, if a further write to the "RapidIO Route Configuration Output Port CSR" was performed the output port for Destination ID 0x21 is configured.

Example Three: Verifying/Reading a Lookup Table Entry

In the following example, output port for destination ID 0x54 is read. To verify and read a lookup table entry, perform the following steps:

1. Write to the "RapidIO Route Configuration DestID CSR" with a value of 0x00000054. This programs the Destination ID 0x54.

Read to the value in the "RapidIO Route Configuration Output Port CSR". This value represents the output port for packets with destination ID 0x54.

The value reported back is assumed to be for all ports but it only reports back the value in port 0.

2.3.2.3 Hierarchical Mode

The hierarchical mode of operation of the LUT allows the full range of 65536 16-bit destination IDs to be mapped. This mode is enabled by setting LUT_512 to 0 in "RapidIO Port x Mode CSR". The hierarchical mode of operation uses two LUTs, each containing 256 entries:

- For packets with 8-bit destination IDs, the ingress port uses the ID as an index into the "local" LUT (see "Flat Mode").
- For packets with 16-bit destination IDs:
  - If the most significant 8 bits of the packet's destination ID match the value configured in the BASE field of "RapidIO Port x Local Routing LUT Base CSR", the ingress port uses the least significant 8 bits of the packet's destination ID to index the "local LUT" and retrieve an egress port number.
  - If the most significant 8 bits of the packet's destination ID do not match the value configured in the BASE field of "RapidIO Port x Local Routing LUT Base CSR", the ingress port uses the most significant 8 bits of the packet's destination ID to index the "global LUT" and retrieve an egress port number. Thus, the majority of the 16-bit destination ID number space is covered by the remote LUT, with groups of 256 destination IDs targeting the same egress RapidIO port.

If the result of a lookup produces an egress port number greater than the value of PORT_TOTAL of "RapidIO Switch Port Information CAR", the incoming packet is routed to DEFAULT_PORT defined by "RapidIO Route LUT Attributes (Default Port) CSR". If the default port is unmapped, the packet is discarded and the Tsi620 raises the IMP_SPEC_ERR bit in the "RapidIO Port x Error Detect CSR"[1].

"RapidIO Route LUT Size CAR" only advertises the RapidIO Switch can map 512 destination IDs. This is due to the fact that "RapidIO Route LUT Size CAR" is a register with global scope, but the ports can be independently configured for either flat mode or hierarchical mode lookup.

2.3.2.4 Mixed Mode of Operation

A system can be operated in a mixed mode of operation, with some ports in flat mode and some ports in hierarchical mode. Each port performs destination ID lookup consistent with its configured mode of operation.

2.3.3 Lookup Table Parity

Each entry in the lookup table is parity protected. A LUT parity error is detected in an entry when an incoming packet causes the ingress port to read that table entry. If the ingress port detects an error, it discards the packet and reports the error (see Table 2). Because the packet is discarded on the ingress port, the packet is never forwarded to the egress port and a stomp control symbol is not required when the packet is discarded.

All LUT entries must be initialized before use to ensure that the parity bits are set appropriately.

2.3.4 Lookup Table Error Summary

Table 2 summarizes error conditions and resulting behaviors associated with the LUTs.

Table 2: Error Summary

Event	Behavior
Packet routed to a shut down port[a]	Packet discarded and no record of packet is kept
Packet routed to disabled port[b]	Switch ISF timeout occurs and a transaction error acknowledge (TEA) interrupt is asserted (if enabled) The TEA signal is asserted when a timeout is detected on the Switch ISF due to the blocking of the requested destination. TEA is only asserted after the output port buffers are full. When this signal is asserted, it indicates to the source of the transaction that the requested transaction could not be completed and is removed from the request queue. The TEA error is reported through a port-write and/or an interrupt. Programmable in the "Switch ISF Control Register" and the interrupt status can be checked in the "Switch ISF Interrupt Status Register".
Packet routed to unconnected port[c]	Packet discarded and no record of packet is kept
Packet routed using an unmapped LUT entry, and the default egress port is also unmapped	Packet header recorded in error capture registers and the packet is discarded • IMP_SPEC_ERR bit is set in the "RapidIO Port x Error Detect CSR" • Port write can be generated (if enabled) • Interrupt can be generated (if enabled).
Parity error on LUT entry	Packet header recorded in error capture registers, packet discarded • IMP_SPEC_ERR bit is set in the "RapidIO Port x Error Detect CSR" • Port write can be generated • Interrupt can be generated • Port number is also captured in "RapidIO Port x LUT Parity Error Info CSR"

Table 2: Error Summary (Continued)

Event	Behavior
Writes to LUT entries through the broadcast LUT registers to shutdown ports	Silently ignored
Access to LUT with a destination ID which exceeds LUT size (greater than 511 in flat mode)	Writes silently ignored, reads return 0xFF

a. When a port is shut down, all clocks are off to that port. Reading to the registers return 0 (except for 0x158 and 0x15C which give the correct values).

b. The port is healthy. Packets routed to that port are disallowed to pass through to the Link Partner. All registers are still functional and when read, and return the current operational values.

c. This is the same as a powered-on port except that the Link Partner behaved as disconnected to the port. The port is healthy and when the Link Partner is resurrected, the link between the port and the Link Partner is re-established.

If a LUT entry is unmapped for a specific port or the DestID does not match any of the LUT entries, packets are routed to the default output port, as defined by "RapidIO Port x Route Configuration DestID CSR". The IMP_SPEC_ERR bit is also set in the "RapidIO Port x Error Detect CSR". If the default output port is itself unmapped, however, then the packet is discarded.

2.3.5 Lookup Table Entry States

A lookup table entry can be in one of three states: mapped, unmapped, or parity error (see Table 3). The lookup table entries for a RapidIO port after the Tsi620 is reset, or a port is reset or powered up, will be one of the three states but it is not known which entry will be in what state. For this reason, all RapidIO ports must have their lookup table entries programmed to ensure deterministic operation.

Lookup table entries can be programmed through the standard RapidIO compliant interface or through an IDT-specific interface Tsi620 specific set of registers that can auto-increment entries reducing the number of maintenance cycles required. The Tsi620 specific set of registers include the "RapidIO Port x Route Configuration Output Port CSR" and "RapidIO Port x Route Configuration DestID CSR".

If a lookup table entry is in the unmapped state, packets are routed according to the contents of the "RapidIO Route LUT Attributes (Default Port) CSR". The default port can be mapped or unmapped. If the default port is mapped, packets are forwarded out of the configured port. If the default port is unmapped, an implementation-specific error is detected and the following actions occur:

- Packet header recorded in the error capture registers (see "RapidIO Port x Packet and Control Symbol Error Capture CSR 0 and Debug 1 Register")
- Packet is discarded
- IMP_SPEC_ERR bit is set in the "RapidIO Port x Error Detect CSR"

- Port write can be generated, if enabled
- Interrupt can be generated, if enabled

Table 3: Lookup Table States

Lookup Table Entry State	How to get into States	Action on Packet Arrival
Mapped	A lookup table entry that routes packets to a port that exists within the Tsi620 is mapped.	Packet is routed to the specified output port
Unmapped port value	A lookup table entry that routes packets to a port that does not exist with the Tsi620 is unmapped.	Default port is used for routing the packet The default port is defined in the "RapidIO Route LUT Attributes (Default Port) CSR"
Parity Error	When a lookup table entry's parity is incorrect, the lookup table entry is in a parity error state.	• Packet header recorded in error capture registers • Packet is discarded • IMP_SPEC_ERR bit is set in the "RapidIO Port x Error Detect CSR" • Port write can be generated • Interrupt can be generated • Port number is captured in the "RapidIO Port x LUT Parity Error Info CSR"

2.4 Maintenance Packets

Maintenance packets are handled differently than other packets by the Tsi620. In a system, the Tsi620 can be the destination of the maintenance packet.

Maintenance packet processing is based on the maintenance packet's hop count value. The hop count value controls how many hops the maintenance packet travels before it reaches its destination. The routing of the maintenance packet is controlled by the destination ID of the packet, the lookup table, and other values programmed in the intervening devices.

Ensure the destination ID of the maintenance packet does not match the destination ID of a multicast packet. If there is a match, system behavior is undefined.

If a maintenance packet has a hop count greater than 0, the Tsi620 decrements the hop count, recalculates the CRC, and routes the packet out the port selected by the LUT. For this reason, all maintenance packets must contain routable source and destination addresses and the routing LUT must be programmed to route both the maintenance transaction and its response.

If a maintenance read or maintenance write request packet has a hop count of 0, the port processes the maintenance request and sends a maintenance response packet. The maintenance request is passed to the register bus as a read or write transaction, an address offset, and any data associated with the request. The maintenance response packet is generated by the Tsi620 using the success or failure of the access and data from a read operation. CRC is computed and the packet is enqueued for transmission on the port that received the maintenance request.

Each port can have only one outstanding maintenance request at time. A maintenance request received while another maintenance is being processed is retried by the RapidIO port.

The Tsi620 supports 4 byte maintenance requests only. When the hop count equals 0, any Maintenance Requests larger than 4 bytes, as well as maintenance packets that are not read or write requests, are dropped and an error is noted in the IMP_SPEC_ERR bit of the "RapidIO Port x Error Detect CSR". Examples of maintenance packets that are dropped are maintenance response and port-write packets received with a hop count of 0.

Table 4: Maintenance Packets with Hop Count = 0 and Associated Tsi620 Responses

Transaction Type	Size Field	Action taken by Tsi620	Error Logging	Notes
Read or Write Request	4 bytes	Response generated with status ok	N/A	Accepted address space = 00000 to 1FFFF
	4 bytes	Send Maintenance Response with Status Error (0111)	N/A	Address space specified > 1FFFF
	≠ 4 bytes	Send Maintenance Response with Status Error (0111)	N/A	Not supported by Tsi620
Write Request with no payload	Do not care	Send Maintenance Response with Status Error (0111)	N/A	Erred Write Request
Read Request with payload	Do not care	Send Maintenance Response with Status Error (0111)	N/A	Erred Read Request
Write Request with payload	4 bytes	Send Maintenance Response with Status Error (0111)	N/A	Size field reports incorrect payload size.
Write Response • Hop count is 0	Do not care	Send port-write and set interrupt, if enabled	Bit 8 in "RapidIO Logical and Transport Layer Error Detect CSR"	Tsi620 is not an end-point device
Read Response • Hop count is 0	Do not care	Send port-write and set interrupt, if enabled	Bit 8 in "RapidIO Logical and Transport Layer Error Detect CSR"	Tsi620 is not an end-point device
Port Write • Hop count is 0	Do not care	Send port-write and set interrupt, if enabled	Bit 9 in "RapidIO Logical and Transport Layer Error Detect CSR"	Tsi620 is not an end-point device

Table 4: Maintenance Packets with Hop Count = 0 and Associated Tsi620 Responses

Transaction Type	Size Field	Action taken by Tsi620	Error Logging	Notes
Reserved Transaction Type • Hop count is 0	Do not care	Send port-write and set interrupt, if enabled	Bit 4 in "RapidIO Logical and Transport Layer Error Detect CSR"	Tsi620 is not an end-point device
Reserved TT Type • Hop count is 0	Do not care	Send port-write and set interrupt, if enabled	Bit 4 in "RapidIO Logical and Transport Layer Error Detect CSR"	Tsi620 is not an end-point device

2.5 Multicast Event Control Symbols

Multicast Event Control Symbol (MCS or MCES) forwarding describes the process where an MCS is received on one RapidIO port is passed out through the other RapidIO ports in the Tsi620. When a RapidIO port receives an MCS, it signals to the other ports that an MCS was received. Each port can then optionally transmit the MCS when it is notified that an MCS has been received by another port.

For more information on Multicast Event Control Symbols, see the *RapidIO Interconnect Specification (Revision 1.3)*.

2.5.1 MCS Reception

When a RapidIO port receives a MCS, it does the following:

1. Raises an interrupt that can be masked.

   Interrupts are masked when the port's MCS_INT_EN bit is set in "RapidIO Port x Mode CSR".

2. Forwards the symbol to the other RapidIO ports.

   Each port then forwards the control symbol if its MCS_EN bit is set in the "RapidIO Serial Port x Control CSR"

Per-port interrupt status appears in the MCS field in "RapidIO Port x Multicast-Event Control Symbol and Reset Control Symbol Interrupt CSR". Additionally, the logical OR of all per-port Multicast Event interrupt status is available in both the "RapidIO Port x Multicast-Event Control Symbol and Reset Control Symbol Interrupt CSR" and "Switch Interrupt Status Register".

Interrupts can be cleared on a per-port basis by setting MCS to 1 in "RapidIO Port x Multicast-Event Control Symbol and Reset Control Symbol Interrupt CSR". Interrupts can be cleared for all ports by setting MCS to 1 in the SPBC version of the same register.

Additionally, the MCES pin in the Tsi620 can output an edge when an MCS is received. The MCES pin toggles to signal an MCS is received; that is, the first MCS causes the pin to go low and the second MCS causes it to go back high. The ports that are used as the source for this MCES pin is selectable using the MCS_INT_EN bit in the "RapidIO Port x Mode CSR". To select the MCES pin as an output, set MCES_CTRL to 10 in the "Switch MCES Pin Control Register". An MCS generated using this method can optionally cause an interrupt to be sent.

Due to the finite time it takes to translate an MCS to a signal on the MCES pin, the time between any two MCS received in Tsi620 must be greater than 500 ns. If this condition is not met, an MCS may be lost.

The MCES pin is only reset on assertion of CHIP_RST_b. The MCES pin is not reset by any other reset.

2.5.2 Generating an MCS

The Tsi620 supports the generation of an MCS in two ways. The first method is called the *software usage model*, which makes use of a maintenance write transaction in a port (see "RapidIO Port x Send Multicast-Event Control Symbol Register"). The write operation to this port does not complete — that is, no response is sent — until the MCS is enqueued for transmission. Subsequent writes to the register are ignored until the MCS is transmitted. A register write can be performed from both JTAG and I^2C.

It is necessary to perform multiple register writes to generate MCS on multiple ports using this method.

The Tsi620 also supports a *hardware usage model*, which generates an MCS using the MCES pin as an input. When enabled a transition on the MCES pin signals all ports that a request to transmit an MCS is received. All ports enabled to forward multicast control symbols then transmit an MCS (see "RapidIO Serial Port x Control CSR"). The time between two transitions on the MCES pin must be no smaller than 1µs. For example, when the host needs to create a "heartbeat" for the entire system at 125 kHz, it should use a 62.5 kHz clock to generate the pulse driving the MCES pin.

The MCES_CTRL setting in the "Switch MCES Pin Control Register" should be completed before traffic. Changing MCES_CTRL setting during operation may result in the transfer of a spurious MCES.

2.5.3 Restrictions

Only one Tsi620 RapidIO port should be assigned to receive MECSs.

If a single RapidIO port or multiple RapidIO ports receive MECSs closely spaced in time, only one control symbol is forwarded correctly. The other control symbols are discarded. The minimum separation between the MECSs is the time on the port with the lowest possible aggregate baud rate to send at least 64 code groups. The 64 code groups is taken from the lowest clock speed (port rate) in the system.

2.6 Reset Control Symbol Processing

One of the functions that can be performed by control symbols is requesting that the link partner reset itself. The Tsi620 can generate link-request/reset control symbols using the standard RapidIO registers defined for the purpose. The Tsi620 generates four link-request/reset control symbols with only one register access to the "RapidIO Serial Port x Link Maintenance Request CSR". For more information on reset control symbol handling, see "Resets".

2.7 Data Integrity Checking

Data integrity checking is performed on both control symbols and packets.

2.7.1 Packet Data Integrity Checking

Packets have two locations where CRC can occur. The first location is 80 bytes into the packet, while the second location is at the end of the packet. This means that packets 80 bytes or smaller have only one CRC, while packets larger than 80 bytes have two 16-bit CRC codes. With one exception, the Tsi620 does not (re)compute CRC codes for packets. The CRC code is forwarded with the packet across the Switch ISF, and the packet is transmitted with the same CRC code it was received with. This ensures that packet corruption within the Tsi620 is detected.

The exception to the rule for CRC codes is the handling of maintenance packets. Maintenance packets have a hop count field, covered by CRC, which must be changed by the Tsi620 if the packet is to be forwarded. So, CRC is recomputed for maintenance packets for each link they cross.

2.7.2 Control Symbol Data Integrity Checking

Control symbols have 24 bits, five of which are devoted to a CRC code. The CRC code is verified to ensure that the control symbol was not corrupted in transmission. Additional checks are performed on a control symbol's fields to ensure that they are valid. If the CRC check or the control symbols fields are invalid, the control symbol is discarded.

2.8 Error Management

The Tsi620 supports the Software Assisted Error Recovery registers as defined by the *RapidIO Interconnect Specification (Revision 1.3)*. For a complete list of registers supported for Software Assisted Error Recovery, see "RapidIO Physical Layer Registers".

2.8.1 Software Assisted Error Recovery

The software-assisted error recovery process is described in terms of the ackIDs of a Tsi620 port connected to a link partner that becomes mismatched. A system host[1], which can be local or remote to the Tsi620 Switch, has access to the device through another port. The system host can be any processor in a system that is tasked with error management responsibility. In a large system, multiple processors may have this responsibility. The link partner is assumed to be register compliant to the *RapidIO Interconnect Specification (Revision 1.3)*. All transactions between the system host and the Tsi620 Switch are maintenance transactions.

Before, during, and at the conclusion of the software-assisted error recovery process, monitor and clear any error bits that were set in the "RapidIO Port x Error and Status CSR".

If an ackID mismatch occurs between a Tsi620 Switch port and an endpoint, the system host manipulates registers in the Tsi620 Switch port connected to the endpoint to perform error recovery. If this occurs, the following software-assisted error recovery process can be used:

1. The system host sets the PORT_LOCKOUT bit in the Tsi620's "RapidIO Serial Port x Control CSR" in order to flush the port's ingress and egress buffers. Please note, PORT_LOCKOUT must be asserted for 50 microseconds to guarantee that all packets are flushed.
2. The system host writes and clears the PORT_LOCKOUT bit on in order to perform a maintenance transaction to the link partner.
3. The system host reads the Tsi620's "RapidIO Serial Port x Local ackID Status CSR" and makes note of the inbound, outbound, and outstanding ACK_IDs.
4. The system host instructs the Tsi620 to generate a link request to its link partner using the "RapidIO Serial Port x Link Maintenance Response CSR".
5. The system host reads the link partner's response in the Tsi620's "RapidIO Serial Port x Link Maintenance Response CSR".
6. The system host sets the Tsi620 Switch's outbound ACK_ID value to match the value in the ACK_ID_STAT field of the "RapidIO Serial Port x Link Maintenance Response CSR". The ACK_ID_STAT indicates the link partner's next expected ACK_ID.
7. The system host sends a maintenance write with a priority 2 (highest) to the link partner. The maintenance write updates the link partner's ACK_ID status register with a new OUTBOUND value that matches the Tsi620's INBOUND value, and an INBOUND value which is incremented by 1 compared to the value returned in step 5. The values must be updated before the link partner sends its maintenance response so the response has the correct ACK_ID.
   - If the link partner's implementation is such that the ackID is not updated before the maintenance response is issued, the system host must wait until the transaction times out (through the TVAL timer), re-issue the link request, and compare again the Tsi620 port's "RapidIO Serial Port x Local ackID Status CSR" values to those in the "RapidIO Serial Port x Link Maintenance Response CSR".
   - The system host should send another link request from the Tsi620 to verify that the ACK_IDs are the same.

1. This type of system host is sometimes referred to as a System Error Management Processor (SEMP).

2.9 Hot Insertion and Hot Extraction

Hot insertion and hot extraction functionality enables systems to safely add, remove, and replace components while the system continues to operate. The system host can use the Tsi620's capability to restrict the access of a newly inserted component to prevent a faulty component from negatively affecting the system.

The following fields in "RapidIO Serial Port x Control CSR" control access to the system:

- PORT_LOCKOUT – When this bit is set, only link-request/response control symbols can be exchanged. When the bit is cleared, access is controlled by OUTPUT_EN and INPUT_EN. When the PORT_LOCKOUT bit is set and the link is initialized, a port write can be sent periodically to notify the system host that a new component is added to the system.
- OUTPUT_EN – Controls whether packets other than maintenance requests/responses can be sent by the Tsi620.
- INPUT_EN – Controls whether packets other than maintenance requests/responses can be received by the Tsi620.

In "RapidIO Port x Interrupt Status Register":

- LINK_INIT_NOTIFICATION – When the PORT_LOCKOUT bit is set, this bit indicates that the link has been successfully initialized. This is an interrupt bit. To disable the generation of interrupt, set LINK_INIT_NOTIFICATION_EN to 0 in "RapidIO Port x Control Independent Register". A port write can be sent if the link is initialized.

In the "RapidIO Port x Interrupt Generate Register", the LINK_INIT_NOTIFICATION_GEN can force a LINK_INIT_NOTIFICATION interrupt to be generated. This is useful for software testing and integration.

In "RapidIO Port x Error and Status CSR":

- PORT_OK – This indicates when a port is functioning and can carry traffic.
- PORT_UNINIT – When the port is not initialized, this bit is set.

The LUTs, although not necessary, can ensure that no traffic is routed to the component being inserted or removed. For more information on lookup table functionality, see "Lookup Tables".

2.9.1 Hot Insertion

When Hot Insertion occurs at Port#N, the following steps should be completed:

1. Power up Port#N in the Tsi620.
2. Lock out Port#N by writing 1 to PORT_LOCKOUT in "RapidIO Serial Port x Control CSR".
3. Insert the card.
   - Re-initialization occurs and a port-write is received once both sides are synchronized.
4. Clear Input Error-Stop state errors in "RapidIO Port x Error and Status CSR".
   - This step is applicable only if extraction occurs on the same Port#N.

5. Send Link Request to clear Input Error-Stop states to Link Partner.
   — This step is applicable only if extraction occurs on the same Port#N.
6. Re-synchronize the Inbound and Outbound ackIDs. System host inquires about the Link Partner Inbound/Outbound ackIDs and reprogrammed Tsi620's accordingly using the "RapidIO Serial Port x Local ackID Status CSR".

Tsi620 ports on which a component insertion event can occur can be configured to notify the system host when this event occurs. The PORT_LOCKOUT bit in the "RapidIO Serial Port x Control CSR" must be set to allow the LINK_INIT_NOTIFICATION bit in the "RapidIO Port x Interrupt Status Register" to be set. To determine that a component insertion event has occurred, the system host has the option of polling the "RapidIO Port x Interrupt Status Register", or of setting the LINK_INIT_NOTIFICATION_GEN bit in the "RapidIO Port x Control Independent Register" to assert an interrupt or send port write transactions (see "RapidIO Port x Control Independent Register").

Once the system host is notified that a new component is inserted, the LINK_INIT_NOTIFICATION bit should be cleared in the "RapidIO Port x Interrupt Status Register" to stop the assertion of interrupts.

If multiple ports are inserted, only one port write is generated.

The PORT_LOCKOUT bit must be cleared to allow the system host to access the new component and to allow the new component to access the remainder of the system. The OUTPUT_EN and INPUT_EN bits must be set according to the amount of access the system designer requires to allow the new component to be brought into the system safely. Error notification for the link should also be enabled, if required by the system designer.

Before any packets can be exchanged, the OUTBOUND field in "RapidIO Serial Port x Local ackID Status CSR" must be programmed to match the INBOUND value of the other side of the link. Usually, the OUTBOUND value can be set to 0 since the component just inserted is reset. Similarly, the OUTBOUND value for the component that was just inserted must be programmed to match the INBOUND value of the Tsi620's port, contained in "RapidIO Serial Port x Local ackID Status CSR".

The next expected inbound and next outbound ackIDs of the link partner are determined through the use of link request/response control symbols.

As with a controlled reset of a link partner (see "Generating a RapidIO Reset Request to a Peer Device"), the writes of the two OUTBOUND values must occur in the order given; that is, the Tsi620 followed by the link partner. Another approach is to write the expected ackID for the inbound.

If the requests are performed in the reverse order, or if other packets are transmitted before the OUTBOUND values are programmed, the link incurs a fatal error due to an ackID mismatch.

2.9.1.1 Link Partner and Unsupported Error Recovery

If the link partner does not support the software-assisted error recovery registers, "RapidIO Serial Port x Local ackID Status CSR" will not exist in the link partner. Since it is impossible to set the link partner's OUTBOUND value in this case, the Tsi620 INBOUND value must become 0.

2.9.2 Hot Extraction

Tsi620 ports where a hot extraction event occurs should not have any transactions flowing through them in preparation for the extraction. The PORT_LOCKOUT bit must be set on the port where the hot extraction event occurs in order to drop all packets arriving from the Switch ISF for transmission, to flush any existing packets in the transmit and receive queues of the port, and to prevent new packets from being received from the device about to be extracted. At this point, the component can be safely extracted.

The LUT entries for all ports in the Tsi620 can be configured to not route any packets to the port on which the hot extraction occurs.

When hot extracting a port (Port#N) originally connected to the Tsi620, the following steps should be completed:

1. Lock out Port#N by writing 1 to PORT_LOCKOUT in "RapidIO Serial Port x Control CSR".
2. Extract the card.

   This causes Port #N to lose synchronization and leads to a port error to be asserted; PORT_UNINIT is set and PORT_OK is de-asserted ("RapidIO Port x Error and Status CSR"). INPUT_ERR_STOP and OUTPUT_ERR_STOP are also set.

2.9.3 Hot Extraction System Notification

System designers may require confirmation of when a component is extracted. There are a number of confirmation methods supported by the Tsi620.

2.9.3.1 Polling

The system can poll the PORT_OK and PORT_UNINIT bits in the "RapidIO Port x Error and Status CSR" for indication that the link partner is no longer present.

2.9.3.2 Interrupts and Port Writes

Interrupts and/or port writes can be used as part of the hot insertion and hot extraction system. For example, while the PORT_LOCKOUT bit and the LINK_INIT_NOTIFICATION bit in "RapidIO Serial Port x Control CSR" are set, interrupts are asserted (if LINK_INIT_NOTIFICATION_EN is set, "RapidIO Port x Control Independent Register") until the component is extracted. A port write can also be sent once when the link is initialized and the port is locked out; or when the port is locked out, the link re-acquires initialization.

2.9.3.3 Link Errors

Another notification scheme uses link errors that occur when a component is extracted. In this design, the PORT_DISABLE bit is set to 1 by software and the LINK_INIT_NOTIFICATION bit in "RapidIO Serial Port x Control CSR" should be set to 0. Error notification continues to be enabled. When the component is removed, lane synchronization and/or lane alignment is lost. The errors detected cause a port write and/or interrupt to be sent to the system host, indicating that a component may have been extracted.

2.10 Loss of Lane Synchronization

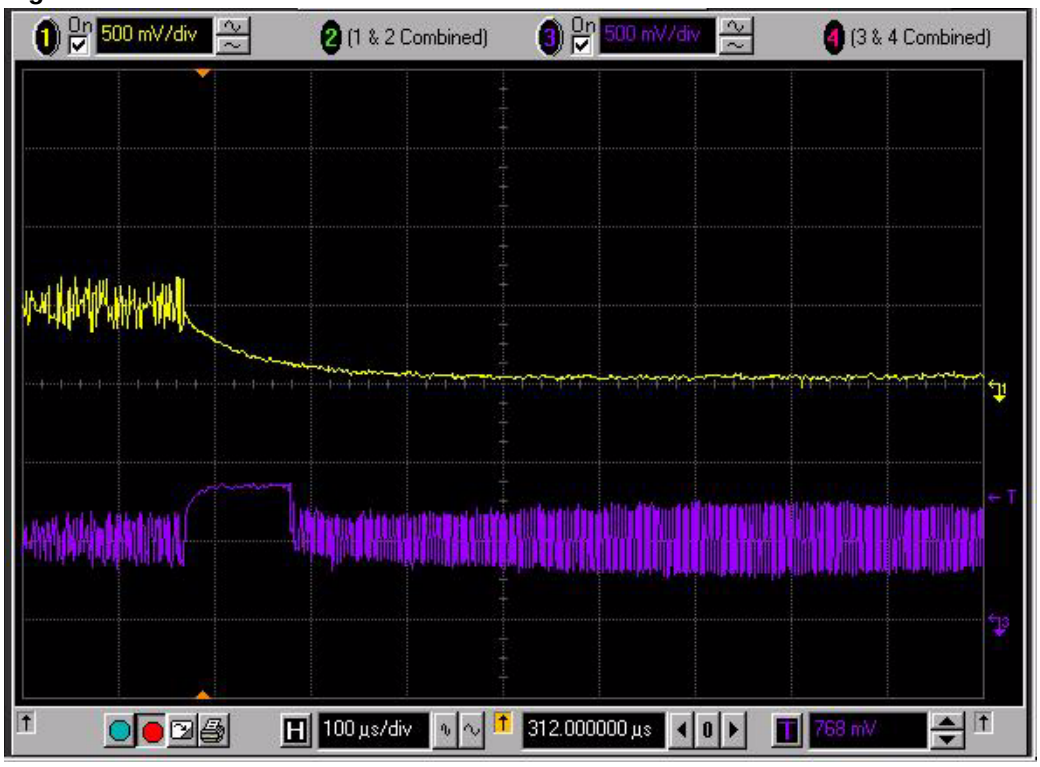
A loss of lane synchronization (LOLS) can occur due to high error rates on a link, reset of a link partner, or hot extraction of a link partner. This section discusses LOLS recovery related to high error rates on a link. For an explanation of LOLS handling due to reset of a link partner, see "Generating a RapidIO Reset Request to a Peer Device". For a discussion of LOLS handling due to hot insertion or hot extraction of a link partner, see "Hot Insertion and Hot Extraction".

When the Tsi620 detects a loss of lane synchronization (LOLS), it attempts to regain synchronization and recover so that no packets are lost, duplicated, or unnecessarily retransmitted. This is in compliance with the *RapidIO Interconnect Specification (Revision 1.3)*. To guarantee that no packets are lost, ensure the duration of the packet time-to-live timer is greater than the duration of the port's silence timer.

When a Tsi620 port detects LOLS, it restarts its synchronization state machine and stops its Timeout Interval Value (TVAL) timer for expected packet and control-symbol acknowledgements. The Tsi620 port is in input-error stopped state due to errors seen on the link. For the duration of this timer, see the TVAL field in the "RapidIO Switch Port Link Timeout Control CSR".

Figure 6 shows the Tsi620 entering the silence period when it experiences the loss of signal from its link partner.

Figure 6: LOLS Silent Period



Once synchronization is re-acquired, the Tsi620 transmitter resumes all timers and resumes sending packets from the next un-sent packet in its transmit queue, using the next available ackID. The transmitter handles the LOLS event as a temporary interruption that is ignored from the perspective of packet transfers and control symbol transfers; the actual duration of the LOLS condition has no impact on the process once the link is re-acquired.

Any packets transmitted to the Tsi620 are not acknowledged because the port is in input-error stopped state. The link partner times out waiting for a packet acknowledge control symbol, and enters output-error stopped state. To recover, the link partner sends a link-request/input-status control symbol to the Tsi620 port. This clears the input-error stopped state on the Tsi620.

The Tsi620 responds to its link partner's link-request/input-status control symbol with a link-response/status control symbol. The link partner accepts the symbol and exits the output-error stopped state. The packet associated with the next expected ackID contained in the link-response/status control symbol (if any) is then retransmitted and accepted by the Tsi620.

2.10.1 Dead Link Timer

When a LOLS event occurs, the loss of communication can continue for an extended length of time. For example, there may be an uncontrolled extraction of the link partner, and a hardware fault on the link partner. Packets continue to be directed to the non-functional (dead) link, but are not able to make forward progress. As a result, this may eventually block every traffic path in the system.

To allow systems to robustly deal with dead links, the Tsi620 has a “dead link timer” feature. This is a proprietary function that is outside of the RapidIO specification. The DLT_EN and DLT_THRESH fields in the “RapidIO SMAC x Digital Loopback and Clock Selection Register” enable/disable the dead link timer and specify the duration of the dead link timer. There is one DLT for each pair of ports (Ports N and N+1). The dead link timer can be disabled by setting DLT_EN to 0.

When the dead link timer is enabled (DLT_EN bit in the “RapidIO SMAC x Digital Loopback and Clock Selection Register”), it is reported in the PORT_ERR bit in the “RapidIO Port x Error and Status CSR”. When the PORT_ERR bit is set, and port-writes are enabled, a port-write is generated.

If the dead link timer expires, which indicates the link is no longer able to transmit or receive, then the port starts removing the impact of the dead link partner from the system. The port drops all packets in its transmit buffers. Any packets that are transferred to the port from the Switch ISF are accepted and dropped. Packets received by the port from its link partner can still be forwarded to the Switch ISF.

The dead link timer register fields affect both RapidIO ports (Ports N and N+1) sharing the Tsi620 MAC.

3. RapidIO Electrical Interface

This chapter describes the IDT-specific electrical layer features of the Tsi620 RapidIO Electrical Interface. For information on the standards-defined RapidIO features that are common to all RapidIO ports, see “RapidIO Interface”.

Topics discussed include the following:

- “Overview”
- “RapidIO Port Numbering”
- “Port Aggregation: 1x and 4x Link Modes”
- “Clocking”
- “Port Power Down”
- “Port Lanes”
- “Programmable Transmit and Receive Equalization”
- “Port Loopback Testing”
- “Bit Error Rate Testing (BERT)”

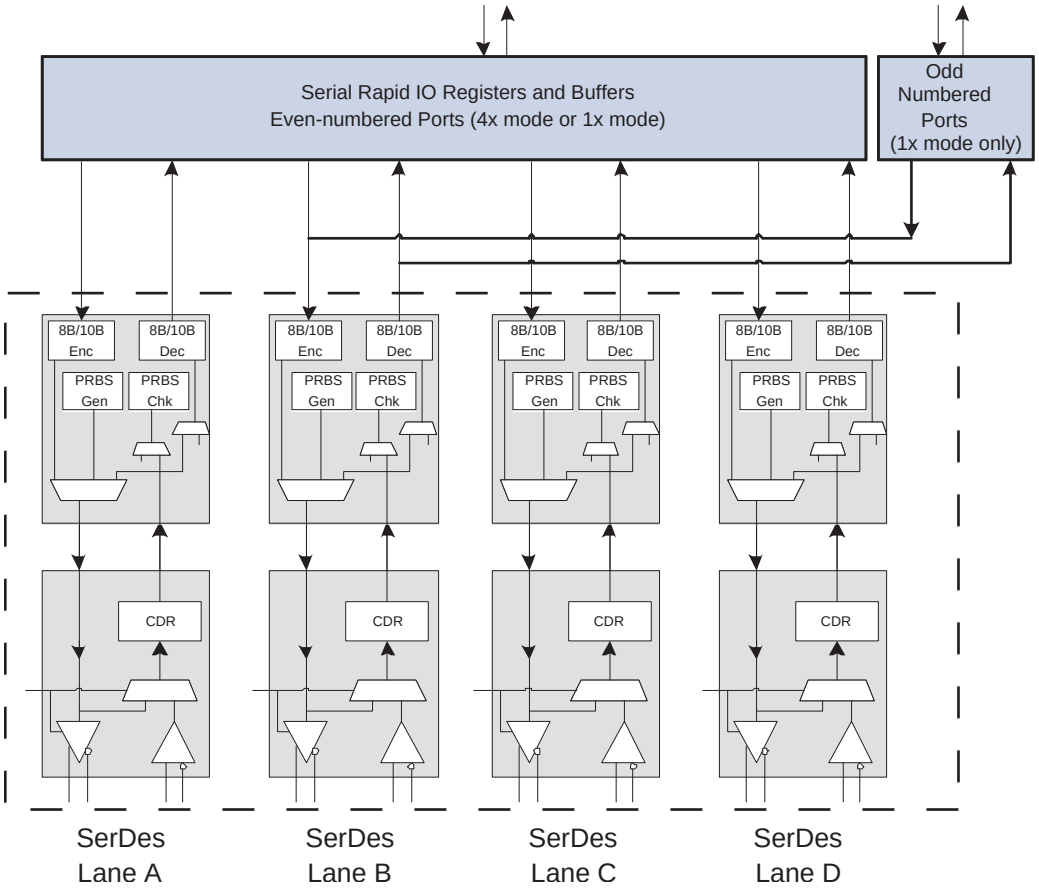
3.1 Overview

The Tsi620 Media Access Controller (MAC) has three RapidIO ports. The three ports are grouped into pairs consisting of one even numbered port and one odd numbered port. Each pair of ports share four differential transmit lanes and four differential receive lanes.

Even and odd number ports have different capabilities. Even numbered ports can operate in either 4x or 1x mode, while odd numbered ports can only operate in 1x mode. When the even numbered port is operating in 4x mode, it has control over all four differential pairs. In 4x mode, the default state of the odd numbered port is powered on. All registers in the MAC are accessible but the MAC does not have access to the PHY. However, the odd numbered port can be powered down in this configuration. When the even numbered port is operating in 1x mode, the odd numbered port can also operate in 1x mode.

Each port has flexible testing features including multiple loopback modes, bit error rate test and signal scope support (for more information, see “RapidIO Electrical Interface”). The Tsi620 MAC block is shown in the following figure.

Figure 7: Tsi620 MAC Block Diagram



The RapidIO ports include the following features:

- Up to three ports in 4x LP-Serial mode
- Up to six ports in 1x LP-Serial mode (each 4x port can be configured as two 1x ports)
- RapidIO standard operating baud rate per data lane: 1.25 Gbps, 2.5 Gbps, or 3.125 Gbps
  - 12.5 Gbps inbound and 12.5 Gbps outbound bandwidth at 3.125 Gbps for a port configured for 4x mode
  - 3.125 Gbps inbound and 3.125 Gbps outbound bandwidth at 3.125 Gbps for a port configured for 1x mode
- Supports non-standard baud rates from 1 Gbps up to 3.2 Gbps
- Programmable serial transmit current with pre-emphasis equalization
- Serial loopback with a built-in testability
- Bit error rate testing (BERT)
- Hot-insertion capable I/Os and hardware support

3.2 RapidIO Port Numbering

The RapidIO ports on the Tsi620 are numbered sequentially from 0 to 8. Table 5 shows the mapping between port numbers and the physical ports. These port numbers are used within the destination lookup tables for ingress RapidIO ports and in numerous register configuration fields.

Table 5: RapidIO Port Numbering

Port Number	RapidIO Port	Mode
0	Serial Port 0 (SP0)	1x or 4x
1	Serial Port 1 (SP1)	1x
2	Serial Port 2 (SP2)	1x or 4x
3	Serial Port 3 (SP3)	1x
4	Serial Port 4 (SP4)	1x or 4x
5	Serial Port 5 (SP5)	1x
6	Serial Port 6 (SP6) FPGA Interface	1x or 4x
7	Serial Port 7 (SP7)	1x
8	Serial Port 8 (SP8)	1x or 4x

3.3 Port Aggregation: 1x and 4x Link Modes

The RapidIO ports on the Tsi620 are grouped into pairs that share the same Serial MAC, or SMAC. The SMAC provides the PMA/PCS encoding/decoding layers, as well as the RapidIO physical, transport, and logical layer functionality required of a RapidIO switch.

The SMACs support the follow ports: serial ports 0 and 1 use SMAC 0, ports 2 and 3 use SMAC 2, and so on. Ports are grouped into pairs of N and N+1, where N is even.

Two configurations are possible on each port:

- Both port N and port N+1 can operate in 1x mode (the 1x + 1x configuration)
- Port N can operate in 4x while port N+1 is unused and can be powered down (the 4x + 0x configuration)

The *1x mode* means that one physical SerDes lane is used between link partners, and *4x mode* means that four physical lanes are used between link partners. 4x mode offers four times the bandwidth as 1x mode at the same baud rate.

Each Tsi620 SMAC has an external pin called SPx_MODE_SEL. This pin can be pulled high or low to configure the SMAC for either 1x + 1x mode or 4x + 0x mode (for information on these pins, see "Signal Descriptions"). These pins are sampled after reset is de-asserted. To ensure that the pins are sampled correctly, the pins must be held at a stable level for 10 clock cycles after reset is de-asserted. The sampled state of these pins is indicated in the PORT_WIDTH field in the "RapidIO Serial Port x Control CSR". After reset, the configuration mode can be programmed through MAC_MODE in the "RapidIO SMAC x Digital Loopback and Clock Selection Register", and the programmed value will override the pin setting of SPx_MODE_SEL. The MAC_MODE bit should be programmed *after:*

- The port is reset using the SOFT_RST_X4 or SOFT_RST_X1 bit in the "RapidIO SMAC x Digital Loopback and Clock Selection Register"
- The port is powered down with a separate register write to avoid race condition

If the SPx_MODE_SEL pin values are changed after they are sampled at reset release, it does not affect the port's operation.

The port width in use can be different from the pin-selected width; the pin indicates what the port was set to operate at while the registers show what it is actually operating at. An even port with the capability to operate in either 1x or 4x mode port can downgraded to a 1x mode port when faults on lanes prevent operation in 4x mode. Additionally, the port width can be overridden through register programming and changed into operating at a different port mode. For status and control fields for port width and "4x Configuration" for downgraded port configuration, see "RapidIO Serial Port x Control CSR".

3.3.1 1x + 1x Configuration

When the even-numbered port in a Tsi620 SMAC is configured to operate in 1x mode, the odd-numbered port in a SMAC can also be used in 1x mode. In this configuration, the even-numbered port uses SerDes lane A and the odd-numbered port uses SerDes lane B. To conserve power, the remaining two SerDes lanes should be powered down by powering down the port. To power down a port see "Port Power Down".

The two ports that share the same SMAC also share the same transmit clock, which means the two ports must have the same bit rate. To select the clock, write the IO_SPEED field (see "RapidIO SMAC x Digital Loopback and Clock Selection Register"), as described in "Clocking". The initial clock rate is selected by the global power-up option for all ports (see "Clocking").

3.3.2 4x Configuration

When the even-numbered port in a Tsi620 SMAC is configured to operate in 4x mode, the odd-numbered port in a SMAC cannot be used. To conserve power, the odd-numbered port can be powered down (see "Port Power Down").

The unusable, odd-numbered port is still a part of the Tsi620's memory map. However, system software must be aware that the port is not usable and that its per-port registers should not be accessed. If the port is accessed the Tsi620's behavior is undefined. For more information on register behavior under power-down conditions, see "Port Power Down".

The even-numbered port configured for 4x mode follows the link-width negotiation rules discussed in the *RapidIO Interconnect Specification (Revision 1.3)*. Depending on the configuration or capabilities of the link partner, or on the quality of the connection, it is possible that a port configured for 4x mode actually operates in 1x mode on either SerDes lane A or C. Under this scenario, the degraded port cannot be configured to an 1x + 1x mode.

System software can force a downgrade in port mode by writing the OVER_PWIDTH field on either the Tsi620 or in its link partner (see "RapidIO Serial Port x Control CSR"). The current operating link width is available in the INIT_PWIDTH field (see "RapidIO Serial Port x Control CSR"). Software may need to manage ackID recovery for the link partner when changing port usage between lanes A and C.

It is necessary to know if the link partner can continue to communicate when changing the port width between Lanes A and C. To determine the capability of the link partner, see "RapidIO Serial Port x Control CSR".

3.3.2.1 Degraded Link Mode

When a 4x port has degraded to a 1x mode, software may attempt to recover to 4x mode by using the FORCE_REINIT bit in the "RapidIO Port x Control Independent Register".

Connecting four 1x links to a 4x port is not supported. Doing so may result in false lane alignment

3.4 Clocking

The RapidIO ports use source clocked transmission; the clock is embedded in the data stream using 8B/10B encoding. The Tsi620 recovers the embedded clock in the received data stream and generates a separate clock to transmit its own data.

The Tsi620 uses only one external differential clock source (S_CLK_P/N) as reference to generate all internal clocks for processing the data. When the frequency of the reference clock is set at 156.25 MHz, Tsi620 can support three different RapidIO standard signaling rates (3.125 Gbps, 2.5 Gbps, and 1.25 Gbps). Table 6 shows the port speeds and bandwidths supported by the Tsi620. For more information on clocking, see "Clock, Reset, Power-up, and Initialization Options".

Table 6: Reference Clock Frequency and Supported RapidIO Data Rates

Reference Clock Frequency (S_CLK_p/n)	Supported Data Rate	SP_IO_SPEED[1:0] setting	Default speed for all ports	User Bandwidth (1x mode)	User Bandwidth (4x mode)
125 MHz	1.25 Gbps	00	1.25 Gbps	1.0 Gbps	4.0 Gbps
	2.50 Gbps	01	2.50 Gbps	2.0 Gbps	8.0 Gbps
	3.125 Gbps	10	3.125 Gbps	2.5 Gbps	10 Gbps
	N/A	11 (illegal)	undefined	undefined	undefined
156.25MHz	1.25 Gbps	00	1.25 Gbps	1.0 Gbps	4.0 Gbps
	2.50 Gbps	01	2.50 Gbps	2.0 Gbps	8.0 Gbps
	3.125 Gbps	10	3.125 Gbps	2.5 Gbps	10 Gbps
	N/A	11 (illegal)	undefined	undefined	undefined

The data rate of all the ports in Tsi620 at power-up is determined by the setting of the SP_IO_SPEED[1:0] pins and the SP_CLK_SEL[1:0] pins (see "Signal Descriptions"). There is only one pair of SP_IO_SPEED pins for the entire device, which means all RapidIO ports default to the same speed at power-up. After reset, the individual port speeds can be configured through registers (IO_SPEED in "RapidIO SMAC x Digital Loopback and Clock Selection Register") or through the I^2C configuration EEPROM.

When a pair of ports on the same SMAC are both operating in 1x mode, both ports operate at the same rate.

The settings of SP_IO_SPEED[1:0] pins and the reference clock used have a strict relationship. Entering an illegal setting causes unpredictable behavior of the device.

The *RapidIO Interconnect Specification (Revision 1.3)* requires the receive and transmit signals must operate at the same baud rate. This means a port must transmit at the same clock rate that it receives within +/-100 ppm.

3.4.1 Changing the Clock Speed

The following procedure changes the signaling rate of a port:

1. Set PWDN_X4 to 1 in the "RapidIO SMAC x Digital Loopback and Clock Selection Register".
2. Select the new clock speed using IO_SPEED in the "RapidIO SMAC x Digital Loopback and Clock Selection Register".
3. Set PWDN_X4 to 0 in the "RapidIO SMAC x Digital Loopback and Clock Selection Register".

For more information about powering down ports and special requirements for powering down port 0, see "Port Power Down".

3.4.2 Changing the Clock Speed Through I2C

The Tsi620 can be configured to power up with ports at different link speeds by setting the "RapidIO SMAC x Digital Loopback and Clock Selection Register" by an external I^2C master access or by an EEPROM boot load.

Care must be taken writing this register by an I^2C master because the port is initialized before the I^2C load is completed and therefore must follow the same rules as outlined in "Changing the Clock Speed".

Initializing the port speed using an I^2C EEPROM boot load must also follow the rules outlined in "Changing the Clock Speed" because the port is loaded with the power-up option selection on reset release and, although the port is prevented from initializing during the EEPROM boot load, the PLL is running.

The most effective way to configure the port link speed through the I^2C register load is to leave the port powered down at boot time through the SP{n}_PWRDN configuration pin and have entries in the I^2C EEPROM to load the appropriate contents of the SMACn_DLOOP_CLK_SEL to power up the port and set the correct port speed.

If port 0 requires a different speed from the default speed, two I^2C EEPROM entries are necessary because port 0 does not have a power down configuration pin. In this case, the first I^2C EEPROM entry for SMAC0_DLOOP_CLK_SEL must power down the port (SMAC0_DLOOP_CLK_SEL = 0xXXXXXXXC). The second I^2C EEPROM entry can power up the port and set IO_SPEED field in the SMAC0_DLOOP_CLK_SEL register to the correct value.

3.4.3 Support for Non-standard Baud Rates

Non-standard baud rates are supported between 1 Gbaud and 3.2 Gbaud by changing the reference clock frequency proportional to the required baud rate. Examples of non-standard baud rates and reference clock frequencies are displayed in Table 7.

Table 7: Non-standard Baud Rates

Baud Rate	Reference Clock Frequency	SP_IO_SPEED[1:0] Pin Setting	SMACx_DLOOP_CLK_SEL[IO_SPEED] Setting
1.2288 Gbaud	122.88 MHz	0b00	0b00
2.4576 Gbaud	122.88 MHz	0b00	0b01
3.0720 Gbaud	153.6 MHz	0b01	0b10

3.5 Port Power Down

All of the Tsi620 RapidIO ports can be powered down to minimize power consumption when the port is not required. However, port 0 has special power-down requirements that must be followed (see "Special Conditions for Port 0 Power Down").

When a port is powered down, some registers return 0 and all writes to these registers are ignored. These values indicate that the port is an uninitialized RapidIO port. The following register types are read only and return zero when a port is powered-down:

- "RapidIO Physical Layer Registers"
- "RapidIO Error Management Extension Registers"
- "IDT-Specific RapidIO Registers"

Both the "RapidIO Port x Error and Status CSR" and "RapidIO Serial Port x Control CSR" registers return 0x00000001 when read instead of 0s.

The following register types can be read and written to when a port is powered-down:

- "Serial Port Electrical Layer Registers"
- "Switch ISF Registers"
- "I2C Registers"
- "Switch Utility Registers"

3.5.1 Default Configurations on Power Down

When a port is powered down, the port loses configuration information that is stored for that specific port. For example, multicast settings and port write settings return to their default power-up settings after a port reset. After port reset, there is no method to determine that the configuration for a specific port is correct.

For example, if a port is shut down and then restored, the port write destination ID in that port is reset to the default value. The port write destination ID must be reset for the whole device after a port is shut down and restored. Similarly, multicast settings for the entire device must be re-written when a port is shut down and restored.

Per-port copies of registers that control the operation of all Tsi620 Switch ports are located in "Per Port Copies of Global Registers". These registers must be reprogrammed when a port is reset, or is powered up after a power down.

3.5.1.1 Registers to Program After Power Reset

The Unicast registers are all accessible through per-port copies of these registers. If it is suspected that a port has been powered down and up by rogue software, the per-port copies of the unicast registers can be checked. For more information on the per-port copies of the unicast registers, see "Per Port Copies of Global Registers".

For both unicast and multicast traffic, the following registers must be re-programmed with the configuration values when a port is reset:

Unicast

- "RapidIO Component Tag CSR"
- "RapidIO Route LUT Attributes (Default Port) CSR"
- "RapidIO Multicast DestID Configuration Register"
- "RapidIO Multicast DestID Association Register"
- "RapidIO Switch Port Link Timeout Control CSR"
- "RapidIO Switch Port General Control CSR"
- "RapidIO Port Write Target Device ID CSR"

Multicast

- "RapidIO Multicast Register Version CSR"
- "RapidIO Multicast Maximum Latency Counter CSR"
- "RapidIO Port x Switch ISF Watermarks Register"
- "RapidIO Port x Prefer Unicast and Multicast Packet Priority 0 Register"
- "RapidIO Port x Prefer Unicast and Multicast Packet Priority 1 Register"
- "RapidIO Port x Prefer Unicast and Multicast Packet Priority 2 Register"
- "RapidIO Port x Prefer Unicast and Multicast Packet Priority 3 Register"

3.5.2 Special Conditions for Port 0 Power Down

Port 0 can only be powered down through register accesses and not through the SPn_PWRDN pin. Port 0 has the has following special conditions after it is powered down:

- The list of registers in "Default Configurations on Power Down" cannot be read while port 0 is in reset
- After reset the listed registers must be re-written (like any other port that was powered down)

3.5.3 Power-down Options

The following power-down options are available on a port:

- A port's main logic can be powered down at boot up through the SPn_PWRDN pins.
- The default configuration provided by the pins can be changed through the SMACx_DLOOP_CLK_SEL registers. This occurs during boot up through an EEPROM on the I^2C bus, or during normal operation through a register write.

3.5.4 Configuration and Operation Through Power Down

The I/Os for the individual bit lanes can be powered down when they are not used. All valid power-down scenarios are shown in Table 8.

Table 8: RapidIO Port Power-down Procedure

Mode for RapidIO Port *n*	Mode for RapidIO Port *n+1*	Required Power-Down Configuration
4x	N/A	• De-assert the SP*n*_PWRDN pin and/or set the PWDN_X4 bit to 0 in the "RapidIO SMAC x Digital Loopback and Clock Selection Register" • To conserve power, assert the SP*n+1*_PWRDN pin and/or set the PWDN_X1 bit to 1 in the "RapidIO SMAC x Digital Loopback and Clock Selection Register". If this bit is not set, Port n+1 consumes unnecessary power.
1x	1x	• De-assert the SP*n*_PWRDN pin and/or set the PWDN_X4 bit to 0 in the "RapidIO SMAC x Digital Loopback and Clock Selection Register" • De-assert the SP*n+1*_PWRDN pin and/or set the SMACx_DLOOP_CLK_SEL[PWDN_X1] register bit to 0
1x	Port Not Used	• De-assert the SP*n*_PWRDN pin and/or set the PWDN_X4 bit to 0 in the "RapidIO SMAC x Digital Loopback and Clock Selection Register" • To conserve power, assert the SP*n+1*_PWRDN pin and/or set the PWDN_X1 bit to 1 in the "RapidIO SMAC x Digital Loopback and Clock Selection Register". If this bit is not set, Port n+1 consumes unnecessary power.
Port Not Used	1x	• Not supported
Port Not Used	Port Not Used	• To conserve power, assert the SP*n*_PWRDN pin and/or set PWDN_X4 bit to 1 in the "RapidIO SMAC x Digital Loopback and Clock Selection Register" • To conserve power, assert the SP*n+1*_PWRDN pin and/or set the PWDN_X1 bit to 1 in the "RapidIO SMAC x Digital Loopback and Clock Selection Register"

3.5.4.1 Signals Sampled After Reset

After a "Chip Reset" is de-asserted, the Tsi620 samples the state of the power-down pins and only powers up the ports that are enabled. Each RapidIO port has a unique pin, SPn_PWRDN.

Port 0 is the default port.

The sampled state of the pins is available in the "RapidIO SMAC x Digital Loopback and Clock Selection Register". This register can be overwritten, allowing the system software to override the pin-based configuration. This can be completed at any time — during boot-up through the I^2C interface, JTAG, or during normal operation through the RapidIO ports — allowing the system software to override the pin-based configuration.

3.5.4.2 4x Mode and Odd Ports

When the pair of ports sharing the same SMAC (for example, ports 2 and 3) are configured in 4x mode, only the even-numbered port is used. The odd numbered port should be powered down by software or configuration pins to minimize power consumption.

3.6 Port Lanes

For ports that support both 1x and 4x mode functionality, even and odd number ports have different capabilities. Even numbered ports can operate in either 4x or 1x mode, while odd numbered ports can only operate in 1x mode. When the even numbered port is operating in 4x mode, it has control over all four differential pairs (designated Lanes A, B, C and D).

In 4x mode, the default state of the odd numbered port is powered on. All registers in the even and odd numbered port are accessible but the odd numbered port does not have access to the PHY. In order to decrease the power dissipation of the port, the odd numbered port can be powered down in this configuration. When the even numbered port is operating in 1x mode it uses only Lane A and the odd numbered port is permitted to operate in 1x mode using Lane B.

3.6.1 Lane Synchronization and Alignment

When coming out of reset, the Transmit side of the port must continuously send out /K28.5/ code groups on each lane to assist the Receive side of its link partner to synchronize. Once a /K28.5/ code group is detected by the Receive port, another 127 /K28.5/ code groups must be received error free before the Receive port can declare synchronized. No other useful information is communicated between the link partners until the ports are synchronized.

For a 4x port, after lane synchronization is complete, lane alignment starts. The port transmits /A/'s (||A||) on all four lanes, according to the *RapidIO Interconnect Specification (Revision 1.3)* idle sequence generation rules. Reception of four ||A||'s without the intervening reception of a misaligned column is the condition for achieving lane alignment. A misaligned column (that is a column with at least one ||A|| but not all ||A||s in a row) causes an error to be asserted. Bit errors, or receptions of rows without all /A/'s, result in sampling/buffering adjustments.

For more information, see the *RapidIO Interconnect Specification (Revision 1.3) Part 6: Physical Layer 1x/4x LP-Serial Specification*.

3.6.2 Lane Swapping

Lane Swap is the ability to reverse the order of the transmit and receive pins. The Tsi620 allows the order of the transmit and/or receive pins of each 4x port to be reversed to simplify board layout issues. Lane swap is only supported when the SMAC is operating in 4x mode.

Lane swap for 1x mode is not supported.

3.6.2.1 Configuration

On de-assertion of CHIP_RST_b, the lane swap setting for the entire device is controlled by two configuration pins: SP_RX_SWAP and SP_TX_SWAP (see "RapidIO Signals").

The SWAP_TX and SWAP_RX fields in the SMACx_DLOOP_CLK_SEL register can also be written at reset time by I^2C initialization or by software override to set lane swap on a per SMAC basis (see "RapidIO SMAC x Digital Loopback and Clock Selection Register"). The reset value of these fields indicates the sampled values of the SP_RX_SWAP and SP_TX_SWAP configuration pins. When a different value is written to either the SWAP_TX or SWAP_RX fields, the SMAC must be reset to ensure that the links retrain and communication can be established with the changed lane configuration.

When changing the lane swap setting for a SMAC, it is necessary to reset the port through the RST_X4 fields in the SMACx_DLOOP_CLK_SEL Register (see "RapidIO SMAC x Digital Loopback and Clock Selection Register").

3.6.2.2 Lanes and Channels

The terms lanes and channels identify input and output signals. Lanes are enumerated using alphabetic characters (A, B, C, D). The pin associated with a lane changes depending on the lane swap settings.

Channels are numbered 0 through 3. Channels are never reordered. When lanes are not swapped, the following mapping between channels and lanes is used:

- Channel 0 maps to Lane A
- Channel 1 maps to Lane B
- Channel 2 maps to Lane C
- Channel 3 maps to Lane D

When lanes are swapped, the following mapping between channels and lanes is used:

- Channel 0 maps to Lane D
- Channel 1 maps to Lane C
- Channel 2 maps to Lane B
- Channel 3 maps to Lane A

Even numbered ports in 4x mode are associated with lanes A-D. When an even numbered port is in 1x mode, it is associated with lane A. Odd numbered ports are associated with lane B.

3.6.2.3 Tx and Rx Swapping

The operations in the SMACx_DLOOP_CLK_SEL register are associated with lanes. When lanes are swapped, the channels associated with these operations must change. The user can independently swap only the Tx or Rx lanes.

Operations on channels, as supported by the SMAC Channel Configuration registers, operate on the specific channels regardless of the lane swap settings for a SMAC (see "RapidIO SMAC x SerDes Configuration Channel 0 Register"). If lane swap functionality is enabled in the system, the proper channels must also be configured. The channel number of a transmit lane and a receive lane differ when Tx lanes are swapped and Rx lanes are not, or vice versa.

Swapping while Doing Loopback

Parallel and analog loopback modes have to operate when both Tx and Rx lanes are swapped *together.* These loopback paths are performed on a channel basis. As such, only a Tx or only an Rx swap during these modes will create unpredictable results.

Switch ISF loopback and digital loopback modes can be used in any lane swap mode, because these loopback paths are associated with lanes.

3.7 Programmable Transmit and Receive Equalization

The Tsi620 has programmable drive strengths and de-emphasis of a transmit lane. The Tsi572 also has the ability to internally boost the received signal. This functionality is described in the following sections.

3.7.1 Transmit Drive Level and Equalization

The Tsi620 has programmable drive strengths and de-emphasis of a transmit lane. This ability adjusts for the electrical characteristics that can degrade the signal quality of a link which connects a device to the Tsi620. Decreasing the drive strength of signals also provides the ability to reduce the power consumption of a port.

The drive strength current of each lane can be controlled through the TX_LVL field (see "RapidIO SMAC x SerDes Configuration Global Register" and Table 159), the TX_ATTEN, and the TX_BOOST fields in the SMAC x SerDes Configuration - Channel 0 through Channel 3 registers (see "RapidIO SMAC x SerDes Configuration Channel 0 Register").

The nominal drive strength for each lane can be adjusted from full to half strength by using the TX_ATTEN[2:0] field in the SMAC x SerDes Configuration - Channel 0 through Channel 3 registers (see "RapidIO SMAC x SerDes Configuration Channel 0 Register"). The actual drive strengths available are 16/16, 14/16, 12/16, 10/16, 9/16 and 8/16 of the nominal driver voltage.

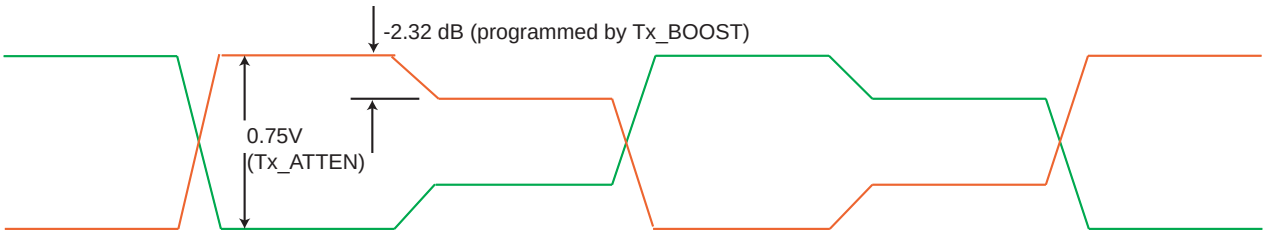
The de-emphasis functionality can be programmed by the TX_BOOST field in the "RapidIO SMAC x SerDes Configuration Channel 0 Register" register. The TX_BOOST field controls the drive level of subsequent non-transitional bits with respect to the transitional ones. The amount of de-emphasis is specified as a ratio of the de-emphasis drive strength to the TX_LVL drive strength, in steps of ~0.37dB.

The Nominal Drive Level is 1.0 V +/-10%.

Figure 8 shows the drive current and equalization settings and where they impact a transmission waveform. The formulae of calculating the actual TX_ATTEN and TX_BOOST are shown in "RapidIO SMAC x SerDes Configuration Channel 0 Register". A non-zero setting of TX_ATTEN forces the TX_BOOST value to 0dB.

Another register control, TX_LVL[4:0], affects the Tx signal swing. For this process node, the TX_LVL should be set at a minimum of 1 V, and for long reach compliance can be programmed up to 1.26 V.

Figure 8: Drive Strength and Equalization Waveform



3.7.2 Receive Equalization

On the Receive end, the received signal can be internally boosted up by controlling the register RX_EQ_VAL ("RapidIO SMAC x SerDes Configuration Channel 0 Register"). The equation involving the 3-bit values of the register field are described by:

Receiver boost = (RX_EQ_VAL + 1)*0.5 dB

For example, setting RX_EQ_VAL[2:0] = 0b100 generates a 2.5dB boost of the received signal. This boost is internal to the device and is useful improving the signal at the slicer when the signal arriving at the pins are degraded.

3.8 Port Loopback Testing

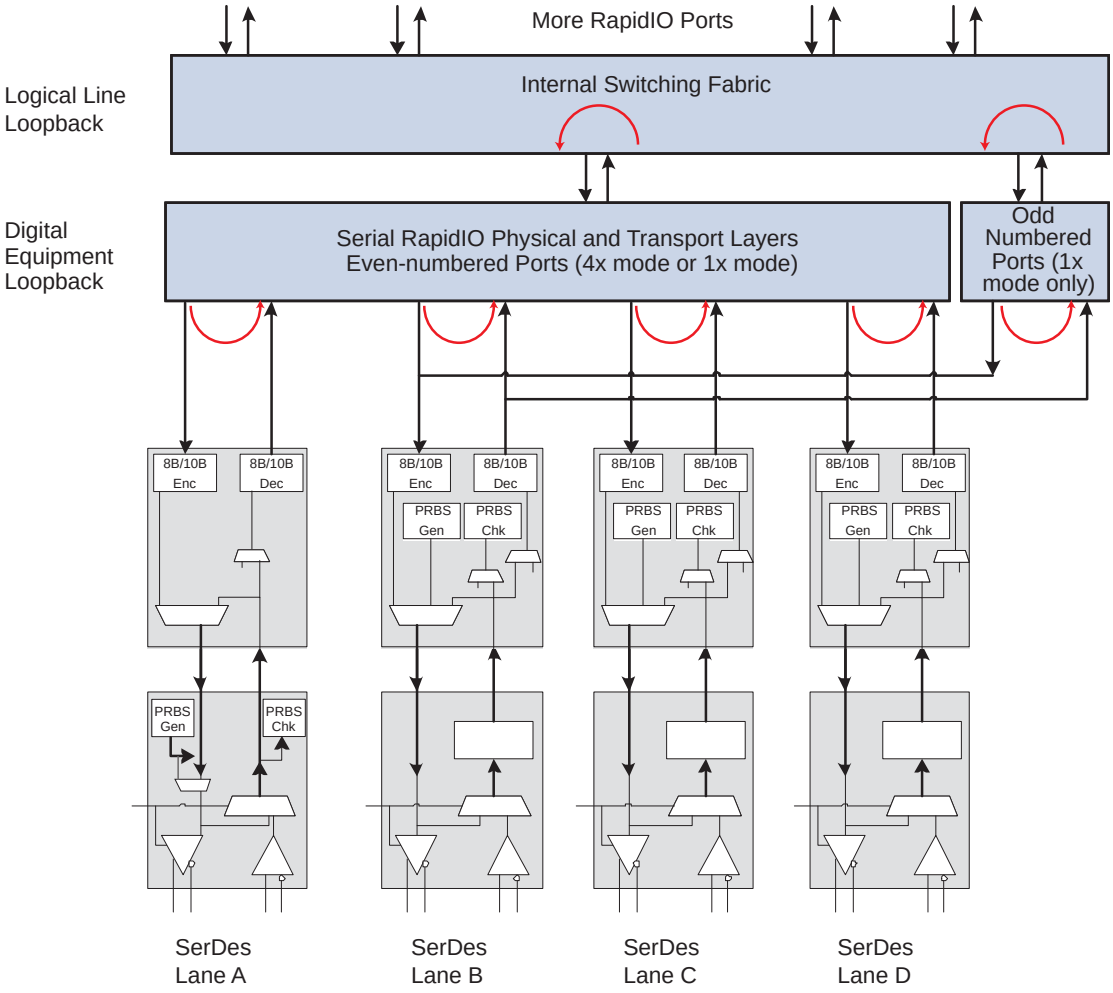
The Tsi620's RapidIO ports support the following kinds of loopback:

- Digital equipment loopback
- Logical line loopback

The Internal Switch Port does not support any loopback test modes.

Figure 9 shows where each loopback resides in the Tsi620 SMAC.

Figure 9: SMAC Loopbacks



3.8.1 Digital Equipment Loopback

When this form of loopback is enabled, the RapidIO port transmit logic is connected to the receive logic just before the 8B/10B encoder and transmitter. Digital equipment loopback requires the use of packets and a correctly configured lookup table in the port that is being looped back. The Bit Error Rate Tester patterns cannot be used when in Digital Equipment Loopback mode. The SerDes does not have to be trained or operational for this loopback to function since the SerDes PHY is not included in the data path.

All incoming data for the port on its external link is ignored when digital equipment loopback is enabled.

3.8.2 Logical Line Loopback

Logical line loopback causes a packet sent into the Tsi620's Switch ISF to be directed back to the originating port. To cause packets to loop back in this fashion, configure the destination ID lookup tables (LUTs) so all destination IDs are destined for the incoming port.

For more information on LUT programming, see "Lookup Tables".

3.9 Bit Error Rate Testing (BERT)

The RapidIO ports on the Tsi620 have a built-in bit error rate test (BERT). This test is based either on fixed symbols or on a pseudo-random bit sequence (PRBS). Each lane within a port has a pair of Pattern Generators and Pattern Matchers.

BERT patterns are not framed RapidIO packets and, therefore, when running BERT testing in Tsi620 the word alignment has to be turned off. This can be completed by de-asserting RX_ALIGN_EN bit for the corresponding lane (see "RapidIO SMAC x SerDes Configuration Channel 0 Register").

3.9.1 BERT Pattern Generator

The BERT Pattern Generator can generate different patterns when the link is put into test mode. Table 9 shows what patterns are supported by programming the register.

Table 9: Patterns Supported by Generator

MODE Setting	Description
0	Pattern Generator is disabled
1	15^{th} order linear feedback shift register (LFSR[a]) polynomial: $x^{15} + x^{14} + 1$
2	7^{th} order LFSR polynomial: $x^7 + x^6 + 1$
3	Fixed 10-bit pattern from bottom of PAT0 field
4	2 byte DC balanced pattern constructed as {PAT0, ~PAT0}
5	4 byte DC balanced pattern constructed as: {0x000, PAT0, 0x3FF, ~PAT0}
6, 7	Reserved

a. Linear Feedback Shift Register.

BERT testing is enabled on a per-bit lane basis, and normal traffic flow on the bit lane ceases when BERT testing is enabled. To enable BERT testing, program the SMACx_PG_CTL registers to enable either normal operation, PRBS-based BERT, or fixed-pattern-based BERT.

BERT testing can be performed when a Tsi620 SMAC is across a link from one Tsi620 SMAC to another Tsi620 SMAC.

When testing a link on the Tsi620 with the BERT feature, the link partner device must support PRBS testing with at least one of the two polynomials shown in Table 9, or it must support fixed-pattern tests. Alternatively, the link partner must support some form of loopback to the Tsi620. Consult the appropriate documentation for other devices to determine if they support these features, and to determine how to configure them.

Other PRBS test sequences may be unsuitable for testing in an AC coupled system. The PRBS pattern must ensure that it does not introduce baseline wander and cause an unrealistically high bit error rate. The PRBS patterns generated by the Tsi620 are DC balanced.

3.9.2 BERT Pattern Matcher and Error Counter

The pattern matcher is capable of synchronizing to and detecting erroneous bytes in the two LFSR patterns mentioned in Table 10. Erroneous bytes are counted in the error counter in the "SerDes N Lane 0 Pattern Matcher Control Register".

Table 10: Patterns Supported by Matcher

MODE Settings	Description
0	Pattern Matcher and Error Counter are disabled
1	Expect 15^{th} order lfsr polynomial: $x^{15} + x^{14} + 1$
2	Expect 7^{th} order lfsr polynomial: $x^7 + x^6 + 1$
3	Expect d[n]=d[n-10]
4	Expect d[n]=!d[n-10]
5:7	Reserved

The Pattern Generator and Matcher are independently controllable within the same lane. They do not need to be enabled, or programmed, the same way. For example, the Tsi620 can transmit a different PRBS pattern than the pattern it is receiving.

When the MODE bit is set to 0b001 or 0b010, the Pattern Matcher operates by generating the expected pattern and synchronizing to the incoming pattern.

The COUNT field in the "SerDes N Lane 0 Pattern Matcher Control Register"is a 15-bit value. Together with the OV14 bit, a total of $2^{22} - 2^7 - 1$ errors can be reported. When OV14 is set, the COUNT value should be read as count*128.

3.9.3 Fixed Pattern-based BERT

Fixed pattern-based BERT uses data in software-configurable registers to send an alternating pattern of 10-bit 8B10B code groups. Fixed pattern-based BERT does not produce error count results.

Fixed patterns are programmed in the PAT0 and select by setting the appropriate MODE field in the "SerDes N Lane 0 Pattern Matcher Control Register".

The following three patterns are useful for BERT testing:

- pat0 = 1010101010 creates a high-frequency pattern, with SMACx_PG_CTL.mode=0b011
- pat0 = 0011111000, ~pat0 = 1100000111 creates a low-frequency pattern, with SMACx_PG_CTL.mode=0b100

3.9.3.1 Fixed Pattern-based BERT — Transmitter Configuration

To configure a Tsi620 transmitter for fixed-pattern BERT operation:

1. Write the bit stream to be transmitted into the PAT0 field in the "SerDes N Lane 0 Pattern Generator Control Register".
2. Set the MODE field in the same register to the desired fixed pattern mode (mode = 011:100).

   Setting this field causes the software defined pattern to transmit.

3.9.3.2 Fixed Pattern-based BERT — Receiver Configuration

The Pattern Matcher can only match fixed-pattern mode of {pat0,pat0} and {pat0, ~pat0}. The error counting method is the same as described in "BERT Pattern Matcher and Error Counter".

1. Tell the transmitter to stop sending PRBS pattern by setting the MODE field to 0 in the "SerDes N Lane 0 Pattern Generator Control Register".
2. Re-enable the receiver's framer by writing to the RX_ALIGN_EN bit in the "RapidIO SMAC x SerDes Configuration Channel 0 Register".

4. FPGA Interface

This chapter describes the IDT-specific electrical layer features of the Tsi620 FPGA Interface. For information on the standards-defined RapidIO features that are common to all RapidIO ports, see "RapidIO Interface".

Topics discussed include the following:

4.1 Overview

The FPGA Interface supports a single, 4x/1x RapidIO port. It is a parallel interface that connects to low-cost FPGAs, and has flexible test features including multiple loopback modes (for more information, see "RapidIO Electrical Interface").

The FPGA Interface includes the following features:

- Configurable as a single 4x/1x RapidIO port
- Operates at 1, 2, or 2.5 Gbps data rate per lane:
  - Maximum 10 Gbps for 4x mode
  - Maximum 2.5 Gbps for 1x mode
- Standards-based interface features:
  - Is a subset of the 10-Gb Ethernet XGMII Interface standard. XGMII specifies a standard connection to external 10-Gb Ethernet PHYs
  - Transmit Interface consists of the following:
    - Four data lanes, each 8 bits in size
    - Four control signals, one for each data lane, to distinguish RapidIO 'K' characters from data
    - One clock signal whose edges are center-aligned with the data and control signals
    - One TxDisable signal, indicating that the data being sent should be interpreted as errors/silence on the link

— Receive Interface consists of the following:

– Four data lanes, each 8 bits in size

– Four control signals, one for each data lane, to distinguish Serial RapidIO ‘K’ characters from data

– One clock signal whose edges are center-aligned with the data and control signals

– One RxError signal, indicating that the data being received on the link should be interpreted as silence/error

— Transmit clock signal supports frequency options of 156.25, 125, or 62.5 MHz

— Receive clock signal must be frequency locked to the transmit clock signal. This is typically done by driving the link partners with the same clock source.

— Uses Dual Data Rate (DDR) signaling - data is presented on both edges of the clock signal

— Supports center-alignment relationship between clock and data/control signals

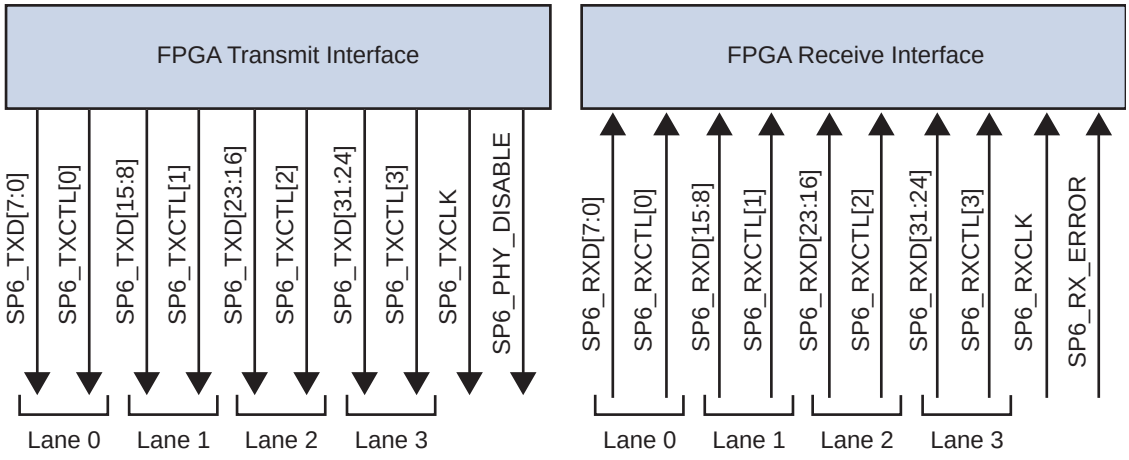
— Supports HSTL1 (High Speed Transceiver Logic) 1.5V electrical standard.

- Serial loopback with a built-in testability
- Can be powered down to conserve power when not in use

4.2 FPGA Electrical Interface

The FPGA Interface operates in the same method as the other 4x/1x RapidIO ports (see “RapidIO Electrical Interface”). The FPGA Interface consists of a transmit path and a receive path, as displayed in the following figure.

Figure 10: FPGA Interface Signals



The FPGA Interface uses source clocked transmission; there are separate clock signals for transmit and receive data. The data and control signals are dual data rate. For more information on the FPGA signals, see “FPGA Interface Signals”.

The FPGA Interface's link partners must be frequency locked to each other. This is completed by feeding the same clock source to both link partners. No clock phase relationship is specified or required between the link partners.

4.2.1 Link Initialization

The electrical aspects of initializing the FPGA Interface are different from a typical RapidIO link. A typical RapidIO link requires the transfer of an idle sequence for a period of time in order for the SerDes to recover and synchronize to the clock signal embedded in the idle sequence. The FPGA Interface has a discrete clock signal. Even though the link initialization sequence can be simpler, the FPGA Interface is compliant to the RapidIO link initialization sequence.

When the port comes out of reset, the Transmit Interface must continuously send out /K28.5/ code groups on each lane. Once a /K28.5/ code group is detected by the Receive Interface, another 127 /K28.5/ code group must be received error free before the Receive Interface can be declared synchronized. No other useful information is communicated between the link partners until the ports are synchronized.

For a 4x port, after lane synchronization is complete, lane alignment starts. The port transmits /A/'s (||A||) on all four lanes according to the *RapidIO Interconnect Specification (Revision 1.3)* idle sequence generation rules. Since the FPGA data on all lanes is aligned with the transmit/receive clock, lane alignment should be successful.

For more information, see the *RapidIO Interconnect Specification (Revision 1.3) Part 6: Physical Layer 1x/4x LP-Serial Specification*.

The FPGA Interface expects the data and control signals it receives to be valid when the SP_RX_ERROR signal indicates that the data is valid. Similarly, the SP_TX_DISABLE signal indicates that the transmission is valid.

It takes less than 20 usec after a reset for the PLLs to lock to the reference clock. During that time, the transmit clock is not driven and no data can be received, regardless of the state of the receive clock signal. The SP_TX_DISABLE signal is asserted until the PLLs are locked.

When the SP_RX_ERROR signal indicates valid data, the standard RapidIO link initialization state machine operates to initialize the link. This state machine specifies when each data lane is enabled and disabled. Note that all lanes are valid or all lanes are invalid for the FPGA Interface.

Individual lanes cannot be disabled on the FPGA Interface. The SP_RX_ERROR signal indicates that data on all lanes is valid, or that data on all lanes is invalid. The user must set the configuration pins for the FPGA Interface to the correct width, 4x or 1x, using the SP6_MODE_SEL pin (see "RapidIO Signals").

FPGA link initialization does not support downgrade from 4x to 1x. It also does not support link initialization when a 1x port is connected to the FPGA Interface and the interface is configured to operate in 4x mode.

4.2.2 Clocking

The FPGA Interface uses the external differential clock source, S_CLK_P/N, as reference to generate all internal clocks for transmitting data. The reference clock can be either 125 or 156.25 MHz. The FPGA Interface supports three signaling rates: 1.0 Gbps, 2.0 Gbps, and 2.0 Gbps (see Table 11; see also "Clock, Reset, Power-up, and Initialization Options").

Table 11: FPGA Reference Clock Frequency and Supported RapidIO Data Rates

Reference Clock Frequency (S_CLK_P/N)	SP_CLK_SEL setting	Supported Data Rate Per Lane	SP_IO_SPEED [1:0] setting	SerDes Baud Rate	FPGA Interface Clock Rate
125 MHz	0b00	1.0 Gbps	00	1.25 GBaud	62.5 MHz
		2.0 Gbps	01	2.5 GBaud	125 MHz
		2.5 Gbps	10	3.125 GBaud	156.25 MHz
		N/A	11 (illegal)	N/A	N/A
156.25 MHz	0b01	1.0 Gbps	00	1.25 GBaud	62.5 MHz
		2.0 Gbps	01	2.5 GBaud	125 MHz
		2.5 Gbps	10	3.125 GBaud	156.25 MHz
		N/A	11 (illegal)	N/A	N/A

The data rate of all the Tsi620's RapidIO ports at power-up is determined by the setting of the SP_CLK_SEL[1:0] and SP_IO_SPEED[1:0] pins (see "Power-up"). There is only one pair of SP_IO_SPEED pins for the entire device, which means all RapidIO ports operate at the same data rate. After reset, the individual port speeds can be configured through the IO_SPEED field in the "RapidIO SMAC x Digital Loopback and Clock Selection Register", or through I^2C.

The FPGA Interface can use non-standard reference clock frequencies, the same as the Tsi620 SMAC (see "Support for Non-standard Baud Rates").

The settings of the SP_IO_SPEED[1:0] pins and the reference clock used have a strict relationship. Entering illegal settings will cause unpredictable behavior of the device.

Unlike the RapidIO ports (SMAC), the link partner on the FPGA Interface must have the same clock source as the FPGA Interface. This means that the link partner is frequency locked to the FPGA Interface, just as is required by the XGMII specification.

Despite having a separate clock signal, the idle sequence/clock compensation sequence is still transmitted by the FPGA Interface in accordance with the *RapidIO Interconnect Specification (Revision 1.3) Part 6: Physical Layer 1x/4x LP-Serial Specification.*

The FPGA Interface expects to receive the idle sequence/clock compensation sequence in accordance with the *RapidIO Interconnect Specification (Revision 1.3) Part 6: Physical Layer 1x/4x LP-Serial Specification.*

Software configuration of the FPGA Interface's clock speed is supported in the same method as per the other RapidIO ports, using IO_SPEED in the "RapidIO SMAC x Digital Loopback and Clock Selection Register" (see "Clocking").

When the FPGA Interface is released from reset, the signal on the SP6_RXCLK must be one of two values:

- Zero
- A clock signal which meets the requirements for the FPGA Interface in the electrical section (see "HSTL (FPGA Interface) Signals")

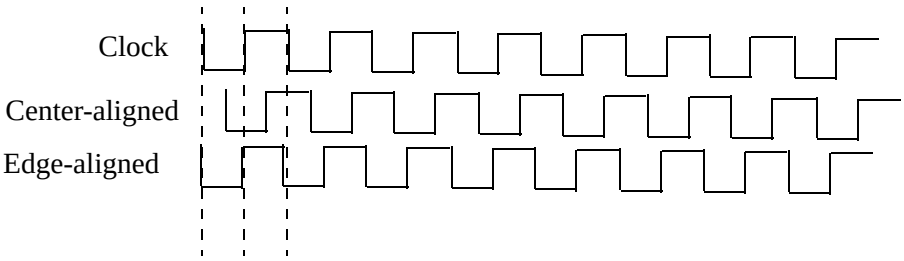
FPGA logic must either:

- Ensure that the transmit clock driven by the FPGA is 0 until the internal PLLs are locked, or
- Transmission of the transmit clock must be enabled by assertion of one or more of the Tsi620 GPIO pins

4.2.2.1 Clock/Data Alignment

DDR interfaces can be used with two different alignments of clock and data signals for the transmit and receive interfaces: center alignment and edge alignment (see Figure 11).

Figure 11: Center-Alignment versus Edge-Alignment



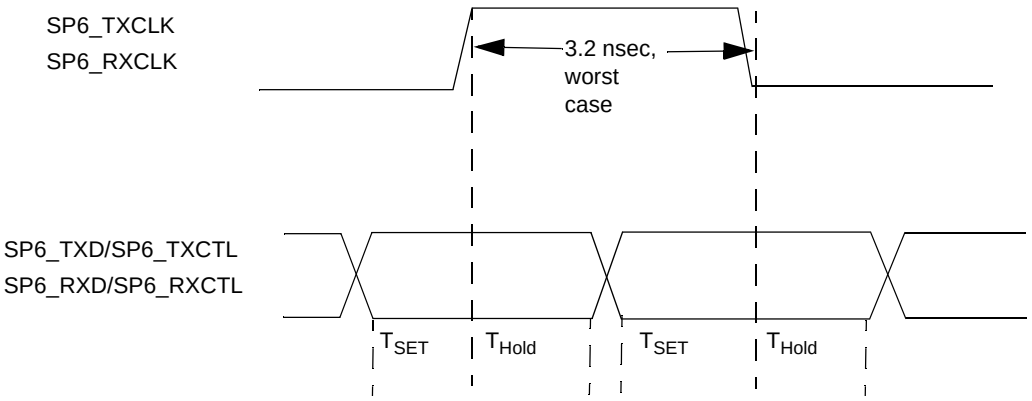
As shown in the figure, when clock and data are edge-aligned, the edges of the clock line up with the changes in the data signal. In this case, it is up to the receiver to delay sampling the data signals until they are valid. When center-alignment is used, the edges of the clock line up with the middle of changes in the data signal. In this case, the receiver should sample the data signals when the clock signal presents an edge.

The FPGA Interface only supports center-alignment on both its transmit and receive paths. The transmit clock is center-aligned to the data and control signals transmitted. The receive clock must be center-aligned to the data received. The specific timing parameter relationships are shown in Figure 12. The timing parameter values are shown in Table 12.

Table 12: Transmit Clock and Receive Clock Timing Parameters — FPGA Interface

Symbol	Transmitter	Receiver	Units
T_{SET}	960	480	picoseconds
T_{Hold}	960	480	picoseconds

Figure 12: Setup and Hold Timing — FPGA Interface



External to the Tsi620, the transmit and receive clocks are center-aligned with respect to data and control. The receive interface samples data when it sees a clock edge, either positive or negative. The data sampled is buffered to allow for phase misalignment between the link partner and the FPGA Interface, and then read out synchronously to the internal clock.

4.2.3 Port Reset

The FPGA Interface and the SerDes Interface can be reset in the same method as the SMAC using the SOFT_RST_X4 bit in the "RapidIO SMAC x Digital Loopback and Clock Selection Register". When this bit is set, the following sequence occurs:

1. No further packets are accepted from the link partner.
2. On the Switch ISF side, transmitted packets are completed, received packets in flight are completed, and no more packets are accepted. On the RapidIO (FPGA) side, transmitted packets are completed, received packets in flight are completed, and no more packets are accepted.

3. The Switch ISF port is shut down.
4. The SP6_PHY_DISABLE signal is asserted to let the link partner know that link initialization has started.
5. The FPGA port is brought back into service as it would be after a Switch Reset.

4.2.4 Port Power Down

The FPGA Interface and the SerDes Interface can be powered down in the same method as the SMAC using the SP6_PWRDN pin (see "Port Power Down").

The FPGA Interface can also be powered down by setting the PWDN_X4 bit in the "RapidIO SMAC x Digital Loopback and Clock Selection Register". When the FPGA Interface is powered down, the SP6_TXCLK, SP6_TXD[31:0], and SP6_TXCTL[3:0] signals are not driven. The SP6_PHY_DISABLE signal is asserted.

4.2.5 Port Mode Selection

The FPGA Interface can operate in either 4x mode or 1x mode. This can be configured using the SP6_MODE_SEL pin, similar to the SMAC operation (see "Port Aggregation: 1x and 4x Link Modes").

Note that unlike SMAC, the FPGA Interface does not support a 1x + 1x configuration. The SP6_MODE_SEL pin, therefore, just controls whether or not the FPGA Interface operates as a 4x port or a 1x port.

When the FPGA Interface operates in 1x mode, data signals 0 to 7 and control signal 0 are driven with valid data on the transmit interface. Data signals 0 to 7 and control signal 0 are sampled on the receive interface.

4.2.6 Lane Swapping

The FPGA Interface does not support lane swapping.

4.2.7 Loopback Testing

Loopback testing of the FPGA SerDes Interface operates identically to that of the other SerDes Interface on Tsi620 (see "Port Loopback Testing").

The FPGA Interface supports the following loopback modes:

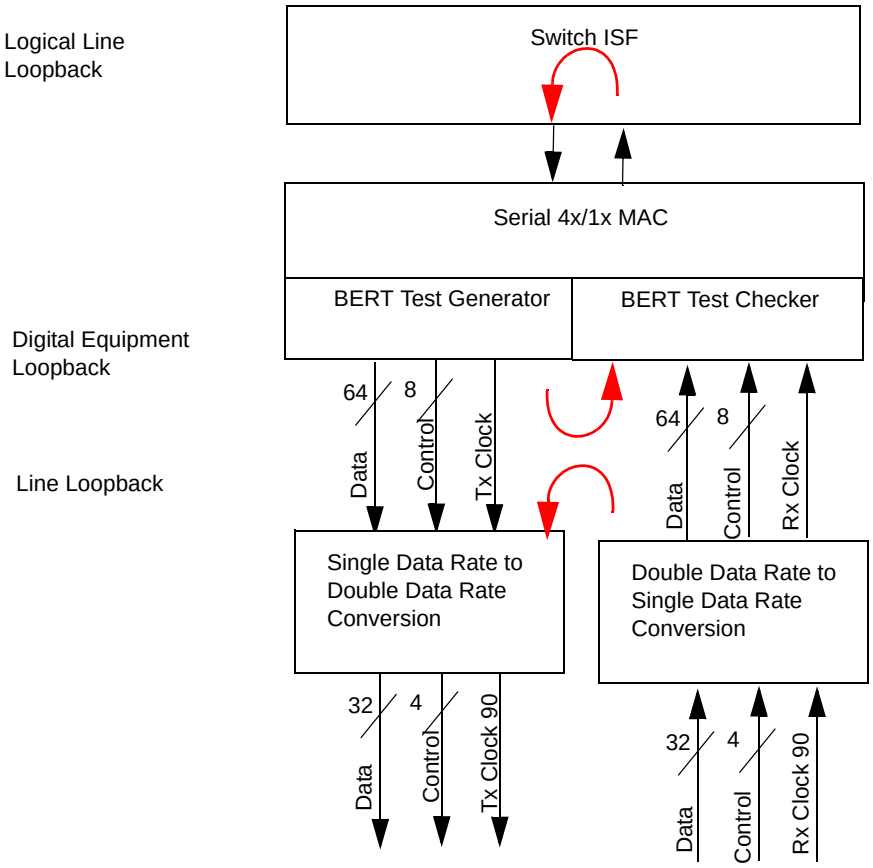
- Line loopback
- Logical line loopback
- Digital equipment loopback

The Internal Switch Port does not support any loopback test modes.

Logical line loopback operates identically for the FPGA Interface or for the SMAC Interface. Other loopback modes differ slightly between the SMAC Interface and the FPGA Interface. The loopback modes operate in a similar manner, but the location of the loopback is different.

Unlike the SMAC Interface, the FPGA Interface loopback modes apply to all lanes of the FPGA Interface. It is not possible to enable loopbacks for just one lane of the FPGA Interface.

Figure 13: FPGA Loopback Locations



4.2.7.1 Line Loopback

When line loopback is enabled, the receive data, control and clock signals entering the device are looped back out through the transmitter (see Figure 13). Data received is still passed to the FPGA Interface. Data transmitted by the FPGA Interface is not compliant to the RapidIO protocol.

To prevent data from passing to the remainder of the Tsi620 when the FPGA Interface is in line loopback, set the PORT_LOCKOUT bit in the "RapidIO Serial Port x Control CSR".

Line loopback is normally used to perform bit error rate testing (BERT). An external device can be configured to send a specific pattern that the FPGA Interface loops back. The external device can then check the bit stream for errors (see "FPGA Interface Bit Error Rate Testing (BERT)").

When RapidIO traffic must be looped back, Logical Line Loopback should be used to loop the traffic back to the same port.

Note Line Loopback can be performed on a per-lane basis, as well as on all lanes at once (see LINE_LB and SWAP_TX in the "RapidIO SMAC x Digital Loopback and Clock Selection Register").

4.2.7.2 Digital Equipment Loopback

Digital equipment loopback is enabled on a per-port basis through the "RapidIO SMAC x Digital Loopback and Clock Selection Register". When this form of loopback is enabled, the FPGA Interface transmit logic is connected to the receive logic just before where the single data rate to double data rate conversion and transmission would occur. Digital equipment loopback requires the use of packets and a correctly configured lookup table. The Bit Error Rate Tester patterns cannot be used when in Digital Equipment Loopback mode.

All incoming data for the FPGA Interface on its external link is ignored when digital equipment loopback is enabled. Note that the FPGA SerDes clock source must be operational for Digital Equipment Loopback to operate correctly.

4.3 FPGA Interface Bit Error Rate Testing (BERT)

The BERT interface for the FPGA Interface is different than that for the SMAC. The SMAC Interface uses bit error rate test support within the SerDes in a manner identical to that implemented for the IDT Tsi57x family of switches (see "Bit Error Rate Testing (BERT)"). Since the FPGA Interface does not have a SerDes, the implementation of BERT is different. However, just like SMAC, the FPGA Interface's BERT is based either on fixed symbols or on a pseudo-random bit sequence (PRBS).

BERT is enabled on a per-lane basis, and normal traffic flow on the bit lane ceases when BERT is enabled. To enable BERT, program the "RapidIO SMAC 6 PRBS Control Register" to select either normal operation, PRBS-based BERT, or fixed-pattern-based BERT.

BERT should be performed only when the FPGA link partner is in line loopback.

When testing a link on the Tsi620 with the BERT feature, the link partner must support PRBS testing with the polynomial $x^7 - 1$, or it must support fixed-pattern tests. Alternatively, the link partner must support some form of loopback to the Tsi620. Consult the appropriate documentation for other devices to determine if they support these features, and to determine how to configure them.

The PRBS test sequence may be unsuitable for testing an AC coupled system. The PRBS may introduce baseline wander and cause an unrealistically high bit error rate.

4.3.1 Fixed Pattern-based BERT

Fixed pattern-based BERT uses data in software-configurable registers to send an alternating pattern of two 9-bit code groups. The most significant bit is sent on the control line, the least significant 8 bits are sent on the data lines associated with the lane. Fixed pattern-based BERT does not produce an automatic pass/fail result.

Fixed patterns are programmed in the "RapidIO SMAC 6 BERT Data Register for Channel 0 Register" through "RapidIO SMAC 6 BERT Data Register for Channel 3 Register" registers.

The following three patterns are useful for BERT testing:

- 0b0_00000000 and 0b1_11111111 create a high-frequency pattern
- 0b0_00000000 and 0b0_00000000 drive continuous 0s
- 0b1_11111111 and 0b1_11111111 drive continuous 1s

4.3.1.1 Transmitter Configuration

To configure a Tsi620 transmitter for fixed-pattern BERT operation:

1. Write the two values to be transmitted into the "RapidIO SMAC 6 BERT Data Register for Channel 0 Register"-3 register. Each 8-bit lane has its own DATA register, which contains two 9-bit patterns that are alternately and repeatedly transmitted on the lane.
2. Set the PATTERN_SEL field in the "RapidIO SMAC 6 PRBS Control Register" to 0b100 for the desired bit lane(s). Setting this field causes the software-defined pattern to transmit.

4.3.1.2 Receiver Configuration

The Tsi620 does not automatically generate pass or fail results for fixed-pattern BERT. Because of this, the Tsi620 receiver is usually not a part of the BERT. To validate that the received fixed-pattern is correct a high-speed logic analyzer should be used.

4.3.2 PRBS BERT

PRBS BERT uses the polynomial $x^7 - 1$ to generate 9-bit code groups. PRBS BERT produces an automatic pass or fail result.

The following sub-sections discuss how to configure the Tsi620's FPGA receiver and transmitter for PRBS BERT testing. The capability of the link partner devices impacts the type of BERT testing that can be performed in a system.

4.3.2.1 Transmitter Configuration

To configure the FPGA Interface's transmitter for PRBS operation, the SerDes must be powered up and out of reset to generate correct clocking to the FPGA Interface. The following actions must be performed:

1. Write the "RapidIO SMAC 6 PRBS Control Register" register, setting the PATTERN_SEL field and the START_PRBS field for the desired bit lane(s) to 1. The transmitter immediately begins sending the PRBS.

To stop the test:

1. Write the "RapidIO SMAC 6 PRBS Control Register", setting the PATTERN_SEL field and the START_PRBS field to 0 for the desired bit lane(s).

4.3.2.2 Receiver Configuration

Once a transmitter has started sending a PRBS pattern, the receiver can be configured to look for errors in the pattern. To configure a Tsi620 receiver to validate an incoming PRBS pattern:

1. Write the "RapidIO SMAC 6 PRBS Control Register", setting PATTERN_SEL to 1, SYNC_PRBS to 1, and CLR_PRBS_CNT to 1 for the desired bit lane(s).
2. Set STOP_PRBS_CNT in the "RapidIO SMAC 6 PRBS Control Register" to 0 for the desired bit lane(s).

After a fixed amount of time, stop the BERT test:

1. Write the "RapidIO SMAC 6 PRBS Control Register", setting PATTERN_SEL to 0 and setting STOP_PRBS_CNT to 1 for the desired bit lane(s). This stops the receiver from counting errors.
2. Tell the transmitter to stop sending PRBS pattern. If the transmitter is a Tsi620, see "Transmitter Configuration" for instructions.

For instructions on interpreting the results of the test, see the following section.

4.3.2.3 Interpreting the Test Results

Once the receiver has stopped a PRBS BERT test, complete the following actions:

1. Read the "RapidIO SMAC 6 PRBS Channel 0 Counter 0 Register" and "RapidIO SMAC 6 PRBS Channel 0 Counter 1 Register" registers for the bit lane under test.

System software must combine the fields of these two registers into a 48-bit code group counter and a 16-bit error counter

The PRBS BERT test must be stopped before reading the "RapidIO SMAC 6 PRBS Channel 0 Counter 0 Register" or "RapidIO SMAC 6 PRBS Channel 0 Counter 1 Register" registers. Reading these registers during a PRBS BERT test gives undefined results.

2. Once the registers are read, the bit error rate can be computed using the ratio of errors to total received code groups.

   The following results can be encountered once the registers are read:

- If the error counter is full (all 1s), the bit error rate of the line is very high and the test should be run for a shorter duration if the exact error rate must be computed.
- If the error counter is non-zero and the code group counter reaches its maximum value, a reliable bit error rate cannot be computed. The test should be re-run with a shorter duration.

5. Switch ISF

This chapter describes the main features and functions of the Tsi620 Switch's Internal Switching Fabric, more commonly referred to as Switch ISF. Topics discussed include the following:

5.1 Overview

The Switch ISF is the crossbar switching matrix at the core of the Tsi620 Switch. It transfers packets from RapidIO ingress ports to egress ports, and prioritizes traffic based on the RapidIO priority associated with a packet and port congestion.

The Switch ISF has the following features:

- Full-duplex, non-blocking, crossbar-based switch fabric
- 10 Gbps fabric ports allow up to 10x internal speedup
- Manages head-of-line blocking on each port
- Buffers hold eight packets per RapidIO ingress port
- Buffers hold eight packets per RapidIO egress port
- Cut-through and store-and-forward switching of variable-length packets

5.2 Port Numbering

The Switch ISF ports are numbered as in the following table.

Table 13: Switch ISF Port Numbering

Switch ISF Port Number	Block	Notes
Port 0	4x/1x port, MAC 0	-
Port 1	1x port, MAC 0	-
Port 2	4x/1x port, MAC 1	-
Port 3	1x port, MAC 1	-
Port 4	4x/1x port, MAC 2	-

Table 13: Switch ISF Port Numbering (Continued)

Switch ISF Port Number	Block	Notes
Port 5	1x port, MAC 2	-
Port 6	4x/1x port, MAC 3	The FPGA Interface. Note that this includes SerDes registers to control the clock generation for this block.
Port 7	1x port, MAC 3	This port is always powered down. Note that Port 6 PRBS registers live in this address space, and must be accessible when this port is powered down.
Port 8	Internal Switch Port	This port mimics the operation of a RapidIO link to/from the SREP.

5.3 Functional Behavior

The Switch ISF transfers packets from an ingress port to an egress port, and to and from the Multicast Engine. When RapidIO packets arrive at the ingress ports, the Tsi620 performs several tests to ensure the packet is valid. If a packet passes these tests, the ingress port consults its Destination ID Lookup Table to determine the egress port for the packet. The Switch ISF transfers entire packets without interruption (while in store-and-forward mode; see "Transfer Modes").

For more information on how RapidIO packets are tested as valid, see "Data Integrity Checking".

The Switch ISF is a crossbar switch, which means that an ingress port can only send one packet at a time to the Switch ISF, and an egress port can only receive one packet at a time from the Switch ISF. However, the Switch ISF can simultaneously transfer packets from multiple disjoint ingress port, egress port pairs. This architecture has no shared memory area that holds packets.

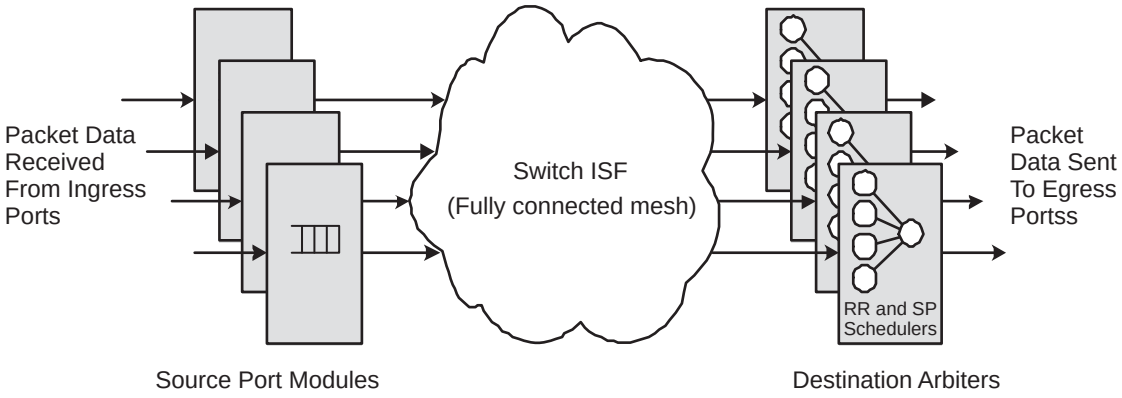
Since many ingress ports can attempt to send a packet to the same egress port, queuing is required at the ingress ports. Special arbitration algorithms at both the ingress and egress sides of the Switch ISF ensure that head-of-line blocking is avoided in these queues.

Queuing is also required at the egress ports. Packets can accumulate when an egress port has to re-transmit a packet (for example, due to a CRC error), or when a high-bandwidth ingress port sends traffic to a lower-bandwidth egress port.

Queuing is also required to support multicast functionality. The Switch ISF supports dedicated connections between each ingress port and the multicast work queue, and a dedicated connection between the work queue and the broadcast buffers. This allows packets to be replicated in parallel. For more information on multicasting, see "Multicast Engine".

Figure 14 shows a conceptual block diagram of the relationship of the components within the Switch ISF.

Figure 14: Switch ISF



5.3.1 Transfer Modes

The Switch ISF supports both *cut-through* and *store-and-forward* transfer modes. These modes are selectable on a per-port basis. By default, all ports are configured for store-and-forward mode. To change the configuration, write the TRANS_MODE bit in the "RapidIO Port x Control Independent Register" when traffic is not flowing through the port.

5.3.1.1 Store and Forward Mode

When a port is configured for store-and-forward mode, it must receive the entire packet before the Switch ISF transfers the packet to an egress port. This increases the latency of all packets received on the port. The increase in latency is directly proportional to the packet size and bit rate of the port.

5.3.1.2 Cut-through Mode

When a port is configured for cut-through mode, it starts sending the incoming packet before the packet has fully arrived at the Tsi620 Switch. This is possible because the RapidIO destination identifier (routing information) appears near the front of a RapidIO packet.

Congestion

If a port is configured for cut-through mode, it does not guarantee that the packet is sent to the Switch ISF immediately after the destination identifier arrives for the packet. Congestion in the Switch ISF can mean that some or all of the packet is received before the switching operation begins.

Cut-through mode normally provides better system performance. However, in cases where there is a mix of high-speed and low-speed ports, a packet sent from a low-speed port to a high-speed port in cut-through mode prevents the high-speed port from maximizing its output bandwidth. If other ports are also sending to the same destination, the high-speed ingress ports could suffer a drop in throughput.

Congestion Example

In this congestion example the following parameters are true:

- Port 0 is currently sending Packet #1 to port 2
- Packet #2, also destined for port 2, starts to arrive on port 1

Packet #2 must wait for the Packet #1 to finish before it has access to port 2. Some or all of Packet #2 must be buffered.

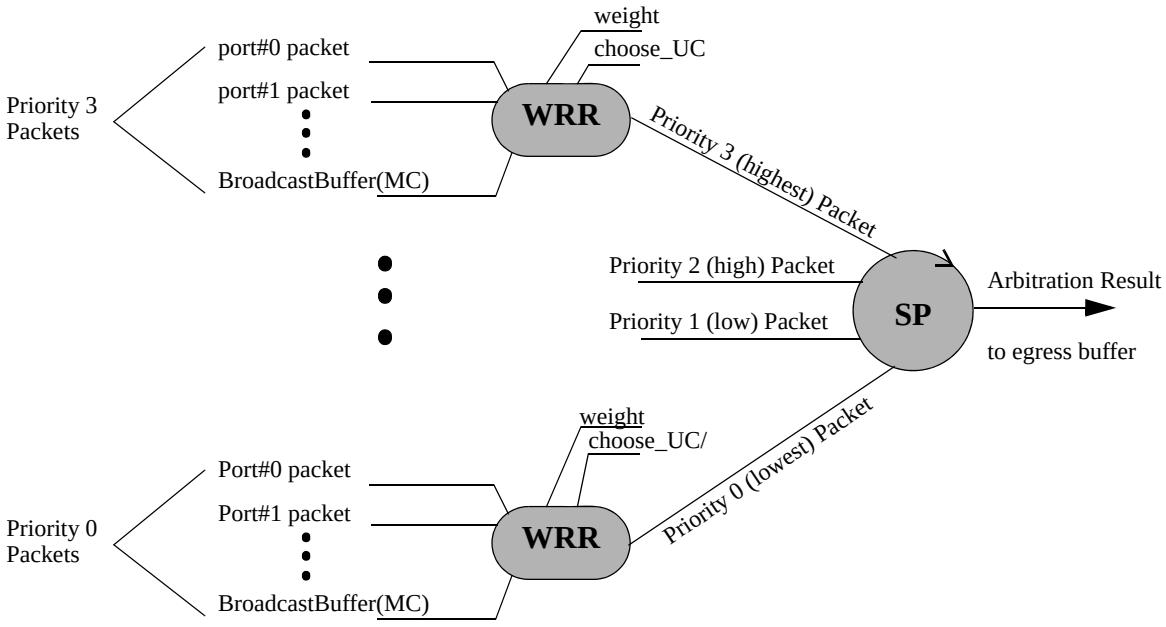
5.4 Arbitration for Egress Port

When multiple ingress ports need to send a packet to the same egress port at the same time, the egress port must make an arbitration decision about which packet to accept.

An output arbiter exists for each egress port. The output arbiters work in conjunction with the input arbiters to maximize throughput and to avoid head-of-line (HOL) blocking. When the Multicast Engine is used, the output arbiter allows system designers to control the maximum number of sequential multicast or non-multicast (unicast) packets that are accepted for a given priority.

The egress port arbiter complies with the RapidIO buffer control rules. It allows the buffer controls to be configured to improve throughput. Two arbitration schemes handle traffic from the ingress and multicast ports, namely Strict Priority and Weighted Round Robin (see Figure 15). Only HOL packets at the ingress queues or broadcast buffers are considered for arbitration.

Figure 15: Egress Arbitration: Weighted Round Robin and Strict Priority



5.4.1 Strict Priority Arbitration

The Switch ISF considers packet priority with a strict priority (SP) service algorithm for egress arbitration.The egress arbiters ensure that all traffic with RapidIO priority *N* is sent before any traffic with RapidIO priority *N-1*.

For more information on packet arbitration, see the *RapidIO Interconnect Specification (Revision 1.3)*.

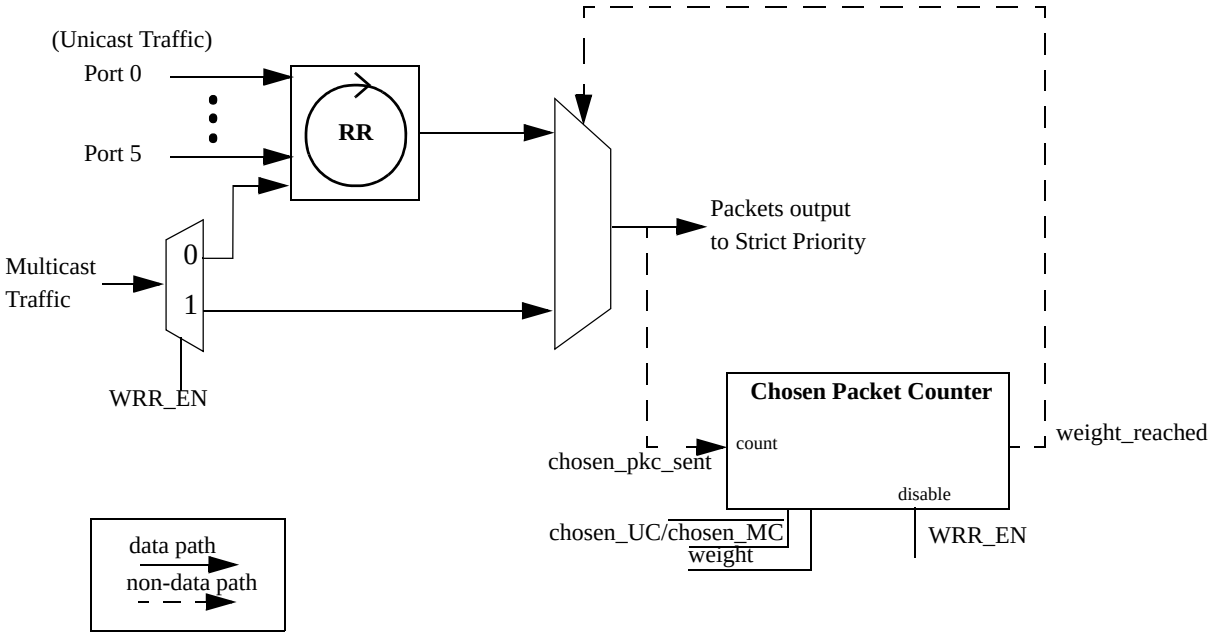
The Strict Priority arbiter gives preference to the highest priority packets among the egress ports. As long as priority 3 packets are being presented for arbitration by any port, those packets are accepted ahead of any priority 2 packets. Similar behavior holds for priority 2 packets being chosen over priority 1; and priority 1 over priority 0 packets.

Priority 3 packets from a specific port are transmitted when the port has its turn. However, Priority 2 or lower priority packets cannot be sent to an egress port when the number of free buffer associated with that specific port is equal to or smaller than the watermark for that specific priority (see "Egress Watermark").

5.4.2 Weighted Round-Robin (WRR) Arbitration

Within the same priority group, the Weighted Round-Robin (WRR) arbiter in each egress port decides which ingress port to receive packets from. This WRR arbiter is a modified round-robin arbiter with the option to assign different weights on unicast or multicast traffic. There are a total of four WRR arbiters, one for each priority, per egress port.

Figure 16: Weighted Round Robin Arbiter per Priority Group



The conceptual diagram of the WRR arbiter is shown in Figure 16. The same arbiter exists for each Priority Group. Depending on the setting of WRR_EN in the "RapidIO Port x Prefer Unicast and Multicast Packet Priority 0 Register", the multicast traffic can participate in the round-robin arbiter. The WRR arbiter consists of a simple round-robin arbiter that services its input one after the other in a sequential manner, starting at Port 0 upon reset. The Chosen Packet Counter is used only when weighted operation between multicast and unicast is desired. Otherwise, the RR arbiter outputs become the arbitration result.

If the system has no preferred traffic, WRR_EN is de-asserted, and the multicast packets are handled as unicast packets. Packets from each port including the multicast ones, if available, are allowed to proceed in order, one after the other, to the Strict Priority arbitration. The averaged probability of multicast packets being serviced, with equal traffic load among multicast and unicast ports, is 1 out of 9 when operating in 8 ports (~11.1%); the same as a unicast port.

When weighted operation is desired, WRR_EN is asserted (WRR_EN=1); the user has the option to set the type of preferred traffic (CHOOSE_UC in "RapidIO Port x Prefer Unicast and Multicast Packet Priority 0 Register"; Port x/Priority y) and the minimum number of packets allocated for the chosen traffic on the egress port (WEIGHT in "RapidIO Port x Prefer Unicast and Multicast Packet Priority 0 Register"; Port x/Priority y). These two register values set the parameters of operation for the Chosen Packet Counter inside the WRR arbiter. The CHOOSE_UC value determines which type of traffic is chosen to be favored (0 = Multicast; 1 = Unicast). The WEIGHT value determines the number of the packets of the *chosen type* are to be sent in between non-chosen ones. Each time a chosen packet (either multicast or unicast) is sent, the Chosen Packet Counter is notified. The chosen packets are selected for transmission as long as the WEIGHT value is not reached. Once the WEIGHT value is reached, a non-chosen packet is selected instead and the Chose Packet Counter is reset.

5.4.2.1 Examples of WRR Arbitration

In the case where no chosen packet is available when its opportunity occurs, the WRR arbiter automatically selects the non-chosen packets. Similar behavior applies to non-chosen packets. When the opportunity to transmit non-chosen packets occurs and there is none available, packets of the chosen type are sent. However, this does not consume the original opportunity allocated.

A few examples of register settings for the WRR arbiter is shown in Table 14.

Table 14: Sample Register Settings for WRR in Given Priority Group (WRR_EN = 1)[a]

CHOOSE_UC	WEIGHT	% of Multicast Packets sent to SP arbiter	% of Unicast Packets sent to SP arbiter	Packet Sequence (M = Multicast, U = Unicast)
0	0	0	100	...UUUUUUUU...
0	1	50	50	...UMUMUMUM...
0	15	93.75	6.25	...MUMMMMMMMMMMMMMUM...
1	0	100	0	...MMMMMMMM...

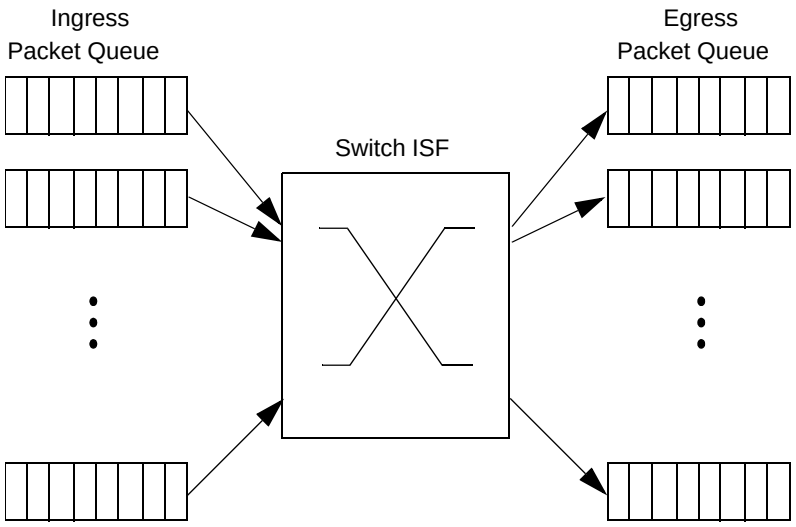
a. The percent values in the table assumes all opportunity for transmission is filled by either the chosen or non-chosen types.

When there is 100% utilization of either unicast or multicast, starvation may be seen in the system.

5.5 Packet Queuing

The Tsi620 has a queuing system on both the ingress and egress ports, as displayed in the following figure.

Figure 17: Ingress/Egress Packet Queues in Tsi620



5.5.1 Output Queuing on the Egress Port

Each egress port has a queue that holds up to eight packets. This buffer is required because packets may need retransmitting. The buffer is also necessary to store the incoming packets when the egress port has a slower baud rate than the Switch ISF. The depth of the buffer queue dictates the Switch ISF flow control. This flow control determines the number of packets of a certain priority an egress port can receive. If the output queue is full, the ingress port is notified and then must begin queuing packets. If the ingress port also runs out of buffers for packets, the RapidIO link level flow control (packet retries) is activated on the ingress RapidIO port.

For more information on packet retries, see *RapidIO Interconnect Specification (Revision 1.3)*.

5.5.1.1 Egress Watermark

The Switch ISF egress arbiter generates flow control for a given priority of traffic based on *watermarks*. Watermarks are defined for priority 0, 1, and 2 packets. No watermark is defined for Priority 3 packets since they are accepted when there are free buffers. The watermarks are programmable through the "RapidIO Port x Switch ISF Watermarks Register".

Rules for Programming Watermarks

The following rules applied when the watermarks are used in Tsi620:

- No watermark is associated with Priority 3 packets
- A Priority x packet is accepted in the buffer if the number of free buffers is greater than the programmed watermark of the associated Priority (PRIOxWM). For example, when PRIO1WM is programmed to 3, a Priority 1 packet is accepted only when there are four or more free buffers.
- The three programmed watermarks (PRIO0WM, PRIO1WM, and PRIO2WM) must contain values where PRIO0WM > PRIO1WM > PRIO2WM > 0 at all times.
- The watermarks must be set according to the following rules:
  - PRIO2WM >= 1
  - PRIO1WM >= 2
  - PRIO0WM >= 3

A deadlock situation occurs in the system if any one of the watermark rules is violated.

This hierarchy of watermarks ensures that packets of lower priority can never consume all buffers and prevents the transfer of higher priority packets. With the correct setting of the watermarks, there is at least one Priority 3 packet in a full buffer. If all buffers are filled then at least one of the buffers must be occupied by a priority 3 packet.

Table 15: Watermark Examples

Packet Buffers Available	Example 1: PRIO2WM = 1 PRIO1WM = 2 PRIO0WM = 3	Example 2: PRIO2WM = 2 PRIO1WM = 4 PRIO0WM = 5
	Packet Priority that can be Accepted	Packet Priority that can be Accepted
8	0, 1, 2, 3	0, 1, 2, 3
7	0, 1, 2, 3	0, 1, 2, 3
6	0, 1, 2, 3	0, 1, 2, 3
5	0, 1, 2, 3	1, 2, 3
4	0, 1, 2, 3	2, 3
3	1, 2, 3	2, 3
2	2, 3	3
1	3	3
0	none	none

Two examples are provided in Table 15. The first example describes the default setting of the three watermarks. This maximizes the number of buffers that can accept lower priority packets, which in turn maximizes the throughput of these priorities. The second example describes a customized setting that favors the Priority 3 and 2 traffic at the expense of the throughput of Priority 1 and 0 packets.

In some systems, it is necessary to guarantee maximum throughput for a burst (continuous sequence) of packets at the same priority. In a congested system, only one buffer may be available for these packets. This can restrict throughput on the egress port, since while one packet in the burst is being transmitted and is awaiting acknowledgment, another packet in the burst cannot be accepted or transmitted. Watermarks can guarantee that two buffers are available for these packets. When two buffers are available, while one packet is transmitted and awaits acknowledgement, another packet can be accepted. This leads to an increase in throughput for packets in the burst.

The packet offered for selection by the egress port is subject to the input queuing arbitration. For information on how the ingress port selects which packet to offer for transmission, see "Input Arbitration".

5.5.1.2 Transmitting Packets from the Egress Port to the Link Partner

Packets in the egress queue are transmitted on the RapidIO link in first-come, first-served (FCFS) order, except during retransmission. Retransmission represents an opportunity for reordering operations (see "Input Arbitration"). When a packet is retried by the link partner, the oldest packet with the highest priority in the egress buffer is selected for transmission.

When a port cannot transmit packets, its egress queue becomes full. Since the queue depth generates flow control across the Switch ISF, a function is required to ensure that the Switch ISF does not suffer performance degradation when a port is unable to retire its packets. Inability to retire packets may be caused by a powered down port or the port being in a error state. This can occur as follows:

- When a port is powered down, it flushes its buffer and continues to accept packets from the Switch ISF. Packets accepted by a powered down port are silently discarded.
- When a port does not enter a normal operating mode with its link partner, this can be detected and the impact to the remainder of the system is limited. For more information on detection and recovery from non-operative links, see "Dead Link Timer".

5.5.2 Input Queue for the Switch ISF Port

Each ingress port has a queue that holds up to eight packets. Buffering is required to deal with any congestion in the Switch ISF. Since the Switch ISF is a crossbar switch, each egress port can receive one packet from the Switch ISF at a time. If multiple ingress ports need to send to the same egress port, all but one of the ingress ports must buffer its packet and try to transfer it at a later time.

5.5.2.1 Ingress Watermarks

Similar to the egress port, the ingress port generates flow control for a specific priority of traffic based on a programmable number of free buffers using watermarks. Watermarks can be programmed for priority 0, 1, and 2 packets. Priority 3 packets are accepted when there are free buffers. The "RapidIO Port x RapidIO Watermarks Register" programs watermarks for the ingress port. The rules for programming the ingress watermarks are the same as for "Egress Watermark".

This hierarchy of watermarks ensures that packets of lower priority cannot consume all buffers and prevent packets of higher priority from passing them. For example, if all buffers are filled, then at least one of the buffers must be occupied by a packet of priority 3. Since priority 3 is the highest priority in the system, the priority 3 packet should be given the first opportunity to make forward progress.

The default watermark values are 1 for priority 2, 2 for priority 1, and 3 for priority 0. This maximizes the number of buffers that can accept lower priority packets, which in turn maximizes the throughput of these packets.

In some systems, it is necessary to guarantee maximum throughput for a burst (continuous sequence) of packets at the same priority. In a congested system, only one buffer may be available for these packets. This can restrict throughput on the egress port since while one packet in the burst is being transmitted and is awaiting acknowledgment, another packet in the burst cannot be accepted or transmitted. Watermarks can guarantee that two buffers are available for these packets. When two buffers are available, while one packet is transmitted and awaits acknowledgement another packet can be accepted. This leads to an increase in throughput for packets in the burst.

If a packet cannot be admitted by the ingress buffer, the packet is dropped and a RETRY control symbol is sent to the link partner. This symbol begins transmission within 12 SYS_CLK cycles of receipt of the first 4 bytes of the packet. This allows the link partner to, at its discretion, select another packet for transmission that has a higher probability of being accepted by the link partner.

The Tsi620 provides registers that system software can use to determine the extent of input congestion on the Tsi620 Switch (see "RapidIO Port x Control Independent Register"). Table 15 shows which priorities of packets can be accepted given the number of free buffers

5.5.3 Input Arbitration

When packets are placed in a single input queue, head-of-line (HOL) blocking can result. HOL blocking occurs when the packet at the head of a queue is blocked, and the packets must remain in the same order. This means that no packet in the queue can be sent across the Switch ISF even if all the packets, except the first packet, have an uncongested path to their respective destinations.

The Switch ISF manages HOL blocking by reordering packets in a manner compliant with the *RapidIO Interconnect Specification (Revision 1.3)*. This technique may allow another packet to proceed if the packet at the head of a queue is blocked, depending on the arbitration mode selected. In other words the packets are reordered in the queue, but this reordering never violates the RapidIO packet ordering rules.

Three modes of ingress arbitration are supported, and can be configured with the IN_ARB_MODE field in the "Switch ISF Control Register":

- First come, first served (default)
- Strict Priority 1
- Strict Priority 2

Each time the Switch ISF reorders a packet within its queues[1], the Tsi620 increments a 16-bit counter field, CTR, in the "RapidIO Port x Reordering Counter Register" on the affected port. This value can be monitored as an indication of the level of switching congestion. The register also contains a threshold field, THRESH. When the counter is incremented and its new value equals the threshold, the Tsi620 raises the maskable INB_RDR interrupt. This interrupt is masked with the "RapidIO Port x Control Independent Register" and is cleared with the "RapidIO Port x Interrupt Status Register".

The number of times a packet is reordered is configurable (see "Reorder Limiting").

5.5.3.1 First Come, First Served Mode

In this mode, packets flow through the ingress queues in order unless reordering is required to manage HOL blocking. The packet closest to the head of the queue that can make progress is selected to make progress regardless of its priority.

Reordering of packets only occurs if the packet at the head of the queue is blocked and there is at least one packet that can make progress. Reordering of packets does not occur if there are no other packets in the buffer.[2]

This input arbitration mode can produce the best throughput when prioritization of traffic is not important.

5.5.3.2 Strict Priority 1 Mode

In this mode, higher priority packets are served ahead of lower priority packets if the higher priority packets are not blocked. Reorder operations only occur when the head of the queue is blocked and there is a packet request with a higher precedence (according to the following rules) which exists in the queue.[2]

The arbiter selects a packet to compete in egress arbitration based on the following rules:

- Select the priority 3 packet that can be accepted by its destination Switch ISF port and is closest to the head of the queue.
- If there are no such packets, select the priority 2 packet that can be accepted by its destination Switch ISF port and is closest to the head of the queue.
- If there are no such packets, select the priority 1 packet that can be accepted by its destination Switch ISF port and is closest to the head of the queue.
- If there are no such packets, select the priority 0 packet that can be accepted by its destination Switch ISF port and is closest to the head of the queue.
- If there are no such packets, select the priority 3 packet closest to the head of the queue. Note that this packet cannot make progress.
- If there are no such packets, select the priority 2 packet closest to the head of the queue. Note that this packet cannot make progress.

1. Note that counting the number of times a packet is reordered within a queue is different from counting the number of times packets are sent out of order. The Switch ISF might reorder the queue several times before finding one packet to send.
2. "Reorder Limiting" affects which packet can be chosen.

- If there are no such packets, select the priority 1 packet closest to the head of the queue. Note that this packet cannot make progress.
- If there are no such packets, select the priority 0 packet closest to the head of the queue. Note that this packet cannot make progress.

5.5.3.3 Strict Priority 2 Mode

In this mode, higher priority packets are served ahead of lower priority packets even when the high priority packets are blocked. This mode sacrifices throughput for the lowest latency on high priority packets. In this mode, reorder operations only occur when the head of the queue is blocked and there is a packet request with a higher precedence (according to the following rules) which exists in the queue.

The arbiter selects a packet to compete in egress arbitration based on the following rules:

- Select the priority 3 packet that can be accepted by its destination Switch ISF port and is closest to the head of the queue.
- If there are no such packets, select the priority 3 packet closest to the head of the queue. Note that this packet cannot make progress. This means all priority 3 packets have to be out of queue before looking at other levels, even if priority 3 packet cannot make progress.
- If there are no such packets, select the priority 2 packet that can be accepted by its destination Switch ISF port and is closest to the head of the queue.
- If there are no such packets, select the priority 2 packet closest to the head of the queue. Note that this packet cannot make progress.
- If there are no such packets, select the priority 1 packet that can be accepted by its destination Switch ISF port and is closest to the head of the queue.
- If there are no such packets, select the priority 1 packet closest to the head of the queue. Note that this packet cannot make progress.
- If there are no such packets, select the priority 0 packet that can be accepted by its destination Switch ISF port and is closest to the head of the queue.
- If there are no such packets, select the priority 0 packet closest to the head of the queue. Note that this packet cannot make progress.

5.5.3.4 Reorder Limiting

When packets leave an input queue in other than first come, first served order, a packet is said to have been *reordered*. Reordering occurs as described in the previous sections on input arbitration algorithms: “First Come, First Served Mode”, “Strict Priority 1 Mode”, and “Strict Priority 2 Mode”.

If a packet is reordered, it is sent earlier than it would otherwise be sent. This causes some packets to be sent later than would otherwise be sent, and others to be sent in the same relative order. For example, if the 5th packet to arrive at an ingress port is sent first, the 1st, 2nd, 3rd, and 4th packets are delayed while the 6th, 7th, and 8th packets are not affected at all.

The Tsi620 never violates the RapidIO protocol when it reorders packets.

Reorder limiting prevents excessive delays of a packet by packets of lower or equal priority. Reorder limiting does not prevent delays of a packet by packets of higher priority. When reorder limiting is enabled, each time a packet X is delayed in the queue because a lower or same priority packet was sent earlier, the Switch ISF decrements the reorder counter associated with packet X. When the packet reordered ahead of packet X has a higher priority than packet X, the reorder counter of packet X is not decremented.

When the reorder counter for packet X reaches 0, no packets of lower or same priority are permitted to be reordered ahead of packet X. When the reorder counter of packet X is 0, packet X must be transmitted ahead of all other packets of lower or equal priority that are positioned after packet X in the queue.

Higher priority packets that appear after packet X in the queue can cause the continued delay of packet X. Higher priority packets can be reordered ahead of packet X, regardless of the value of the packet X reorder counter.

One of the properties of reorder limiting is that when the reorder counter of a packet X of given priority Y reaches 0, the reorder counters of all packets with a priority equal or greater than Y that appear ahead of packet X in the queue must also be 0.

Reorder limiting is disabled by default and can be enabled by setting the RDR_LIMIT_EN bit to 1 in the "Switch ISF Control Register". Enabling this feature is recommended. Note that reorder limiting applies to all ports and all packets in the Tsi620.

The number of times a packet is permitted to be delayed by a lower or same priority packet is configurable through the RDR_LIMIT field in the "Switch ISF Control Register".

Note that reorder limiting can change the packet chosen by First come, first serve and Strict Priority 1 arbitration. For example, assume three packets, X1, X2, and X3, are held in the ingress queue in that order. Packets X1 and X2 have the same priority, and packet X3 has a higher priority. Packet X1 and X3 cannot make progress. Using Strict Priority 1 arbitration without reorder limiting results in packet X2 being reordered to the head of the queue. However, if reorder limiting is used, and packet X1's reorder limit counter has reached 0, then the Strict Priority 1 arbitration algorithm cannot select packet X2. Packet X3 would be chosen in this case.

5.5.3.5 Transaction Error Acknowledge

A Transaction Error Acknowledge (TEA) signal is used in the Switch ISF request queue to control the time a packet can be at head of the request queue. When an ingress packet at the head of the request queue sends a request to the Switch ISF, the TEA timer is started to keep track of the request time. If the timer value reaches a customer programmable threshold due to congestion at the destination port (defined in TEA_OUT in the "Switch ISF Control Register"), the TEA interrupt is asserted (maskable by an enable in "Switch ISF Control Register" and "Switch ISF Interrupt Status Register"). The packet is then removed from the request queue to free up space for ingress traffic and the transaction is deemed incomplete. The TEA error can also be reported back to the host by a port-write.

Port-Write due to TEA can be disabled in two ways. The first is to write a 1 to PW_DIS in the "RapidIO Port x Mode CSR", which also turns off other port writes. The second is to disable the TEA interrupt generation by writing a 0 to TEA_EN in the "Switch ISF Control Register".

In the situation when there is reordering in the request queue, the TEA timer is reset and counting starts with the new head-of-queue packet.

5.5.4 Input Queuing Model for the Multicast Work Queue

The multicast work queue accepts packets from all of the ingress ports. The multicast work queue does not accept packets from the broadcast buffers. The queue accepts packets based on strict priority (see "Multicast Engine"). Any priority N packet is accepted before packets of priority N-1. Within each priority, the multicast work queue uses the round-robin algorithm.

The multicast work queue operates in strict First-In, First-Out (FIFO) order and has no watermarks associated with it. The queue allows packets to cut-through to the broadcast buffer (see "Cut-through Mode").

5.5.4.1 Multicast Work Queue Ingress Flow Control

Unlike the ingress port and egress port queues, the multicast work queue operates as a bounded buffer. Packets can begin at any point within the bounded buffer. For more information on the operation of the multicast work queue, see "Multicast Engine".

The multicast work queue has space for up to eight maximum sized (276 byte) packets, or 245 minimum sized (8 byte) packets. The queue's ingress arbitration operates based on the following rules:

- If there is not sufficient space for a maximum sized (276 byte) packet to be received, all ingress ports are signalled that no packets can be accepted by the multicast work queue.
- If there is sufficient space for at least one maximum sized (276 byte) packet to be received, all ingress ports are signalled that packets of any priority can be accepted by the multicast work queue.

5.5.5 Input Queuing Model for the Broadcast Buffer

The broadcast buffer receives data from only one source, the multicast work queue. The buffer operates in strict First-In, First-Out (FIFO) order. The broadcast buffer does not use watermarks, and it operates in "Store and Forward Mode".

5.5.5.1 Broadcast Buffer Ingress Flow Control

Like the multicast work queue, the broadcast buffer operates as a bounded buffer. Packets can begin at any point within the bounded buffer. For more information on the operation of the multicast work queue, see "Multicast Engine".

The multicast work queue has space for one maximum sized (276 byte) packet, or up to eight smaller packets. Up to eight packets can be accepted provided that their individual sizes, rounded up to the nearest multiple of 8 bytes, sum to less than 280 bytes.

The broadcast buffer ingress arbitration operates based on the following two rules:

- If there is not sufficient space for 8 more bytes of data to be received, the multicast work queue is signalled that no more packet data can be accepted by the broadcast buffer.
- If there is sufficient space for 8 more bytes of data to be received, the multicast work queue is signalled that more packet data can be accepted by the broadcast buffer.

5.5.6 Output Queuing Model for Multicast

Both the multicast work queue and the broadcast buffer operate in FIFO order. No packet reordering is performed.

The multicast work queue allows packets to cut-through to the broadcast buffers. This reduces the latency of multicast operations.

Broadcast buffers operate in store-and-forward mode. This ensures that packet transmission to the egress port is never delayed by packet replication.

5.5.7 Bandwidth

The Switch ISF supports 10 Gbps of bandwidth in both transmit and receive directions. This is sufficient to handle a 4x RapidIO port operation at 3.125 Gbps. Switch ISF packets can be sent back-to-back, without interruption.

Delays due to arbitration occur only for the first packet to be sent after a period when no packets were available for transition. After the first packet is sent, the following packets can be sent back to back.

Bandwidth can be wasted during transfers in cut-through mode. If the ingress port operates at a slower rate than the egress port, the egress port receives idles when the ingress port has not yet received data for transmission. However, during the transfer, the egress port cannot receive information from ports other than the ingress port. Therefore, when transferring data between ports of different bandwidths, it is recommended that the slower port operate in store-and-forward mode. For information how to control a port's transfer mode, see TRANS_MODE in the "RapidIO Port x Control Independent Register".

6. Multicast Engine

Topics discussed include the following:

6.1 Overview

The Tsi620 multicast functionality is compliant to the *RapidIO Version 1.3 Part 11 Multicast Specification*. A RapidIO multicast operation is a single operation sent to more than one target endpoint.

6.1.1 Multicast Operation

In a multicast operation, packets are received at the speed of the ingress ports (up to 10 Gbps) and broadcast at the speed of the egress ports (up to 10 Gbps). This results in a maximum aggregate multicast egress rate of up to 40 Gbps from the Switch ISF. The maximum amount of data that can be transferred from the four egress ports (three SMACs and one FPGA Interface) is 40 Gbps when operating at maximum width and lane speed.

Packets are routed to the multicast engine based on their destID and TT field value. If no match is detected for the destID and TT field in the multicast group table then the lookup tables route the packet. A maximum of eight different destIDs/TT field combinations can be routed to the multicast engine. Each destID/TT set can be multicast to a different set of egress ports. A set of egress ports that packets are multicast to is called a multicast group and is represented by the multicast mask in the group table. A multicast packet is never sent out on the port that it was received on, so that a number of ports can share the same multicast group.

Multicast packets are accepted by egress ports based on priority. If multicast and non-multicast traffic compete for access to an egress port, multicast-specific egress arbitration can optimize the performance of multicast and non-multicast traffic.

6.1.2 Features

The Tsi620 multicast engine supports multicast packet replication in accordance with *RapidIO Specification Version 1.3, Part 11 Multicast*. The multicast engine includes the following features:

- Provides dedicated multicast resources without impacting throughput on the ports.
- Supports eight multicast groups
- Provides sustained multicast output bandwidth of up to 10 Gbps per egress port
- Supports 10 Gbps of instantaneous multicast input bandwidth[1]
- Packets are replicated to each egress port in parallel
- Accepts traffic bursts with different packet sizes
- Arbitration at the egress port to allow management of resource contention between multicast or non-multicast traffic.

System behavior for the multicasting of requests that require responses is not defined in the *RapidIO Interconnect Specification (Revision 1.3) - Part 11 Multicast Specification*.

6.1.3 Terminology

Table 16 contains the terms that describe the multicast functionality.

Table 16: Terminology

Term	Definition
Multicast Group	A multicast group is a set of ports that must all receive a copy of a packet. A system can support multiple multicast groups, each of which is independent of the other (a group can have all, some, or no ports in common with another group). A multicast group is identified by the destID and TT fields of a packet. Note: A packet is never multicast back out of the port that it is received on, regardless of whether or not this port is included in the multicast group.
Multicast Mask	The set of ports in a multicast group.
Multicast Vector	The set of ports in a multicast group that will receive the multicast packets.
Original Packet	A single multicast packet that arrives at an egress port and gets replicated and sent to multiple egress ports according to the multicast mask.
Packet Copy	A copy of an original packet. A packet copy is sent out on the egress ports.
Multicast Traffic	Packets that are sent to the multicast engine from ingress ports, and packets that are received from the multicast engine by egress ports.
Unicast Traffic	All packets that are not sent to or received from the multicast engine.

1. All bandwidths assume the Switch ISF is clocked at 156.25 MHz.

6.2 Multicast Behavior Overview

The multicast operation involves the following logic:

- Multicast Engine
- Multicast Group Table
- Multicast Work Queue
- Broadcast Buffer for each egress port
- Switch ISF ingress arbitration models for the multicast port
- Switch ISF egress arbitration models for each egress port

The operation of the Work Queue and Broadcast Buffer is described in this section. For more information on ingress or egress arbitration algorithms, see “Input Queuing Model for the Multicast Work Queue”, “Input Queuing Model for the Broadcast Buffer”, and “Output Queuing Model for Multicast”.

Multicast packets received by an ingress port are routed to the Multicast Engine (MCE) port based on the destID and TT field of the packet. A packet arriving at the Tsi620 is directed to the MCE by a multicast group table on the packet’s ingress port. The Work Queue in the MCE selects a multicast group for the packet based on the packet’s destID and TT field. If the ingress port for the multicast packet is a part of the multicast group, the port is removed from the Multicast Vector.

Once the Multicast Vector is computed, the Work Queue transfers the original packet to the Broadcast Buffers associated with the ports. Transmission between the Multicast Work Queue and the Broadcast Buffers uses a dedicated Switch ISF path that is separate from those that route unicast traffic. Once the packet copies are completely received by all of the Broadcast Buffers in the Multicast Vector, each Broadcast Buffer arbitrates with its associated egress ports to accept the packet copies. The Broadcast Buffer begins to transfer the packet copy to the egress port buffers when the egress port indicates that it can accept packets.

Packets can cut-through from the ingress port to the Multicast Work Queue, and from the Multicast Work Queue to the Broadcast Buffers. A complete packet copy must be received by a Broadcast Buffer before it attempts to forward the packet copy to the egress port.

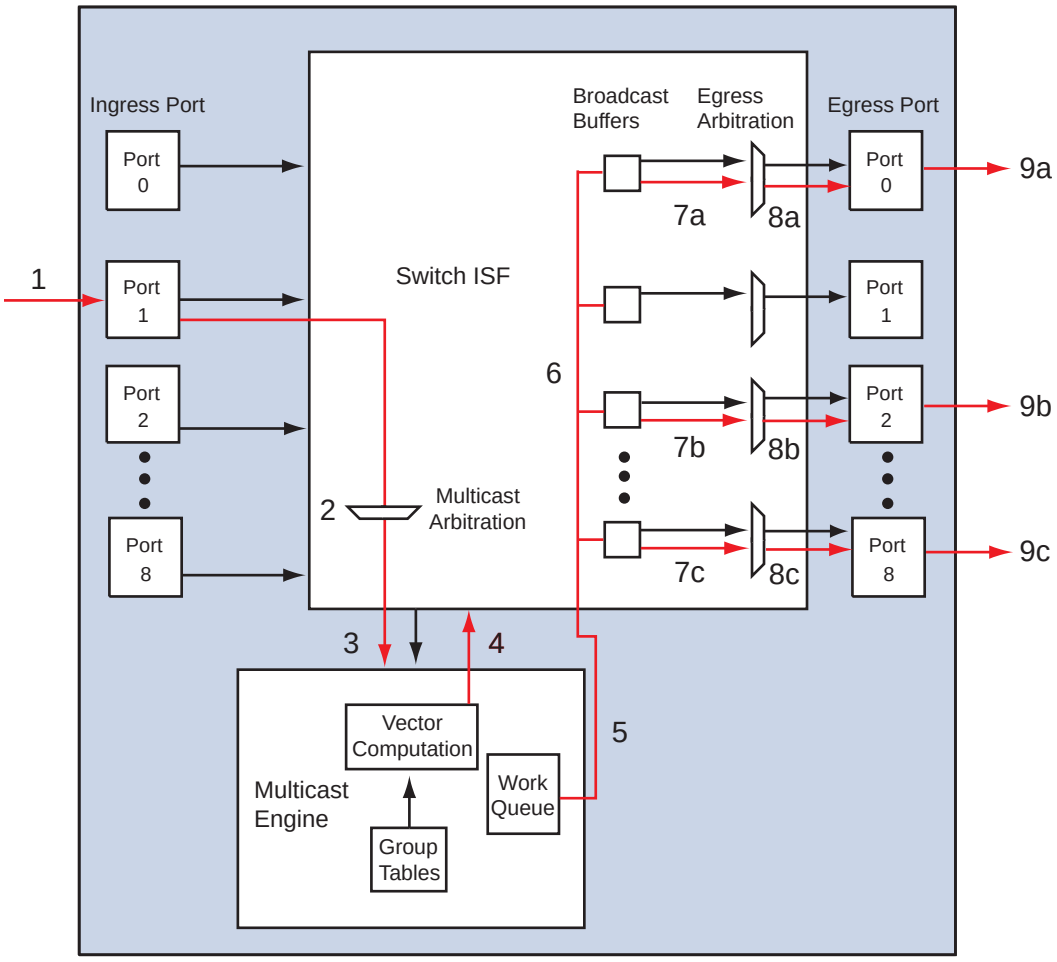
Multicast packets are forwarded to the egress ports as they were received. That is, the source and DestIDs are not altered by the MCE. This means that the replicated packets emerge from the egress port with the same destID as the original packets.

Endpoints that receive multicast and unicast traffic must support multiple DestIDs.

The routing of unicast packets to a system-wide unique device ID is supported in a multicast application because endpoints are expected to support two or more device IDs in order to distinquish unicast from multicast packets.

Figure 18 shows a step-by-step multicast operation through the Tsi620. Numbered descriptions of the multicast events following the diagram.

Figure 18: Multicast Packet Flow in Tsi620



The following list contains a description of events in the previous figure:

1. Port 1 receives a packet and consults the Multicast Group Table. It determines that the packet is a multicast packet by examining the packet's DestID and TT fields. Port 1 requests permission to transfer the packet to the MCE port. The multicast packet is placed in port 1's ingress buffer. Multicast packets residing in the ingress buffer are subject to "Packet TEA" and packet reordering.
2. The packet request competes for access to the MCE and is granted transfer permission by the MCE arbiter.
3. After Port 1 wins arbitration to the MCE it sends the multicast packet through the Switch ISF to the MCE. The transfer occurs at 10 Gbps[1] and starts after the first 8 or 16 bytes of the multicast packets are received in the ingress buffer (cut-through mode).

1. When the Switch ISF clock speed is set to 156.25 MHz.

4. The multicast packet is buffered by the MCE in the Multicast Work Queue. The Multicast Mask is derived from the DestID and TT fields, which indicate the multicast group. Multicast packets residing in the Multicast Work Queue are not subject to reordering and are processed in a FIFO manner. The Multicast Work Queue operates in cut-through mode.
5. By consulting the Multicast Group Table, Ports 0, 1, 2, and 8 are identified as the receiving ports.

If a DestID in the Multicast Group Table is disassociated while the packet with that DestID is still in Multicast Work Queue, the packet is silently dropped.

6. The MCE transmits the Multicast Vector to the Switch ISF. The Switch ISF broadcasts the packets to the egress ports through the Broadcast Buffers according to the Multicast Vector. The replication of the multicast packets can occur at a rate of up to 70 Gbps. The transfer of the packet copies to the selected Broadcast Buffers is a concurrent and parallel operation.
7. The individual Broadcast Buffers must fully buffer the packet copies before presenting the corresponding packet request to the destination arbiter. This is called “store and forward mode”. Packet copies residing in the Broadcast Buffers are subject to the “Multicast Maximum Latency Timer”. However, they are not subject to packet reordering and are processed in a FIFO manner.
8. When the individual Broadcast Buffer is granted arbitration, the replicated multicast packet is transferred to the egress buffer at a rate of up to 10 Gbps. Although represented by the same Multicast Vector, each individual Broadcast Buffer operates independently.
9. Once in the egress buffer, the packet copies are subjected to STOMP (“Multicast Packet Stomping”) and packet reordering. The packet copies at each egress port are transferred from the egress buffer independent of each port.

When a packet is transferred in cut-through mode, it may have an error detected in it (that is, CRC), or the packet may be STOMPed by the RapidIO link partner. In either case, the packet is still accepted and replicated by the Multicast Work Queue and stored in the Broadcast Buffers. The last datum in the packet is marked with a STOMP bit in the Multicast Work Queue, the Broadcast Buffer, and the egress port buffer, and the packet is stomped when it is sent out.

System behavior of multicast which require responses is not defined in the *RapidIO Interconnect Specification (Revision 1.3)*.

6.2.1 Multicast Group Tables

Each ingress port contains a multi-stage lookup table. The Tsi620 compares the incoming packet’s DestID and TT field to the entries in the lookup table to determine the correct egress port. The DestID and TT field in the packet uniquely identify a multicast group. The DestID/TT number space for multicast groups is shared with the number space for unicast destinations.

The first stage of the lookup table contains eight entries used only for multicast packets, and is called the multicast group table. This table resides in both the ingress ports and the MCE. An entry is configured either to match a 16-bit DestID (TT=0) or an 8-bit DestID (TT=1). For example, to match both the 8-bit DestID of five and the 16-bit DestID of five requires two entries in the multicast group table. If the DestID contained in an incoming packet matches any of the eight entries, the ingress port sends the packet to the MCE for replication.

The matching table entry contains a list of ports (multicast mask) to which the MCE sends a copy of the packet. However, a packet copy is never sent out from the port that the original packet was received on, regardless of the contents of the port list.

At the ingress port, if none of the multicast DestIDs in the group table match the packet's DestID, the Tsi620 assumes the packet is a unicast packet and consults the unicast lookup table.

The eight entries in the multicast group table are configured as follows:

1. Add a set of ports to one of eight multicast masks through repeated writes to the "RapidIO Multicast Mask Configuration Register". The Tsi620 silently ignores attempts to configure masks greater than the mask number. The mask number is defined with the MAX_MASKS field of the "RapidIO Switch Multicast Information CAR".

The Tsi620 silently ignores attempts to add or remove non-existent port numbers to and from multicast masks. A non-existent port number is a port number greater than that which exists on the device. For the Tsi620, port numbers greater than 8 are ignored. It is possible to add and remove powered down or disabled ports to and from the multicast masks.

2. Write the DestID that identifies the multicast group, and the multicast mask number from the previous step to the "RapidIO Multicast DestID Configuration Register".
3. Write the "RapidIO Multicast DestID Association Register", setting the LARGE field to indicate whether the DestID is an 8-bit or a 16-bit ID; and setting the CMD field to 11. This associates the DestID to the list of ports that must receive copies of the packet. Note that there must be a one-to-one association between DestIDs and multicast masks.

In alignment with the RapidIO multicast specification, if multiple DestID association operations occur for a multicast mask, the last association operation executed determines which DestID and TT value is associated with a multicast mask.

Ports can be removed from a multicast mask by writing the "RapidIO Multicast Mask Configuration Register", even when the mask is associated with a DestID. If a port that is powered down or detected as faulty is a part of a multicast mask, packets are still replicated and sent to that port. However, the port silently drops the packets.

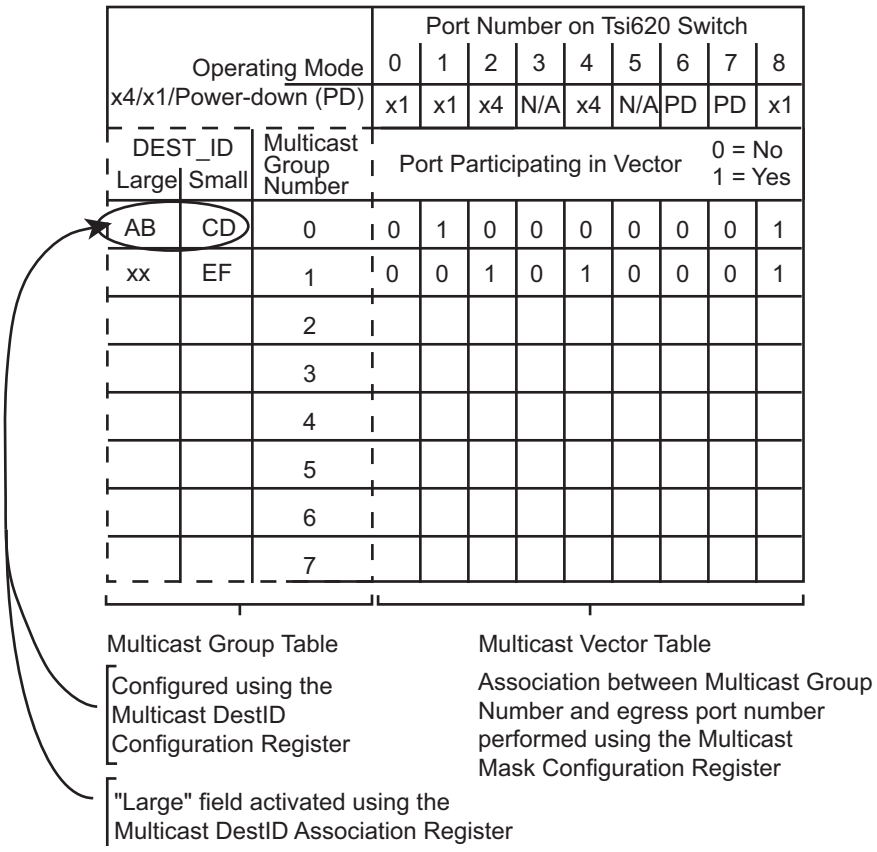
Multicast groups can be deleted by breaking the association between DestIDs and multicast mask numbers through the "RapidIO Multicast DestID Configuration Register" and "RapidIO Multicast DestID Association Register".

To execute either of the previous two operations (port removal or group deletion), the system software must remember what port is associated with which multicast masks, and to which multicast mask number the DestID is bound. If the software designer selects not to maintain a state table, it is possible to determine what multicast mask a DestID/TT value is associated with through the use of Write-to-Verify commands.

It is possible that a multicast mask has no ports selected, or the only port selected is the ingress port the original packet was received on. In this case, the MCE silently discards the packet. No interrupt is raised and only the register flag MC_PKT_DISCARD is set.

Figure 19 is a representation of the relationship between the DestID, multicast group number, multicast vector, and egress port.

Figure 19: Multicast Relationship Representation



6.2.1.1 Configuring Basic Associations

It is necessary to associate a DestID with each multicast mask. The following example assumes that neither block association nor per-ingress-port association is supported by the Tsi620 Switch.

Following upon the previous example, assume the following additional system requirements:

- The 16-bit DestID, 0x1234, must be associated with multicast mask 0.
- The 8-bit DestID, 0x44, must be associated with multicast mask 1.
- The 16-bit DestID, 0xFEED, must be associated with multicast mask 2.

To view what the completed tables contain at the end of configuration, see Figure 20.

In order to create the multicast mask associations, the following register accesses are required (the individual association operations can be performed in any order):

1. Set up the operation to associate DestID 0x1234 with multicast mask 0.

   Write the value 0x1234_0000 to the "RapidIO Multicast DestID Configuration Register".

2. Associate DestID 0x1234 with multicast mask 0.

   Write the value 0x0000_00E0 to the "RapidIO Multicast DestID Association Register".

3. Set up the operation to associate DestID 0x44 with multicast mask 1.

   Write the value 0x0044_0001 to the "RapidIO Multicast DestID Configuration Register".

4. Associate DestID 0x44 with multicast mask 1.

   Write the value 0x0000_00E0 to the "RapidIO Multicast DestID Association Register".

5. Set up the operation to associate DestID 0xFEED with multicast mask 2.

   Write the value 0xFEED_0002 to the "RapidIO Multicast DestID Configuration Register".

6. Associate DestID 0xFEED with multicast mask 2.

   Write the value 0x0000_00E0 to the "RapidIO Multicast DestID Association Register".

6.2.1.2 Configuring Multicast Masks

This section discusses assigning an egress port list to a multicast mask.

Clearing Multicast Masks

In this example, the state of the multicast masks is unknown, and therefore the masks must be cleared before they can be configured. To clear the masks the following register accesses are made:

The accesses to the "RapidIO Multicast Mask Configuration Register" can be performed in any order.

1. Remove all egress ports from multicast mask 0.

   Write the value 0x0000_0040 to the RapidIO Multicast Mask Configuration Register.

2. Remove all ports from multicast mask 1.

   Write the value 0x0001_0040 to the RapidIO Multicast Mask Configuration Register.

3. Remove all ports from multicast mask 2.

   Write the value 0x0002_0040 to the RapidIO Multicast Mask Configuration Register.

Assigning Ports to Multicast Masks

To configure mask 0 to multicast to ports 6 and 7, mask 1 to multicast to ports 3, 4, and 5, and mask 2 to multicast to every port, requires the following series of register accesses:

The accesses to the "RapidIO Multicast Mask Configuration Register" can be performed in any order.

1. Add port 6 to multicast mask 0.

   Write the value 0x0000_0610 to the RapidIO Multicast Mask Configuration Register.

2. Add port 7 to multicast mask 0.

   Write the value 0x0000_0710 to the RapidIO Multicast Mask Configuration Register.

3. Add port 3 to multicast mask 1.

   Write the value 0x0001_0310 to the RapidIO Multicast Mask Configuration Register.

4. Add port 4 to multicast mask 1.

   Write the value 0x0001_0410 to the RapidIO Multicast Mask Configuration Register.

5. Add port 5 to multicast mask 1.

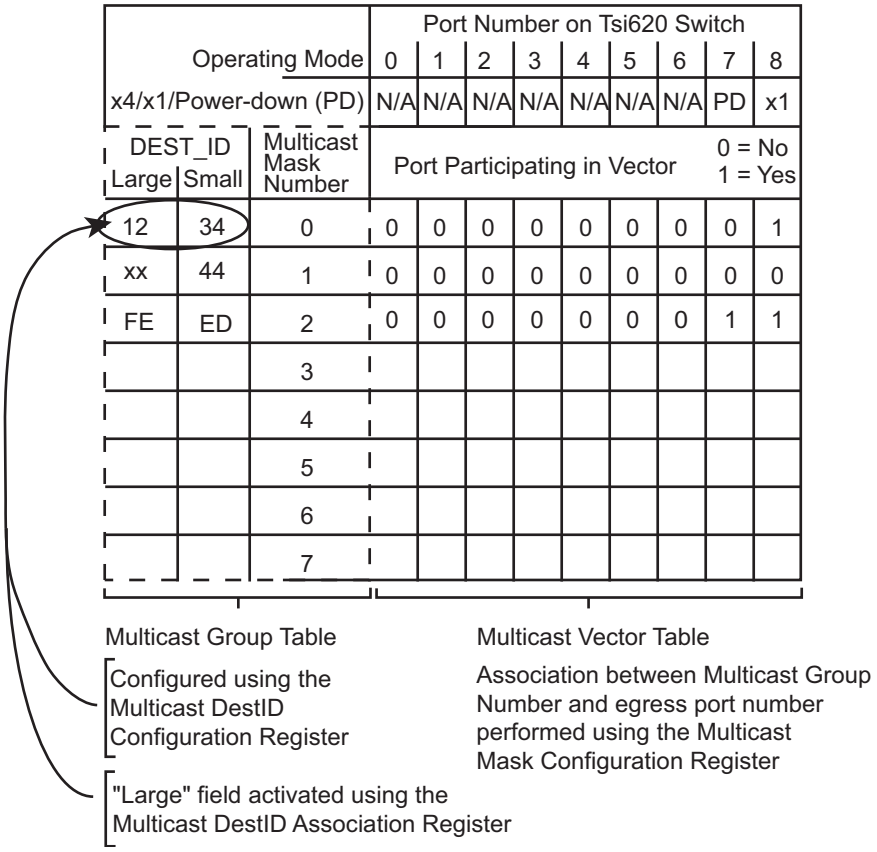
   Write the value 0x0001_0510 to the RapidIO Multicast Mask Configuration Register.

6. Add all ports to multicast mask 2.

   Write the value 0x0002_0050 to the RapidIO Multicast Mask Configuration Register.

Figure 20 shows the completed configuration.

Figure 20: Completed Tables at the End of Multicast Mask Configuration



Removing a Port from a Multicast Mask

In this example, the device attached to port 4 must be removed from the system. The following register accesses modify multicast masks one and two to stop port 4 from being a multicast destination:

The accesses to the "RapidIO Multicast Mask Configuration Register" can be performed in any order.

1. Remove port 4 from multicast mask 1.

   Write the value 0x0001_0420 to the RapidIO Multicast Mask Configuration Register.

2. Remove port 4 from multicast mask 2.

   Write the value 0x0002_0420 to the RapidIO Multicast Mask Configuration Register.

Querying a Multicast Mask

In this example, a system designer needs to determine which port is included in multicast mask 2. The following accesses are to be performed to provide this information:

In each case, the write operation setting up the *write to verify* operation must be performed before the subsequent read to check the Port Present bit status. The individual multicast masks can be queried in any order.

1. Verify that port 0 is included in mask 2.

   Write the value 0x0002_0000 to the RapidIO Multicast Mask Configuration Register.

   Read the value 0x0002_0001 from the RapidIO Multicast Mask Configuration Register.

2. Verify that port 1 is included in mask 2.

   Write the value 0x0002_0100 to the RapidIO Multicast Mask Configuration Register.

   Read the value 0x0002_0101 from the RapidIO Multicast Mask Configuration Register.

3. Verify that port 2 is included in mask 2.

   Write the value 0x0002_0200 to the RapidIO Multicast Mask Configuration Register.

   Read the value 0x0002_0201 from the RapidIO Multicast Mask Configuration Register.

4. Verify that port 3 is included in mask 2.

   Write the value 0x0002_0300 to the RapidIO Multicast Mask Configuration Register.

   Read the value 0x0002_0301 from the RapidIO Multicast Mask Configuration Register.

5. Verify that port 4 is not included in mask 2.

   Write the value 0x0002_0400 to the RapidIO Multicast Mask Configuration Register.

   Read the value 0x0002_0400 from the RapidIO Multicast Mask Configuration Register.

6. Verify that port 5 is included in mask 2.

   Write the value 0x0002_0500 to the RapidIO Multicast Mask Configuration Register.

   Read the value 0x0002_0501 from the RapidIO Multicast Mask Configuration Register.

7. Verify that port 6 is included in mask 2.

   Write the value 0x0002_0600 to the RapidIO Multicast Mask Configuration Register.

   Read the value 0x0002_0601 from the RapidIO Multicast Mask Configuration Register.

8. Verify that port 7 is included in mask 2.

   Write the value 0x0002_0700 to the RapidIO Multicast Mask Configuration Register.

   Read the value 0x0002_0701 from the RapidIO Multicast Mask Configuration Register.

Removing a DestID to Multicast Mask Association

In this example, assume packets from DestID 0xFF02 on port 4 should no longer be allowed to multicast to all nodes (multicast mask 2). To remove DestID 0xFF02 from being associated with multicast mask two on port 4, the following register accesses need to be performed in order:

1. Set up the operation to remove the association between DestID 0xFF02 and multicast mask two

   Write the value 0xFF02_0002 to the "RapidIO Multicast DestID Configuration Register".

2. Remove the association between DestID 0xFF02 and multicast mask 2 on ingress port 4.

   Write the value 0x0000_04C0 to the "RapidIO Multicast DestID Association Register".

Querying an Association

There are multiple ways to query the DestID to multicast mask associations in the Tsi620.

Example One

In this example, it is assumed that a system designer needs to know which multicast masks are associated with DestID 0xFF01 on port four.

Since a read of the RIO Multicast DestID Configuration Register causes the last command written to be executed, that register is only written at the beginning of the sequence.

The individual associations can be queried in any order.

1. Set up the associate operations for DestID 0xFF01 and multicast mask 0.

   Write the value 0xFF01_0000 to the "RapidIO Multicast DestID Configuration Register".

2. Verify that DestID 0xFF01 is not associated with multicast mask 0 for port 4.

   Write the value 0x0000_0480 to the "RapidIO Multicast DestID Association Register".

   Read the value 0x0000_0480 from the RapidIO Multicast DestID Configuration Register.

3. Set up the associate operations for DestID 0xFF01 and multicast mask 1.

   Write the value 0xFF01_0001 to the RapidIO Multicast DestID Configuration Register.

4. Verify that DestID 0xFF01 is not associated with multicast mask 1 for port 4.

   Read the value 0x0000_0480 from the RapidIO Multicast DestID Association Register.

5. Set up the associate operations for DestID 0xFF01 and multicast mask 2.

   Write the value 0xFF01_0002 to the RapidIO Multicast DestID Configuration Register.

6. Verify that DestID 0xFF01 is associated with multicast mask 2 for port 4.

   Read the value 0x0000_0481 from the RapidIO Multicast DestID Association Register.

7. Set up the associate operations for DestID 0xFF01 and multicast mask 3.

   Write the value 0xFF01_0003 to the RapidIO Multicast DestID Configuration Register.

8. Verify that DestID 0xFF01 is not associated with multicast mask 3 for port 4.

   Read the value 0x0000_0480 from the RapidIO Multicast DestID Association Register.

Example Two

In this example, it is assumed that the system designer needs to know which DestIDs from 0xFF00 through 0xFF07 are associated with multicast mask 0 on port 4:

1. Set up the associate operations for DestID 0xFF00 and multicast mask 0.

   Write the value 0xFF00_0000 to the "RapidIO Multicast DestID Configuration Register".

2. Verify that DestID 0xFF00 is associated with multicast mask 0 for port 4.

   Write the value 0x0000_0480 to the "RapidIO Multicast DestID Association Register".

   Read the value 0x0000_0480 from the RapidIO Multicast DestID Association Register.

3. Set up the associate operations for DestID 0xFF00 and multicast mask 0.

   Write the value 0xFF01_0000 to the RapidIO Multicast DestID Configuration Register.

4. Verify that DestID 0xFF01 is not associated with multicast mask 0 for port 4.

   Read the value 0x0000_0480 from the RapidIO Multicast DestID Association Register.

5. Set up the associate operations for DestID 0xFF00 and multicast mask 0.

   Write the value 0xFF02_0000 to the RapidIO Multicast DestID Configuration Register.

6. Verify that DestID 0xFF02 is not associated with multicast mask 0 for port 4.

   Read the value 0x0000_0480 from the RapidIO Multicast DestID Association Register.

7. Set up the associate operations for DestID 0xFF00 and multicast mask 0.

   Write the value 0xFF03_0000 to the RapidIO Multicast DestID Configuration Register.

8. Verify that DestID 0xFF03 is not associated with multicast mask 0 for port 4.

   Read the value 0x0000_0480 from the RapidIO Multicast DestID Association Register.

9. Verify that DestID 0xFF03 is not associated with multicast mask 3 for port 2.

   Write the value 0x0000_0280 to the RapidIO Multicast DestID Association Register.

   Read the value 0x0000_0280 from the RapidIO Multicast DestID Association Register.

10. Verify that DestID 0xFF03 is associated with multicast mask 3 for port 3.

    Write the value 0x0000_0380 to the RapidIO Multicast DestID Association Register.

    Read the value 0x0000_0381 from the RapidIO Multicast DestID Association Register.

11. Verify that DestID 0xFF03 is not associated with multicast mask 3 for port 4.

    Write the value 0x0000_0480 to the RapidIO Multicast DestID Association Register.

    Read the value 0x0000_0480 from the RapidIO Multicast DestID Association Register.

12. Verify that DestID 0xFF03 is not associated with multicast mask 3 for port 5.

    Write the value 0x0000_0580 to the RapidIO Multicast DestID Association Register.

    Read the value 0x0000_0580 from the RapidIO Multicast DestID Association Register.

13. Verify that DestID 0xFF03 is not associated with multicast mask 3 for port 6.

    Write the value 0x0000_0680 to the RapidIO Multicast DestID Association Register.

    Read the value 0x0000_0680 from the RapidIO Multicast DestID Association Register.

14. Verify that DestID 0xFF03 is not associated with multicast mask 3 for port 7.

    Write the value 0x0000_0780 to the RapidIO Multicast DestID Association Register.

    Read the value 0x0000_0780 from the RapidIO Multicast DestID Association Register.

6.2.1.3 Configuring Multicast Masks Using the IDT Specific Registers

The Tsi620 also has a device-specific implementation for configuring the multicast masks. This implementation allows the direct writing of configuration information into the multicast group and vector tables through the "RapidIO Multicast Write Mask x Register" and the "RapidIO Multicast Write ID x Register".

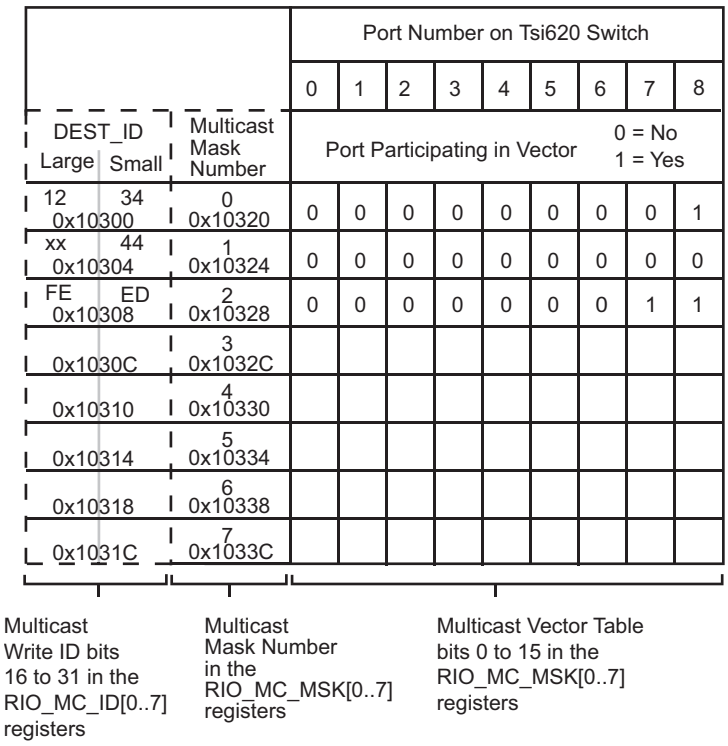
The method described in this section is a Tsi620-specific implementation. The implementation described in "Configuring Multicast Masks" conforms to the *RapidIO Interconnect Specification (Revision 1.3)*.

Eight registers contain the association between a DestID and the Multicast Mask number, and eight registers contain the association between the Multicast egress port vector table and the Multicast Mask.

In a closed architecture system, IDT recommends the use of the IDT-specific implementation. However, in an open-architecture system the use of the RapidIO compliant register set is recommended. The RapidIO compliant register permits switch device independent drivers that can be re-used.

Figure 21 shows the table architecture and the association between the registers and bit fields.

Figure 21: IDT-specific Multicast Mask Configuration



6.2.2 Multicast Work Queue

The multicast work queue accepts packets from ingress ports and forwards them to the broadcast buffers according to the multicast group table. The multicast work queue can store a maximum of 2208[1] bytes of packet data or 7 maximum-sized packets inside the multicast packet buffer. Once the packet buffer has stored 1936 bytes of data, it forces the Switch ISF to stop further packet transmission. For information on Switch ISF arbitration for the multicast work queue, see "Input Queuing Model for the Multicast Work Queue" and "Output Queuing Model for Multicast".

Once the first 8 bytes of a packet are received by the multicast work queue, the DestID and TT field of the packet are used by the multicast work queue to decide which ports must receive packet copies. The multicast group table computes an original multicast mask which indicates which ports (based on the original packets destID and TT field) that the packet copies should be sent to. The ingress port that the original packet was received on is removed from the multicast mask to form the multicast vector. If the broadcast buffer for a port has detected a maximum latency violation, a system designer can optionally automatically remove this port from the multicast mask (see "Multicast Maximum Latency Timer").

When the multicast work queue computes the multicast vector, it arbitrates to transmit packet copies to the broadcast buffers. The work queue operates in a cut-through method.

1. Packets are stored in an 8-byte boundary. Packets with length of non-multiple of 8 bytes are rounded up to the nearest multiple of 8 and stored in the buffer.

If a packet is STOMPed when it is received in the multicast work queue; it is never dropped. The STOMPed packet is replicated to the broadcast buffers. For more information on error scenarios, see "Error Management".

6.2.3 Broadcast Buffers

Each egress port has a dedicated broadcast buffer associated with it. The broadcast buffers accept packet copies from the multicast work queue, and forward these packet copies to the egress port. A broadcast buffer can accept one maximum sized packet (276 bytes) or up to eight smaller packets. Eight packets can only be accepted if the size of the packets, individually rounded up to the nearest multiple of 8, sum to less than 280 bytes.

For more information on Switch ISF arbitration for the broadcast buffers, see "Input Queuing Model for the Broadcast Buffer" and "Output Queuing Model for Multicast".

The broadcast buffers wait until a packet is received before starting arbitration with the egress port. Once the egress port acknowledges the broadcast buffer's request, the broadcast buffer transmits datums to the egress port at sustained rates of up to 10 Gbps[1]. The egress port receives the broadcast buffers data and may start to transfer the data as soon as the first datum is received.

In RapidIO technology, a datum means a word of data sent in a single clock cycle. When the Tsi620 is in 4x mode, a datum is 32 bits and in 1x mode, a datum is 8 bits.

If a packet is STOMPed when it is received at the broadcast buffer, it is never dropped. The STOMPed packet is transmitted to the egress port.

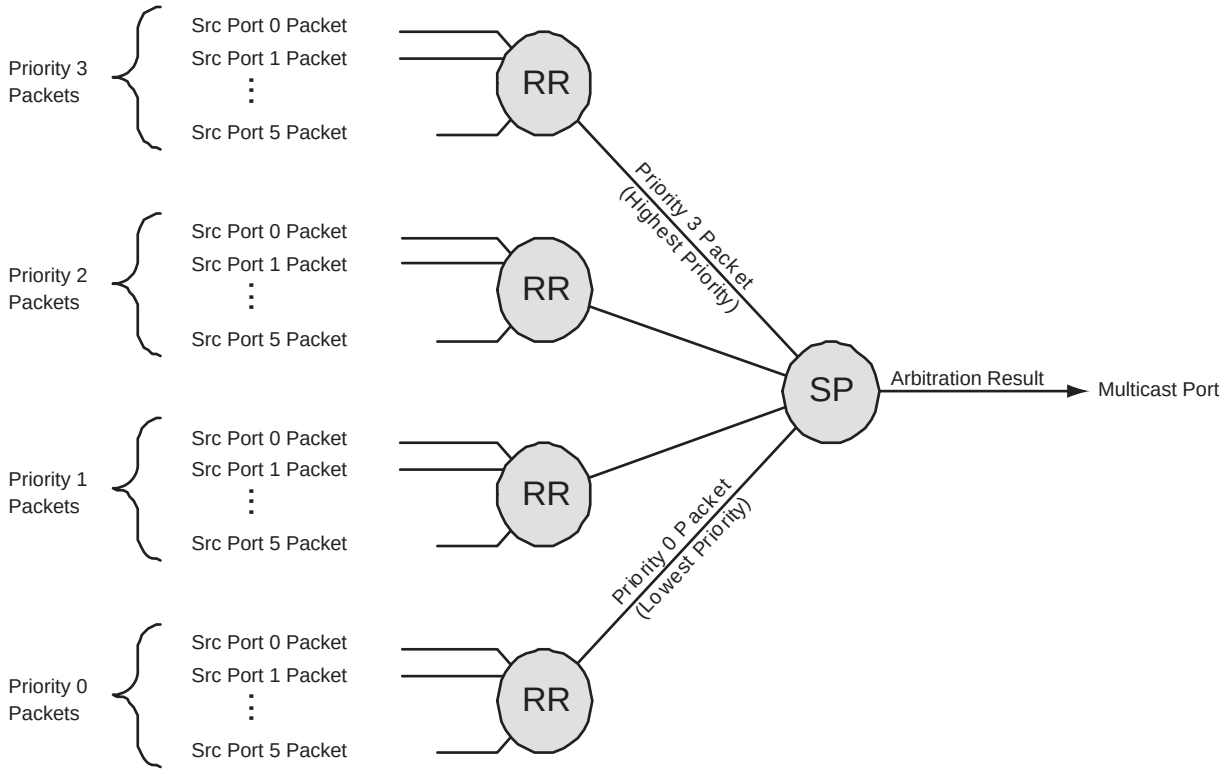
6.3 Arbitration for Multicast Port

The arbitration schemes that handles multiple ingress access to the Multicast port are simple round robin (RR) followed by the Strict Priority (see Figure 22). Each ingress port goes through RR arbitration based within their priority group. The outputs of the RR arbitration are handled with Strict Priority arbitration where higher priority packets are sent before lower priority ones.

The RR arbiter looks through the ports sequentially starting with Port 0, one after the other, to accept packets when available. No one port can monopolize the RR arbiter. When a port skips an opportunity to transmit because it carries no packet at the moment, the RR arbiter does not compensate for the lost chance and moves to the next port, in sequence, for available packet. The packets from the RR arbiters are then selected by the Strict Priority arbiter based on their Priority.

1. When the Switch ISF clock speed is set to 156.25 MHz.

Figure 22: Arbitration Algorithm for Multicast port



6.4 Error Management

Multicast packets have three sources of error:

- Packet TEA
- Packet STOMPing
- Exceeding the multicast maximum latency timer

It is also possible for packets to be dropped by the MCE based on the multicast mask (see “Multicast Group Tables”).

6.4.1 Packet TEA

A multicast packet at the head of the ingress queue is subject to TEA. The TEA function does not distinquish between multicast and unicast packets. A TEA’d multicast packet is dropped by the ingress port, and does not reach the multicast work queue (see “Multicast Work Queue”).

6.4.2 Multicast Packet Stomping

Transfers from the ingress port through the multicast work queue into the broadcast buffers operate in cut-through mode. It is possible for the ingress port's link partner to STOMP a packet, or terminate it due to some other error condition. It is also possible for the ingress port to detect a CRC or some other error with the packet. To handle these error situations, the Tsi620 supports STOMPing of a packet while it is transferred from ingress port to broadcast buffer.

When a packet arrives at the Tsi620 and a stomp is received part way thought the packet, the packet is still multicast to the egress ports, where each in turn stomps the packet when they transfer the packet.

6.4.3 Multicast Maximum Latency Timer

Each broadcast buffer has a separate maximum latency timer to enforce latency limits for a packet. The timer starts counting when the packet is completely received by the broadcast buffer. The timer has a maximum period of 0xFFFFFF*6.4 ns. The multicast latency timer is programmed using the "RapidIO Multicast Maximum Latency Counter CSR".

The maximum latency timer is associated only with the packet at the head of the broadcast buffer. The timer is reset when the packet is transferred from the broadcast buffer to the egress port.

Once a packet is received in the broadcast buffer, the multicast latency timer starts counting. If the multicast latency timer expires, an interrupt is raised (see "Switch ISF Broadcast Buffer Maximum Latency Expired Error Register") and a port-write may be sent, if enabled. All packets in the broadcast buffer at the time the multicast latency timer expires are discarded. A packet copy being transferred from the multicast work queue to the broadcast buffer when the multicast latency timer expires is also discarded. Depending on the system latency restrictions on multicast and the frequency with which the maximum latency timer expires, discarding packets within the broadcast buffer may be sufficient to allow the system to continue to operate. Note that the broadcast buffer is purged only once when the multicast latency timer expires. The next packet copy that comes from the multicast work queue is stored in the broadcast buffer as usual. The multicast latency timer is reset and when the new packet reaches the head of the buffer, the timer starts counting again.

In systems where exceeding the maximum latency timer is an indication of the failure of a port, system designers can set the AUTODEAD bit in "RapidIO Multicast Maximum Latency Counter CSR" to 1. If this bit is 1 when the multicast latency timer expires, in addition to packets being purged, the port is removed from multicast operation. By clearing the multicast latency timer error for that failed port, the traffic from the MCE to the broadcast buffer is restored and new packet copies can be received by the port. The AUTODEAD bit should only be set to 1 if the expiry of the maximum multicast latency timer means that an error is detected, and the continued operation of the system requires removal of the offending port. For the register bits related to notification and handling of a multicast latency timeout error, see "Switch ISF Broadcast Buffer Maximum Latency Expired Error Register" and "Switch Interrupt Status Register".

Note that the "Switch ISF Broadcast Buffer Maximum Latency Expired Override Register" can verify the operation of software associated with "Switch ISF Broadcast Buffer Maximum Latency Expired Error Register".

6.4.4 Silent Discard of Packets

The MCE may silently discard packets. The following are examples of where a multicast packet are dropped:

- A multicast group has no egress ports selected. The multicast mask and multicast vector for a packet using this multicast group is empty.
- The multicast group has only one port selected. If the multicast group has only one port selected which corresponds to the ingress port on which the packet was received, the multicast vector will be empty.
- The multicast group contains a port where the AUTODEAD bit is set in the “RapidIO Multicast Maximum Latency Counter CSR”.

Only this copy of the packet is dropped.

In these cases, no interrupt is issued, and no other information is latched regarding the packet that were dropped.

7. Switch Event Notification

This chapter describes the system of error and event notification in the Tsi620 RapidIO Switch. Topics discussed include the following:

7.1 Overview

The Tsi620 Switch uses the following methods to notify external devices about events that occur in its switching component:

1. Generate a RapidIO Port-write maintenance message when enabled [see the *RapidIO Interconnect Specification (Revision 1.3)*]
2. Assert an event through the Tsi620 interrupt scheme. Assert the INT_b interrupt pin when an enabled interrupt is generated.

Most events can generate both types of notification, however, some events only generate interrupts.

There is no priority or precedence between events in the error notification scheme of the Tsi620 Switch.

7.2 Switch Event Summary

Table 17 describes all the events that can be raised within the Tsi620 Switch, and whether these events generate an interrupt, a port-write, or both. The events detected by the Internal Switch Port (as indicated in Table 17) are a subset of those detected by the other Tsi620 Switch ports, since the Internal Switch Port does not have a serial link with the SREP Physical Layer.

Table 17: Switch Events

Switch Event Name (Status Bit)	Type	Description	Block Where Event Occurs	Can Generate Interrupt?	Can Generate Port-write?	Detected by Internal Switch Port
Maximum Retry Occurred	Error	This event occurs when a port's retry counter reaches the configured retry counter threshold. The same retry counter is incremented for retries of all packets (it is not a per-packet retry counter). The status of this event is contained in the IMP_SPEC_ERR bit in the "RapidIO Port x Error Detect CSR" and the MAX_RETRY bit in the "RapidIO Port x Interrupt Status Register". Note: The Retry Counter is reset if: • the counter reaches the retry threshold • a packet acceptance is received • a packet non-acceptance is received	RapidIO	Yes	Yes	No
Illegal Transaction	Error	This event occurs when an inbound port receives a transaction with one of the following errors: • Unmapped entry in LUT • Reserved TT field value for data packet and/or hop count not equal to zero The status of this event is contained in the IMP_SPEC_ERR bit in the "RapidIO Port x Error Detect CSR" and the ILL_TRANS_ERR bit in the "RapidIO Port x Interrupt Status Register".	RapidIO	Yes	Yes	Yes
LUT Parity Error	Error	This event occurs when a parity error is detected when a port is performing a destination ID lookup (see "RapidIO Port x LUT Parity Error Info CSR"). The status of this event is contained in the IMP_SPEC_ERR bit in the "RapidIO Port x Error Detect CSR" and the LUT_PAR_ERR bit in the "RapidIO Port x Interrupt Status Register".	RapidIO	Yes	Yes	Yes

Table 17: Switch Events (Continued)

Switch Event Name (Status Bit)	Type	Description	Block Where Event Occurs	Can Generate Interrupt?	Can Generate Port-write?	Detected by Internal Switch Port
Error Rate Failed Threshold Reached	Error	This event occurs when the error rate counter in the "RapidIO Port x Error Rate CSR" hits the selected Link-Failed threshold in the "RapidIO Port x Error Rate Threshold CSR". For more information on the Error Rate reporting function, see "Switch Error Rate Thresholds". The status of this event is contained in the OUTPUT_FAIL bit in the "RapidIO Port x Error and Status CSR". Note: Only the first TEA or MC_TEA error which changes the OUTPUT_DROP bit in the "RapidIO Port x Error and Status CSR" from 0 to 1 will be counted towards an Error Rate Degraded Threshold Reached or Error Rate Failed Threshold Reached event. Subsequent TEA or MC_TEA errors that occur while the OUTPUT_DROP bit is 1 do not cause the error rate counter in the "RapidIO Port x Error Rate CSR" to increment. Once the OUTPUT_DROP bit is cleared, the next TEA or MC_TEA error will cause the error rate counter to increment.	RapidIO	Yes	Yes	Yes
Error Rate Degraded Threshold Reached	Error	This event occurs when the error rate counter in the "RapidIO Port x Error Rate CSR" register hits the selected Link-Degraded threshold value in the "RapidIO Port x Error Rate Threshold CSR". For more information on the Error Rate reporting function, see "RapidIO Port x Error Rate Threshold CSR". The status of this event is contained in the OUTPUT_DEG bit in the "RapidIO Port x Error and Status CSR". Note: Only the first TEA or MC_TEA error which changes the "RapidIO Port x Error and Status CSR".OUTPUT_DROP bit from 0 to 1 will be counted towards an Error Rate Degraded Threshold Reached or Error Rate Failed Threshold Reached event. Subsequent TEA or MC_TEA errors which occur while the OUTPUT_DROP bit is 1 do not cause the error rate counter in the "RapidIO Port x Error Rate CSR" to increment. Once the OUTPUT_DROP bit is cleared, the next TEA or MC_TEA error will cause the error rate counter to increment.	RapidIO	Yes	Yes	Yes

Table 17: Switch Events (Continued)

Switch Event Name (Status Bit)	Type	Description	Block Where Event Occurs	Can Generate Interrupt?	Can Generate Port-write?	Detected by Internal Switch Port
Reset Request Received	Status	This event occurs when four consecutive reset requests are received by an inbound port. For more information on this event, see "Switch Reset". The status of this event is contained in the RCS bit in the "RapidIO Port x Multicast-Event Control Symbol and Reset Control Symbol Interrupt CSR", as well as the RCS bit in the "Switch Interrupt Status Register".	RapidIO	Yes	No	Yes
Multicast Symbol Received	Status	This event occurs when an inbound port receives a multicast control symbol. The status of this event is contained in the MCS bit in the "RapidIO Port x Multicast-Event Control Symbol and Reset Control Symbol Interrupt CSR", as well as the MCS bit in the "Switch Interrupt Status Register".	RapidIO	Yes	No	Yes
Outbound Queue Threshold Reached	Status	This event is raised when the "outbound queue threshold exceeded counter" reaches the counter threshold configured for that port. It detects congestion on outbound queues. The status of this event is contained in the OUTB_DEPTH bit of the "RapidIO Port x Interrupt Status Register".	RapidIO	Yes	Yes	No
Inbound Queue Threshold Reached	Status	This event is raised when the "inbound queue threshold exceeded counter" reaches the counter threshold configured for that port. This detects congestion on inbound queues. The status of this event is contained in the INB_DEPTH bit of the "RapidIO Port x Interrupt Status Register".	RapidIO	Yes	Yes	Yes
Inbound Reorder Count Threshold Reached	Status	This event is raised when the reorder count for a specific port reaches the configured reorder counter threshold for that port. The status of this event is contained in the INB_RDR bit of the "RapidIO Port x Interrupt Status Register". Note: The "reorder counter threshold" is set to 16hFFFF by default.	RapidIO	Yes	Yes	Yes

Table 17: Switch Events (Continued)

Switch Event Name (Status Bit)	Type	Description	Block Where Event Occurs	Can Generate Interrupt?	Can Generate Port-write?	Detected by Internal Switch Port
TEA in Fabric (Output Drop)	Error	This event is raised when a Switch ISF transmission request times out and a packet is dropped. The status of this event is contained in the TEA bit of the "RapidIO Port x Interrupt Status Register", the OUTPUT_DROP bit in the "RapidIO Port x Error and Status CSR", and the bits of the "Switch ISF Interrupt Status Register", and the TEA bit of the "Switch Interrupt Status Register".	Switch ISF	Yes	Yes	Yes
Multicast TEA	Error	This event is raised when the Multicast Engine fails to transfer a packet to the Broadcast Buffer before a timeout. The status of this event is contained in the MC_TEA bit of the "RapidIO Port x Interrupt Status Register".	Switch ISF	Yes	Yes	Yes
Fatal Port Error	Error	Inbound or Outbound port has encountered an error from which the hardware was unable to recover (fatal error). The following fatal errors are included: • Four link-request tries with link-response, but no outstanding ackID • Four link-request tries with timeout error for link-response • "Dead Link Timer" expires The status of this event is contained in the PORT_ERR bit of the "RapidIO Port x Error and Status CSR". This also causes the LINK_TO bit in the "RapidIO Port x Error Detect CSR" to be asserted.	RapidIO	Yes	Yes	No
Port Available Event	Status	This event is raised when a RapidIO port completes its automatic interface initialization after it detects a peer on the interface. This event is normally used to detect hot-swap events. The status of this event is contained in the LINK_INIT_NOTIFICATION bit of the "RapidIO Port x Error Detect CSR".	RapidIO	Yes	Yes	Yes

Table 17: Switch Events (Continued)

Switch Event Name (Status Bit)	Type	Description	Block Where Event Occurs	Can Generate Interrupt?	Can Generate Port-write?	Detected by Internal Switch Port
Illegal ackID in Control Symbol	Error	This event is raised when a RapidIO port receives a control symbol (packet-accepted, packet-not-accepted, or retry) with an ackID that is not in use. The status of this event is contained in the CS_ILL_ACKID bit of the "RapidIO Port x Error Detect CSR".	RapidIO	Yes	Yes[a]	No
Illegal ackID in Packet	Error	This event is raised when a RapidIO port receives a packet with an ackID that is not in sequence. The status of this event is contained in the PKT_ILL_ACKID bit of the "RapidIO Port x Error Detect CSR".	RapidIO	Yes	Yes[b]	No
Illegal Packet Size	Error	This event is raised when a RapidIO port receives a packet that is larger than 276 bytes. The status of this event is contained in the PKT_ILL_SIZE bit of the "RapidIO Port x Error Detect CSR".	RapidIO	Yes	Yes[b]	No
Illegal ackID in Link Response	Error	This event is raised when a RapidIO port receives an unused ackID in a Link-Response control symbol. Link-Response control symbols clear retry and error conditions on a link. The status of this event is contained in the LR_ACKID_ILL bit of the "RapidIO Port x Error Detect CSR".	RapidIO	Yes	Yes[b]	No
Protocol Error	Error	This event is raised when a RapidIO port receives an unexpected, but otherwise correctly composed, control symbol. An example would be receiving a link-response control symbol when no link-request is outstanding. The status of this event is contained in the PROT_ERR bit of the "RapidIO Port x Error Detect CSR".	RapidIO	Yes	Yes[b]	No
Delineation Error	Error	This event is raised when a RapidIO port receives an unaligned /SC/ or /PD/ symbol, or an undefined code group. The status of this event is contained in the DELIN_ERR bit of the "RapidIO Port x Error Detect CSR".	RapidIO	Yes	Yes[b]	No

Table 17: Switch Events (Continued)

Switch Event Name (Status Bit)	Type	Description	Block Where Event Occurs	Can Generate Interrupt?	Can Generate Port-write?	Detected by Internal Switch Port
Unexpected Acknowledge	Error	This event is raised when a RapidIO port receives an unexpected packet-accepted control symbol. The status of this event is contained in the CS_ACK_ILL bit of the "RapidIO Port x Error Detect CSR".	RapidIO	Yes	Yes[b]	No
Link Timeout	Error	This event is raised when a RapidIO port does not receive an acknowledgement (either a packet-accepted control symbol or a link-response control symbol) in time. The time is controlled by the "RapidIO Switch Port Link Timeout Control CSR". The status of this event is contained in the LINK_TO bit of the "RapidIO Port x Error Detect CSR".	RapidIO	Yes	Yes[b]	No
Control Symbol CRC Error	Error	This event is raised when a RapidIO port receives a Control Symbol packet with a CRC error. The status of this event is contained in the CS_CRC_ERR bit of the "RapidIO Port x Error Detect CSR"	RapidIO	Yes	Yes[b]	No
Control Symbol Not Accepted	Error	This event is raised when a RapidIO port receives a "Packet-Not-Accepted" Control Symbol. The status of this event is contained in the CS_NOT_ACC bit of the "RapidIO Port x Error Detect CSR"	RapidIO	Yes	Yes[b]	No
Packet CRC Error	Error	This event is raised when a RapidIO port receives a packet with a CRC error. The status of this event is contained in the CS_CRC_ERR bit of the "RapidIO Port x Error Detect CSR"	RapidIO	Yes	Yes[b]	No

Table 17: Switch Events (Continued)

Switch Event Name (Status Bit)	Type	Description	Block Where Event Occurs	Can Generate Interrupt?	Can Generate Port-write?	Detected by Internal Switch Port
I^2C Event	Status and Error	This event is raised when the I^2C block has an internal interrupt. The status of this event is contained in the I2C bit in the "Switch Interrupt Status Register", which is a logical OR of all bits in "I2C Interrupt Status Register" when the corresponding bits in "I2C Interrupt Enable Register" are enabled.	I^2C	Yes	No	N/A
Multicast Latency Exceeded	Error	This event occurs when a multicast request for a specific port cannot be transmitted in the time specified by the "Switch ISF Broadcast Buffer Maximum Latency Expired Error Register". This causes the MC_LAT_ERR bit to be set in the "Switch Interrupt Status Register". The port will not receive multicast requests unless/until the appropriate error bit in the "Switch ISF Broadcast Buffer Maximum Latency Expired Error Register" is cleared.	Multicast	Yes	Yes	N/A

a. Part of Error Rate Failed/Degraded Threshold counter.

b. For error events related to Maintenance packets with hop count = 0, see Table 4.

7.3 Switch Error Rate Thresholds

The error rate detection threshold function for the Tsi620 Switch works as follows:

- Each RapidIO port maintains a single error counter that is incremented each time one of the RapidIO errors is enabled (see Table 17).

  The RapidIO port's hardware can be configured to automatically decrement this counter. The rate at which the counter is decremented is configurable through the "RapidIO Port x Error Rate CSR".

- There are two error event thresholds configured per port on the Tsi620 Switch: Error Rate Failed Threshold Reached and the Error Rate Degraded Threshold Reached. These thresholds specify the counter level at which the port is considered either degraded or failed.

  — When the *failed threshold* is hit, the Error Rate Failed Threshold Reached event is raised. The port can be configured to raise an interrupt or issue a Port-write (or both). Another Failed event is not be raised until the counter falls below the threshold and then reaches it again, due to subsequent errors. It is also possible to configure the port to drop packets when the Error Rate Failed Threshold Reached event occurs.

  — When the *degraded threshold* is hit, the Error Rate Degraded Threshold Reached event is raised. The port can be configured to raise an interrupt or issue a Port-write (or both). Another Degraded event is not be raised until the counter falls below the threshold and then reaches it again, due to subsequent errors.

7.4 Switch Event Capture

When an enabled Tsi620 Switch error occurs, the port where the error occurred also logs information about the packet that caused the event. This information is stored within the "RapidIO Port x Packet Error Capture CSR 1 and Debug 2 Register" registers.

When a packet is logged in these registers, the VAL_CAPT bit is set in the "RapidIO Port x Error Capture Attributes CSR and Debug 0 Register". While the VAL_CAPT bit is set, further errors do not capture any packet information in order to preserve the first packet information that caused the enabled error. When the capture information is retrieved, the VAL_CAPT bit must be written to zero to clear it and allow subsequent error packets to be captured.

Table 18 lists the errors that cause the error counter to be incremented. All of these errors except the Implementation Specific Logical Error are defined in the *RapidIO Interconnect Specification (Revision 1.2)*. The Implementation Specific Logical Error is set when any of the Illegal Transaction, Maximum Retry, or Lookup Table parity error occur regardless of whether they are enabled or not.

When most of the errors in Table 18 occur, they are logged in the "RapidIO Port x Error Detect CSR". If the error is enabled in the "RapidIO Port x Error Rate Enable CSR", the error counter is incremented and information about the packet that caused the error is logged in the error capture registers (as long as the VAL_CAPT field is not already set in the "RapidIO Port x Error Capture Attributes CSR and Debug 0 Register").

For more information on these events, see the *RapidIO Interconnect Specification (Revision 1.2)* — Error Management.

Table 18: Switch Error Rate Error Events

RapidIO Error	Description	Capture Registers
Implementation Specific	The Tsi620 Switch uses the implementation-specific error to combine with other error events, so that they can be included within the Error Rate reporting function. These events are: • Reserved Transport Type detected (tt field = 10 or 11 for all but maintenance packets with hop count =0) • Maximum Retry Occurred Error • Unmapped DestID Error • Parity Error in Lookup Table • Switch ISF TEA Error • Multicast TEA Error • Port Fatal Error When any of these events occur, the Implementation Specific Error event is considered to have occurred. The status of this error is contained in the IMP_SPEC_ERR bit. [a]	Yes[b]
Received corrupt control symbol	Received a control symbol with a bad CRC value. The status of this error is contained in the CS_CRC_ERR bit. [a]	Yes

Table 18: Switch Error Rate Error Events (Continued)

RapidIO Error	Description	Capture Registers
Received acknowledge control symbol with unexpected ackID	Received an acknowledge control symbol with an unexpected ackID (packet-accepted or packet_retry). The status of this error is contained in the CS_ILL_ACKID bit. [a]	No
Received packet-not-accepted control symbol	Received packet-not-accepted acknowledge control symbol. The status of this error is contained in the CS_NOT_ACC bit. [a]	Yes
Receive packet with unexpected ackID	Received packet with unexpected ackID value - out-of-sequence ackID. The status of this error is contained in the PKT_ILL_ACKID bit. [a]	Yes
Received packet with bad CRC	Received packet with a bad CRC value. The status of this error is contained in the PKT_CRC_ERR bit. [a]	Yes
Received packet exceeds 276 Bytes	Received packet which exceeds the maximum allowed size. The status of this error is contained in the PKT_ILL_SIZE bit. [a]	Yes
Non-outstanding ackID	Link_response received with an ackId that is not outstanding. The status of this error is contained in the LR_ILL_ACKID bit. [a]	No
Protocol error	An unexpected packet or control symbol was received. The status of this error is contained in the PROT_ERR bit. [a]	Yes
Delineation error	Received unaligned /SC/ or /PD/ or undefined code-group. The status of this error is contained in the DELIN_ERR bit. [a]	No
Unsolicited acknowledge control symbol	An unexpected acknowledge control symbol was received. The status of this error is contained in the CS_ACK_ILL bit. [a]	Yes
Link timeout	An acknowledge-response or Link-response is not received within the specified timeout interval, see the "RapidIO Switch Port Link Timeout Control CSR". The status of this error is contained in the LINK_TO bit. [a]	No

a. This bit is in the "RapidIO Port x Error Detect CSR".

b. Capture Register information is valid only for the first 3 Implementation Specific Error, namely, Reserved Transport Type detected, Max Retry Error and Unmapped DestID Error.

7.5 Switch Port-Write Notifications

All RapidIO ports in the Tsi620 Switch can generate port-write messages based on interrupt events. The system is notified of most events that occur in the RapidIO ports through the RapidIO port-write message. The port-write function is enabled by default but can be disabled through the PW_DIS field in the "RapidIO Port x Mode CSR". Table 17 lists the events that cause RapidIO port-write messages.

When the port-write function is enabled, the occurrence of an enabled port-write capable event causes a port-write message to be sent to the destination ID specified in the "RapidIO Port Write Target Device ID CSR". If the event occurs but the interrupt capability is disabled (through the appropriate interrupt enable register) no port-write message is generated. The port-write message is generated for each event regardless of whether there is already a pending interrupt bit for the event set in the interrupt status register. However, when a new event occurs before the previous port-write is sent, no port-write is sent for the new event. The second port-write is sent only when the first one is cleared. The outstanding port-writes, if any, are indicated in the "Switch Port Write Outstanding Request Register".

The port-write packet does not have a guaranteed delivery and does not have an associated response (see *RapidIO Interconnect Specification (Revision 1.3)*). Depending on system design, a port write can be sent repeatedly until cleared. A programmable counter is defined in the PW_TIMER field of the "Switch Port Write Timeout Control Register" to control the frequency of transmission of the Port-write packets. When this timer expires and the port write has not yet been cleared, another port write is sent and the timer begins counting again.

7.5.1 Internal Switch Port Port-Write Notification

The Internal Switch Port cannot generate port-writes. Port-writes for Internal Switch Port events are generated by the SREP. The status of the Internal Switch Port events is used to trigger port-writes generated by the SREP. The SREP trigger for generation of port-writes by the Internal Switch Port is controlled by the PW_DIS bit in the "RIO Port 8 Mode CSR". If PW_DIS is 0, and any of the events in the Internal Switch Port is enabled for notification, then a port-write is generated by the SREP.

When the SREP must send a port-write for an event, the PORT_W_PEND bit in the "SREP Error and Status CSR" is set. The SREP continues to send port-writes at a programmable interval until the PORT_W_PEND bit is cleared.

The PORT_W_PEND bit in the Port 8 "RapidIO Port x Error and Status CSR" is never set.

If an Internal Switch Port event caused the SREP to generate a port-write, then the "SREP Switch Port Error" bit is set in the port-write data (see "SREP Event Notification").

7.5.2 Destination ID

There is only one port-write destination ID programmed for the entire Tsi620 Switch; a port-write event that occurs at any RapidIO port is sent to the same destination ID. The specified destination ID must be mapped within the port's lookup table (see "RapidIO Port Write Target Device ID CSR").

"RapidIO Port Write Target Device ID CSR" must be reprogrammed when a port is powered down. Other registers, such as "RapidIO Logical and Transport Layer Error Enable CSR" must also be reprogrammed when any port is powered down.

Note that since the Tsi620 Switch is handled as a separate RapidIO device, the SREP has a separate destination ID for its port-write packets.

7.5.3 Payload

The 16-byte data payload of the maintenance port-write packet contains the contents of several CSRs, the port that encountered the error condition, and implementation-specific information. The layout of the port-write packet is shown in the Table 19.

The payload of the maintenance port-write packet is defined by the *RapidIO Interconnect Specification (Revision 1.2) RapidIO Error Management Extensions*.

Table 19 shows the port-write packet data payload for error reporting.

Note that port-writes are sent at the priority programmed into the PW_PRIORITY field of the "RapidIO Port x Discovery Timer Register", which by default is priority 3 (highest priority).

Table 19: Port Write Packet Data Payload — Error Reporting

Data Payload Byte Offset	Word 0		Word 1
0x0	Component Tag CSR		Port n Error Detect CSR for Port ID
0x8	• Implementation-specific bits ("RapidIO Port x Interrupt Status Register"): • bit 12 = MC_TEA • bit 13 = LINK_INIT_NOTIFICATION • bit 14 = LUT_PAR_ERR bit • bit 15 = OUTPUT_DEG bit • bit 16 = OUTPUT_FAIL bit • bit 17 = INB_RDR bit • bit 18 = INB_DEPTH bit • bit 19 = OUTB_DEPTH • bit 20 = PORT_ERR • bit 21 = ILL_TRANS_ERR • bit 22 = OUTPUT_DROP • bit 23 = MAX_RETRY	Port ID (8 bits) (bits 24 to 31)	Logical/Transport Layer Error Detect CSR

7.5.4 Servicing Port Writes

The Tsi620 Switch supports a programming model for servicing port writes that minimizes the number of port writes generated by the Tsi620 Switch, and therefore, the number of specific interrupt events that must be handled by a system host.

As an example of this model, assume that a system host receives a port write because of an event on Port N. The host then follows these steps:

1. Determines what error caused the port-write to be generated by checking the "RapidIO Logical and Transport Layer Error Detect CSR" and the following registers of Port N:
   - "RapidIO Port x Error and Status CSR" (bits 6, 7, and 29)
   - "RapidIO Port x Interrupt Status Register"
2. Corrects the error conditions and clear the errors sources.
3. Clears the PORT_W_PEND bit in the "RapidIO Port x Error and Status CSR".

In the case when there are other errors from other ports that have generated a port-write, bits in the register "Switch Port Write Outstanding Request Register" are set.

7.5.5 Port-writes and Hot Insertion/Hot Extraction Notification

Port-write requests support hot insertion/extraction notification (see "Hot Insertion and Hot Extraction"). The sending device sets the PORT_W_PEND status bit in the "RapidIO Port x Error and Status CSR". Software indicates that it has seen the port-write operation by clearing the PORT_W_PEND bit. To clear the PORT_W_PEND bit, software must first clear the "RapidIO Port x Error Detect CSR".

7.5.6 Port-writes and Multicast

Port-writes can be multicast to multiple output links, depending on the DestID of the port-write. Using the multicast feature improves the likelihood of delivery of port-writes for link failures.

If a blocked or failed port becomes unblocked port-writes may be delivered late.

If a port generates a port-write packet, and the DestID of the port-write packet that is contained in the "RapidIO Port Write Target Device ID CSR" is also contained in a multicast group, the port write will be routed to the Multicast Engine. The Multicast Engine does not distinguish between port-write packets and other packet types. The Multicast Engine will multicast the port-write packet to ports according to the mask associated with the DestID of a multicast group. However, in respecting the RapidIO Multicast specification, if the originating port is contained in the multicast mask, the Multicast Engine will remove the source port from the mask list preventing the port-write packet from being transmitted out of the port that originated the port-write packet.

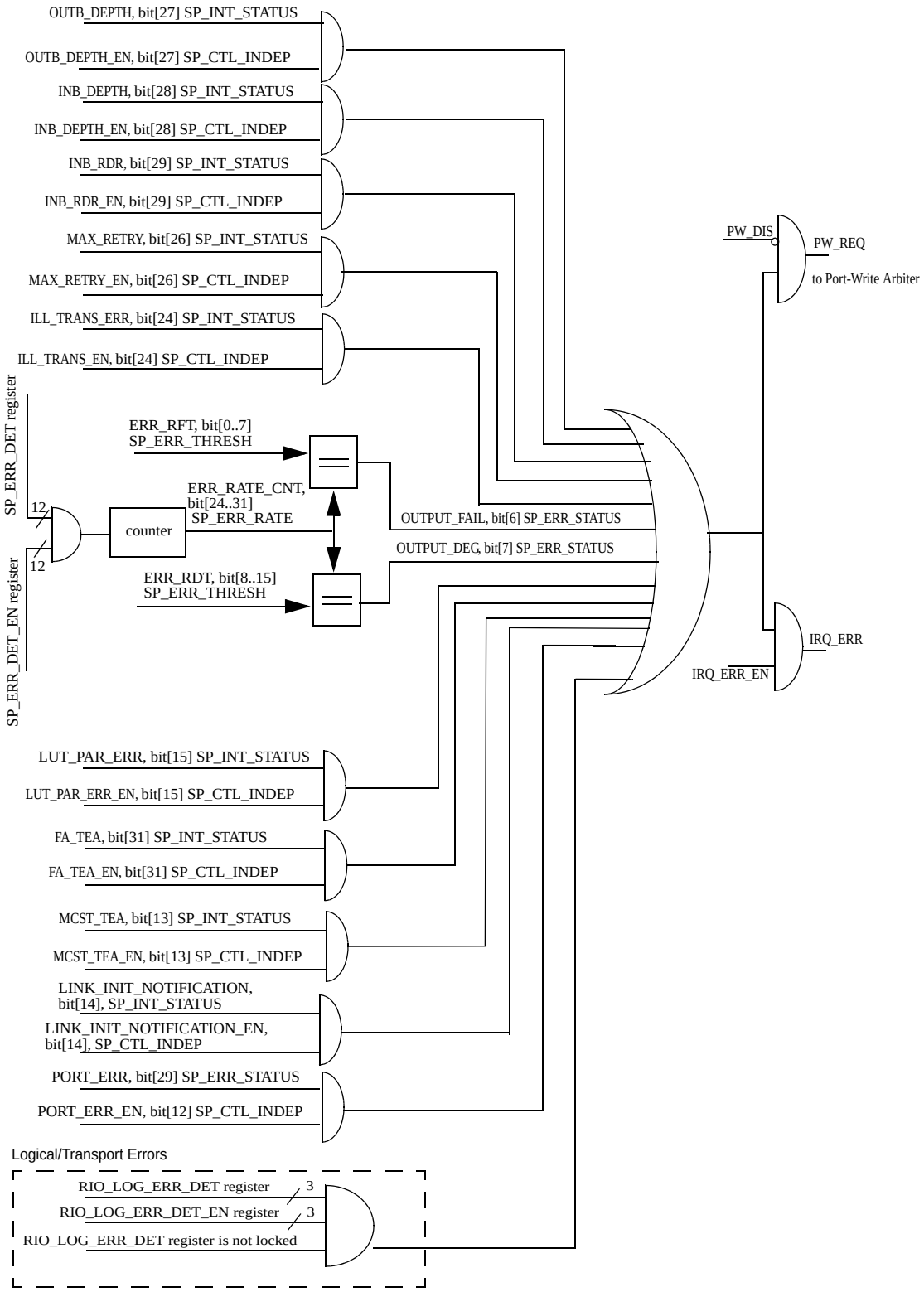
7.6 Switch Interrupt Notifications

In the Tsi620 Switch, interrupts are hierarchical which allows software to determine the cause of the interrupt with minimum register access.

System designers must decide on a maximum rate of interrupt notifications, and set the error thresholds appropriately.

Figure 23 shows the interrupt hierarchy within the Tsi620 Switch's RapidIO ports.

Figure 23: RapidIO Block Interrupt and Port Write Hierarchy



7.6.1 INT_b Signal

The external interrupt signal, INT_b, is at the top level of the interrupt hierarchy. This active-low signal is asserted when any fully enabled interrupt occurs. The INT_b signal remains asserted until all interrupts are cleared within the Tsi620 Switch. The INT_b signal is an input to the Tsi620 Event Notification scheme.

The INT_b signal is driven by the "Switch Interrupt Status Register". The signal is asserted when any bit within this register is set and its corresponding enable bit is set in the "Switch Interrupt Enable Register". If an interrupt in the Enable register is not enabled, the bit in Status register is still set when an interrupt occurs, but INT_b is not asserted.

Interrupts can be cleared by either writing the interrupt status register bit or by disabling that interrupt. When a previously asserted interrupt is disabled, the interrupt bit remains set in the interrupt status register, but the interrupt is no longer passed up the interrupt hierarchy.

7.6.2 Switch Interrupt Status Register and Interrupt Handling

The "Switch Interrupt Status Register" must be read to determine why the interrupt was raised. Interrupt sources in this register allow the interrupt service routine to decide which port raised an interrupt. The I^2C Interface has a separate indicator bit, as it is not associated with any port.

Two functions that are port specific have separate indicator bits to allow for faster interrupt handling. These functions are Multicast Event Control Symbol reception, and reception of a valid reset control symbol sequence. Both interrupts can be cleared with one register write to the broadcast address of the "RapidIO Port x Mode CSR".

After the software has read the "Switch Interrupt Status Register" and determined which port has an interrupt pending, the port's interrupt status registers must be accessed to determine the cause. Each port contains an interrupt status register (see "RapidIO Port x Interrupt Status Register") and an associated interrupt enable register (see "RapidIO Port x Control Independent Register"). When an interrupt occurs within aport, the associated bit for that interrupt is set within the interrupt status register regardless of the setting of the interrupt enable register. The port only notifies the "Switch Interrupt Status Register" if that interrupt is enabled.

7.6.2.1 Other Interrupts Types and Interrupt Handling

TEA events have a separate register that allows an interrupt handler to determine on which port the TEA occurred (see "Switch ISF Interrupt Status Register"). Similarly, multicast latency errors have a separate register to indicate which port is unable to receive multicast packets (see "Switch ISF Broadcast Buffer Maximum Latency Expired Error Register").

There is only one logical layer error per device. In the "Switch Interrupt Status Register", the occurrence of a logical error is associated with a port (see PORTx in the "Switch Interrupt Status Register"). When a port error is indicated, the logical error status registers must also be checked. For information on the different logical layer errors detected by the Tsi620 Switch, see "Switch Event Summary".

For those port-specific interrupt causes that are not visible in the "Switch Interrupt Status Register", the interrupt handler must access the port's registers to determine the cause of an interrupt. There are two RapidIO standard registers that must be accessed: "RapidIO Port x Error and Status CSR" and "RapidIO Port x Error Rate CSR".

The IMP_SPEC_ERR (implementation-specific error) bit in the "RapidIO Port x Error Rate CSR" leads to a number of other IDT-specific error and performance-related interrupts. These interrupts are located in another register, the "RapidIO Port x Interrupt Status Register". The Tsi620 also provides the implementation-specific option of sending an interrupt for some of the bits in the "RapidIO Port x Error Rate CSR".

All interrupt sources and their associated data can be configured by register writes to facilitate the testing of software.

Table 20: Port x Error and Status Register Status

Status Bit	Further Information	Interrupt Enable	Interrupt Clearing
OUTPUT_DROP	"RapidIO Port x Error and Status CSR" "RapidIO Port x Error Rate CSR" "RapidIO Port x Error Rate Threshold CSR" "RapidIO Port x Interrupt Status Register"	"RapidIO Port x Control Independent Register" for TEA interrupts. There is no specific interrupt status bit for OUTPUT_DROP of packets when OUTPUT_FAIL is asserted. This must be inferred by the fact that OUTPUT_DROP is set and TEA interrupts are not asserted.	Write 1 to OUTPUT_DROP to clear the "RapidIO Port x Error and Status CSR" TEA interrupt bit for the port. Writing 1 to this bit does not clear "RapidIO Port x Error Rate CSR" [ERR_RATE_CNT] so the next time a packet is sent to this port it may be dropped again.
OUTPUT_FAIL	"RapidIO Port x Error Rate CSR" "RapidIO Port x Error Rate Threshold CSR"	"RapidIO Port x Error Rate Threshold CSR"	Write 1 to OUTPUT_FAIL to clear this interrupt. Writing 1 to this bit does not clear "RapidIO Port x Error Rate CSR" [ERR_RATE_CNT] so this bit may become set again immediately.
OUTPUT_DEG	"RapidIO Port x Error Rate CSR" "RapidIO Port x Error Rate Threshold CSR"	"RapidIO Port x Error Rate Threshold CSR"	Write 1 to OUTPUT_DEG to clear this interrupt. Writing 1 to this bit does not clear "RapidIO Port x Error Rate CSR"[ERR_RATE_CNT], so this bit may become set again immediately.

7.6.3 Interrupt Notification and Port Writes

In the Tsi620, all RapidIO ports can generate port-write messages based on interrupt events. Because of this feature, the RapidIO interrupt enables also control whether a port-write message is issued for each interrupt. In each port the IRQ_EN bit in the "RapidIO Port x Control Independent Register" controls whether any enabled interrupts are passed to the "Switch Interrupt Status Register" and therefore create a port-write. If the IRQ_EN bit is disabled, no interrupt passes to the "Switch Interrupt Status Register" from this port.

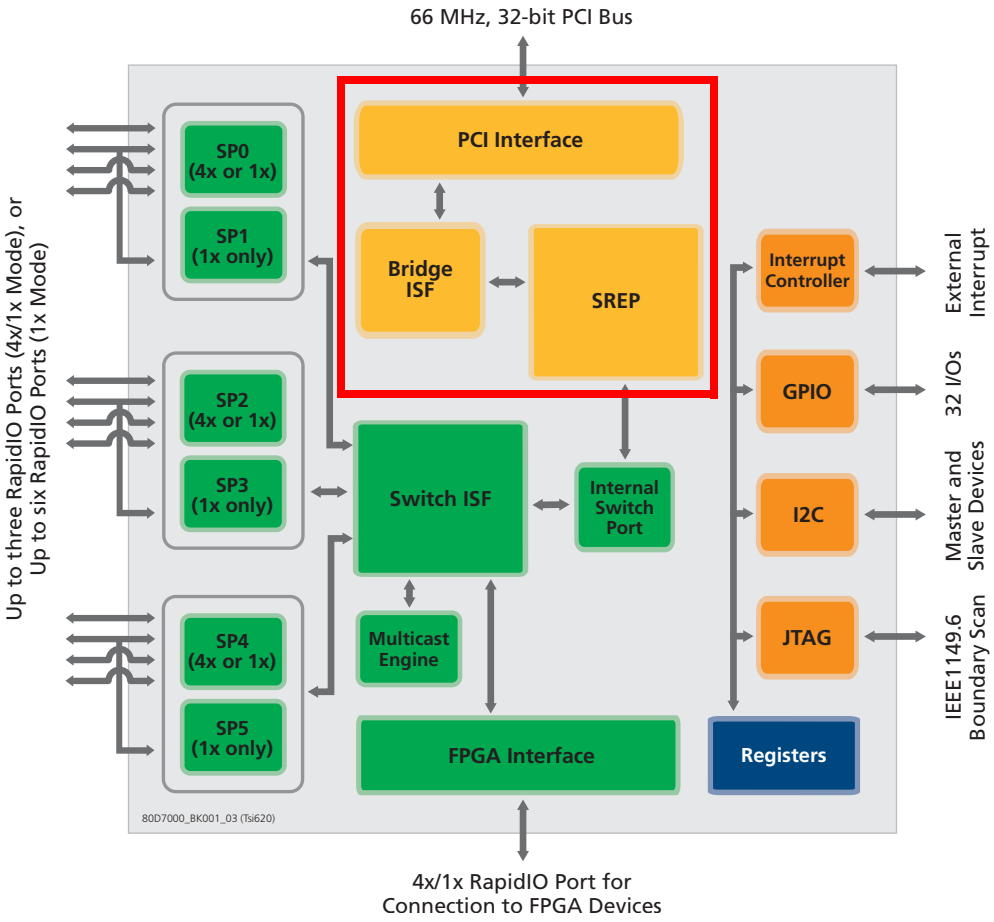
7.6.4 Reset Control Symbol and Interrupt Handling

The receipt of a valid reset control symbol sequence is a port-specific feature that has separate indicator bits to allow for faster interrupt handling (see "RapidIO Port x Multicast-Event Control Symbol and Reset Control Symbol Interrupt CSR").

PART 3: RAPIDIO-to-PCI BRIDGE (Tsi620 BRIDGE)

The bridging component of the Tsi620 (also called the “Tsi620 Bridge”), includes the modules that perform the various RapidIO-to-PCI and PCI-to-RapidIO bridging functions. The Tsi620 Bridge provides non-transparent bridging between up to four PCI devices and devices connected to the Tsi620’s RapidIO ports.

The Tsi620 Bridge topics that are discussed in this part of the document are highlighted in the following figure.



8. SREP Physical Layer

Topics discussed include the following:

8.1 Overview

The Physical Layer of the SREP is connected to the Internal Switch Port of the Tsi620. The Physical Layer consists of the following:

8.1.1 Features

The SREP Physical Layer has the following capabilities:

- Supports standard CARs and CSR registers for an endpoint
- Mimics operation of PORT_DIS, OUTPUT_EN, INPUT_EN, PORT_LOCKOUT to allow software to control traffic between the switch and the bridge.
- Mimics transmission of link-request/input-status, link-request/reset, and multicast event control symbols to support link maintenance and reset functions
- Supports Physical Layer Flow Control (SREP watermarks and buffer release management)
- Supports cut-through and store-and-forward modes of operation

8.1.2 Features Not Supported

The following features are not supported in the Physical Layer endpoint port:

- No MAC registers, with the exception of SMAC_DLOOP_CLK_SEL
- No SerDes registers
- No performance monitoring registers

8.2 Link Maintenance Functions

The SREP Physical Layer, and the Internal Switch Port mimics the operation of a RapidIO compliant link between the Switch and Bridge. One of the functions for RapidIO compliance is link maintenance. Link maintenance includes:

- The ability to restrict the kinds of transactions exchanged between the Internal Switch Port and the SREP Physical Layer
- Discard packets if the link is deemed “dead”
- Receive notification of when the link partner is available and not available

The SREP Physical Layer has the capability to restrict access from the Internal Switch Port, in accordance with the following standard bit fields in the “SREP Port Control CSR”:

- PORT_DIS – Controls whether or not any communication can occur between the Internal Switch Port and the SREP Physical Layer.
- PORT_LOCKOUT – When the PORT_LOCKOUT bit is set, only link request (RESET and INPUT_STATUS) and MECS control symbols can be exchanged. Packets cannot be exchanged when PORT_LOCKOUT is set. When the PORT_LOCKOUT bit is cleared, access is controlled by OUTPUT_EN and INPUT_EN.
- OUTPUT_EN – If OUTPUT_EN is 0, then only maintenance requests/responses can be transmitted. All other transactions are discarded.
- INPUT_EN – If INPUT_EN is 0, then only maintenance requests/responses can be transmitted. All other transactions are discarded.

These control bits are located in both the Internal Switch Port (see Port 8 in the “RapidIO Serial Port x Control CSR”) and SREP Physical Layer (see “SREP Port Control CSR”).

When the PORT_DIS bit is set to 0 in both the Internal Switch Port and SREP Physical Layer ports, control symbols and packets can be exchanged (see Table 21). If PORT_DIS is set to 1 in either the Internal Switch Port and SREP Physical Layer port, the PORT_OK bits in both ports are set to 0. No control symbols or packets of any kind can be exchanged when PORT_OK is 0. Any packets to be transmitted when PORT_DIS is set to 1 are discarded. No packets can be received when PORT_DIS is set to 1.

Note that both ports must be powered up by setting the PWRDN_X4 bit in the “SREP Digital Loopback and Clock Selection Register” and Port 8 in the “RapidIO SMAC x Digital Loopback and Clock Selection Register” before PORT_DIS can have an effect.

If either PWRDN_X4 bit is set, the link behaves as if the PORT_DIS bit is set.

If PORT_DIS is set to 0 in both the Internal Switch Port and SREP Physical Layer ports, then control symbols can be exchanged. Packet exchange is controlled by the PORT_LOCKOUT bits of both ports. If PORT_LOCKOUT is 1 in the receiver, then attempts at packet transmission cause the OUTPUT_ERR and OUTPUT_ERR_STOP bits in the transmitter to be set to 1 (see Table 22). This mimics the behavior of attempting to send a packet when PORT_LOCKOUT is set to 0 in the transmitter but is 1 in the link partner. LINK_REQUEST/INPUT_STATUS, LINK_REQUEST/RESET and MECS control symbols can be exchanged.

If PORT_LOCKOUT is set to 0 in both Internal Switch Port and the SREP Physical Layer, then the exchange of packets is controlled by the OUTPUT_EN and INPUT_EN bits in each of the ports (see Table 23). Regardless of the settings of OUTPUT_EN and INPUT_EN, maintenance packets and responses can be received and sent by the Internal Switch Port and SREP Physical Layer port. For non-maintenance packets, three situations are possible:

- If the transmitters OUTPUT_EN is 0 and a non-maintenance packet is to be transmitted, then the non-maintenance packet is dropped.
- If transmitters OUTPUT_EN is 1 and the link partners INPUT_EN is 0, then the non-maintenance packet is dropped, the transmitters port enters the OUTPUT_ERR state and the link partner's port enters the INPUT_ERR state.
- If the transmitters OUTPUT_EN is 1 and the link partners INPUT_EN is 1, then the non-maintenance packet is transferred successfully. Note that any response packets are still subject to these conditions in the opposite direction.

Table 21: PORT_DIS Controls

Transmitter PORT_DIS	Link Partner PORT_DIS	Transmitter			Link Partner		
		PORT_OK/ PORT_ UNINIT	OUTPUT_ ERR, OUTPUT_ ERR_STOP	INPUT_ERR, INPUT_ERR _ STOP	PORT_OK/ PORT_ UNINIT	OUTPUT_ ERR, OUTPUT_ ERR_STOP	INPUT_ERR, INPUT_ERR _ STOP
1	1	0/1	0/0	0/0	0/1	0/0	0/0
1	0	0/1	0/0	0/0	0/1	0/0	0/0
0	1	0/1	0/0	0/0	0/1	0/0	0/0
0	0	1/0	0/0	0/0	1/0	0/0	0/0

Table 22: PORT_LOCKOUT Controls

Transmitter PORT_ LOCKOUT	Link Partner PORT_ LOCKOUT	Transmit Type	Transmitter			Link Partner		
			PORT_OK/ PORT_ UNINIT	OUTPUT_ ERR, OUTPUT_ ERR_STOP	INPUT_ER R, INPUT_ER R_STOP	PORT_OK/ PORT_ UNINIT	OUTPUT_ ERR, OUTPUT_ ERR_STOP	INPUT_ER R, INPUT_ER R_STOP
1	N/A	Control Symbol[a]	1/0	0/0	0/0	1/0	0/0	0/0
		Maint	1/0	0/0	0/0	1/0	0/0	0/0
		Other	1/0	0/0	0/0	1/0	0/0	0/0

Table 22: PORT_LOCKOUT Controls (Continued)

Transmitter PORT_ LOCKOUT	Link Partner PORT_ LOCKOUT	Transmit Type	Transmitter			Link Partner		
			PORT_OK/ PORT_ UNINIT	OUTPUT_ ERR, OUTPUT_ ERR_STOP	INPUT_ER R, INPUT_ER R_STOP	PORT_OK/ PORT_ UNINIT	OUTPUT_ ERR, OUTPUT_ ERR_STOP	INPUT_ER R, INPUT_ER R_STOP
0	1	Control Symbol	1/0	0/0	0/0	1/0	0/0	0/0
		Maint	1/0	1/1	0/0	1/0	0/0	1/1
		Other	1/0	1/1	0/0	1/0	0/0	1/1

a. No control symbols (MECS, LINK_REQUEST/INPUT_STATUS and LINK_REQUEST/RESET) can be exchanged.

Table 23: OUTPUT_EN and INPUT_EN Controls

Transmitter		Link Partner			Transmitter		Link Partner	
OP_EN	IP_EN	OP_EN	IP_EN	Pkt Type	OUTPUT_ ERR OUTPUT_ ERR_STOP	INPUT_ ERR INPUT_ ERR_STOP	OUTPUT_ ERR OUTPUT_ ERR_STOP	INPUT_ ERR INPUT_ ERR_STOP
N/A	N/A	N/A	N/A	Mtc	0	0	0	0
0	N/A	N/A	N/A	Oth	0	0	0	0
1	N/A	N/A	0	Oth	1	0	0	1
1	N/A	0	1	NWRITE, SWRITE	0	0	0	0
				NREAD, NWRITE_ R	0	0	0	0
1	0	1	1	NWRITE, SWRITE	0	0	0	0
				NREAD, NWRITE_ R	0	1	1	0
1	1	1	1	Oth	0/0	0/0	0/0	0/0

Note that PORT_OK, PORT_UNINIT, INPUT_ERR and OUTPUT_ERR indicate the current status of the link, and so change automatically when PORT_DIS, PORT_LOCKOUT, OUTPUT_EN and INPUT_EN change. OUTPUT_ERR_STOP and INPUT_ERR_STOP require software to clear them status bits.

The SREP Physical Layer can generate events based on the status of the virtual RapidIO link for hot insertion and extraction. The registers and bit fields relevant to event notification for SREP Physical Layer link status are located in the Internal Switch Port and SREP Physical Layer port registers:

- PORT_LOCKOUT bit in "SREP Port Control CSR" and Port 8 in the "RapidIO Serial Port x Control CSR" – When the PORT_LOCKOUT bit is set, and the link has successfully initialized, an interrupt or port-write can be used to notify the system host that the Internal Switch Port and SREP Physical Layer can exchange packets.
- LINK_INIT_NOTIFICATION_EN bit in the "SREP Control Independent Register" and Port 8 in the "RapidIO Port x Control Independent Register" – When the LINK_INIT_NOTIFICATION bit is set, and the PORT_LOCKOUT bit is set, and the link has successfully initialized, an interrupt is asserted to notify a system host that the Internal Switch Port and SREP Physical Layer can exchange packets.
- LINK_INIT_NOTIFICATION bit in the "SREP Interrupt Status Register" and Port 8 in the "RapidIO Port x Interrupt Status Register" – When the PORT_LOCKOUT bit is set, this bit indicates that the Internal Switch Port and SREP Physical Layer can exchange packets.
- LINK_INIT_NOTIFICATION bit in the "SREP Interrupt Generate Register" and Port 8 in the "RapidIO Port x Interrupt Generate Register" – This bit causes the LINK_INIT_NOTIFICATION bit of the "SREP Interrupt Status Register" and Port 8 in the "RapidIO Port x Interrupt Status Register", respectively, to be set. This function is useful for testing software.

8.2.1 Link Initialization Events

The SREP Physical Layer can be configured to notify the system host when PORT_DIS in the SREP Physical Layer port is set to 0. The PORT_LOCKOUT bit must be set to allow the LINK_INIT_NOTIFICATION bit in the "SREP Interrupt Status Register" to be set. To determine that PORT_DIS has occurred, the system host has the option of polling the "SREP Interrupt Status Register", or of setting the LINK_INIT_NOTIFICATION_EN bit in the "SREP Control Independent Register" to assert an interrupt or send port write transactions.

Once the system host is notified that a new component is inserted, the LINK_INIT_NOTIFICATION bit should be cleared in the "SREP Interrupt Status Register" to stop the assertion of interrupts and/or transmission of port writes.

The PORT_LOCKOUT in the "SREP Port Control CSR" must be cleared to allow the system host to allow packets to be exchanged between the Internal Switch Port and SREP Physical Layer. The OUTPUT_EN and INPUT_EN bits must be set according to the amount of access the system designer requires. Error notification for the Internal Switch Port and SREP Physical Layer should also be enabled, if required by the system designer.

8.2.2 Bridge Shutdown

In some applications, the Bridge may not be required. To conserve power, power down the Internal Switch Port and SREP Physical Layer. The Internal Switch Port and SREP Physical Layer should not have any transactions flowing through them when they are powered down. The "SREP Port Control CSR". PORT_LOCKOUT bit must be set to drop all packets being transferred from the Internal Switch Port. Once the Port 8 "RapidIO Serial Port x Control CSR".PORT_LOCKOUT is also set, the Internal Switch Port and SREP can be powered down safely.

Any packets sent to a powered down Internal Switch Port or SREP Physical Layer port are discarded.

The SREP Physical Layer does not have an external SPn_PWRDN pin. The SREP Physical Layer can only be powered down by software action, or I2C BOOT writing to the appropriate registers.

The SREP Physical Layer port can be powered down through the use of the PWDN_X4 bit in the "SREP Digital Loopback and Clock Selection Register".

If either the Internal Switch Port or SREP Physical Layer port are powered down, then the link acts as if the PORT_DIS bit is set in the port that is powered down. For more information on the port's behavior when PORT_DIS is set, see "Link Maintenance Functions".

To conserve power when the Bridge function is not required, power down both the Internal Switch Port and SREP Physical Layer at the same time. This powers down the PCI PLL, and removes the clock from the Internal Switch Port, SREP, the Bridge ISF, and the PCI Interface. Transactions are not accepted by the PCI Interface.

Clearing the PWDN_X4 bits will not power up the Internal Switch Port, SREP, the Bridge ISF, or the PCI Interface. If both PWDN_X4 bits are set, the only method to return the Internal Switch Port, the SREP, the Bridge ISF, and the PCI Interface to service is to reset the entire device using a "Chip Reset".

8.2.3 Operation of the Dead Link Timer

The Dead Link Timer is not supported by the SREP Physical Layer. When PORT_DIS is set in either the Internal Switch Port or SREP, then packets destined to the Switch from the Bridge, and vice versa, are discarded (see "Link Maintenance Functions").

8.2.4 Software-assisted Error Recovery

The SREP Physical Layer mimics the operation of the software-assisted error recovery function located in the RapidIO standard registers:

- "SREP Link Maintenance Request CSR"
- "SREP Link Maintenance Response CSR"
- "SREP Local ackID Status CSR"

Software-assisted error recovery works as follows:

1. Write to the "SREP Link Maintenance Request CSR" to send a link-request/input-status to the link partner.
2. Poll "SREP Link Maintenance Response CSR" until a response is returned, as indicated by the RESP_VLD bit in these registers.
3. Program the "SREP Local ackID Status CSR" registers according to the values in the response.

Control symbols can only be exchanged when the PORT_OK bit is 1.

If PORT_OK is 0 and a link-request/input-status request is sent to the link partner, then the request is discarded and the "SREP Link Maintenance Response CSR".RESP_VLD bit never becomes 1 (see "Link Maintenance Functions").

The "SREP Local ackID Status CSR".OUTSTANDING field is always 0 since no acknowledgements are necessary with the Internal Switch Port or SREP Physical Layer. Writing to the CLR_PKTS bit in the "SREP Local ackID Status CSR" has no effect, because there are never any outstanding transactions.

The INBOUND and OUTBOUND registers in the "SREP Local ackID Status CSR" can be written with any value.

The contents of the "SREP Local ackID Status CSR".INBOUND field is in the "SREP Link Maintenance Response CSR".ACK_ID_STAT field when "SREP Link Maintenance Response CSR".RESP_VLD is 1. The "SREP Link Maintenance Response CSR".LINK_STAT field is set as follows when RESP_VLD is 1:

- 0b00101 = Error-Stopped - Port 8 "RapidIO Port x Error and Status CSR".INPUT_ERR_STOP bit was 1.
- 0b10000 = OK - Port 8 "RapidIO Port x Error and Status CSR".INPUT_ERR_STOP bit is 0.

Sending a link-request/input-status control symbol clears the INPUT_ERR_STOP bit in the link partner, and the OUTPUT_ERR_STOP bit in the requestor.

When RESP_VLD is 0, then the "SREP Link Maintenance Response CSR".ACK_ID_STAT field has not been updated.

8.2.5 SREP Hot Insertion and Extraction

The SREP Physical Layer supports "hot insertion" and "hot extraction" in a manner that is compatible with the Tsi57x family on the mimiced link (see "Hot Insertion and Hot Extraction").

8.2.5.1 Hot Extraction Procedure

1. Set the PORT_LOCKOUT bit in the "SREP Port Control CSR" or Port 8 "RapidIO Serial Port x Control CSR".
2. Once the 'PORT_OK' bit in the "SREP Error and Status CSR" or Port 8 "RapidIO Port x Error and Status CSR" has changed status from 1 to 0, the link partner is disconnected. The PORT_OK bit will change to 0 if the PORT_DIS bit is set in the link partner, or if the PWDN_X4 bit is set in the "SREP Digital Loopback and Clock Selection Register".
3. A LINK_INIT_NOTIFICATION event occurs when the link partner is disconnected. This sets the LINK_INIT_NOTIFICATION bit in the "SREP Interrupt Status Register" or Port 8 "RapidIO Port x Interrupt Status Register".
4. If the LINK_INIT_NOTIFICATION_EN bit is set in the "SREP Control Independent Register" or Port 8 "RapidIO Port x Control Independent Register", then an interrupt is asserted or a port-write occurs.

8.2.5.2 Hot Insertion Procedure

1. Set the PORT_LOCKOUT bit in the "SREP Port Control CSR" or Port 8 "RapidIO Serial Port x Control CSR".
2. Once the 'PORT_OK' bit in the "SREP Error and Status CSR" or Port 8 "RapidIO Port x Error and Status CSR" has changed status from 0 to 1, the link partner is reconnected. The PORT_OK bit will change to 1 if the PORT_DIS bit is cleared in the link partner and the PWDN_X4 bit is cleared in the "SREP Digital Loopback and Clock Selection Register".
3. A LINK_INIT_NOTIFICATION event occurs when the link partner is reconnected. This sets the LINK_INIT_NOTIFICATION bit in the "SREP Interrupt Status Register" or Port 8 "RapidIO Port x Interrupt Status Register".
4. If the LINK_INIT_NOTIFICATION_EN bit is set in the "SREP Control Independent Register" or Port 8 "RapidIO Port x Control Independent Register", then an interrupt is asserted or a port-write occurs.

8.2.6 Port Loopback Testing

The SREP Physical Layer does not support any loopback functionality.

8.3 Physical Layer Events

The only error event that the SREP Physical Layer detects is the link initialization notification event. This event is signaled when the LINK_INIT_NOTIFICATION bit of the "SREP Control Independent Register" is set to 1, and the PORT_OK bit is 1 in the "SREP Error and Status CSR". This SREP Physical Layer causes a PHY_IMP_SPEC event to be signalled by the SREP.

All physical layer error management registers are supported by the SREP Physical Layer, but none of the physical layer errors can be detected, since a RapidIO link does not exist between the SREP Physical Layer and the Internal Switch Port.

The SREP Physical Layer port does not support the "degraded" or "fail" thresholds, since a RapidIO link does not exist between the SREP Physical Layer and Internal Switch Port. In all other ways, the SREP Physical Layer port error registers are compliant to the *RapidIO Interconnect Specification (Revision 1.3)*.

The SREP Physical Layer does mimic the exchange of Multicast-Event Control Symbols (MECS) as well as Link-Request/Reset control symbols with the Internal Switch Port. Reception of a MECS causes the MECS bit to be set in the "SREP Interrupt Status Register". Reception of a reset request causes the RCS bit to be set in the same register. The RCS and MECS events are provided for reasons of interrupt latency - there's only one register to read to determine the cause of the interrupt, and only one register to write to clear the interrupt.

8.3.1 Multicast-event Control Symbols

The SREP Physical Layer supports the exchange of multicast-event control symbols (MECS) between itself and the Internal Switch Port.

Note that for MECS to be exchanged, the PORT_OK bit in both the SREP Physical Layer and the Internal Switch Port must have a status of 1.

For more information on MECS, see Part 6 of the *RapidIO Interconnect Specification (Revision 1.3)*.

8.3.1.1 Sending MECS

The SREP Physical Layer supports the ability to send an MECS directly to the Internal Switch Port through the use of the "SREP Send Multicast-Event Control Symbol Register". When this register is written with the SEND field set to 1, the Internal Switch Port will receive a MECS event.

The Internal Switch Port can receive MECS as fast as the SREP Physical Layer can send them.

8.3.1.2 Interrupt Notification and Handling of MECS

Interrupt notification of MECS is most useful when a software-based entity needs to be notified of the MECS reception. To configure interrupt notification for the reception of MECS, the following must be completed:

1. Set the MECS_EN bit to 1 in the "SREP Interrupt Event Enable Register".

   When a MECS is received, the MECS bit in the "SREP Interrupt Status Register" is set to 1.

2. To clear the interrupt, Write 1 to the MECS bit in the "SREP Interrupt Status Register".

8.3.2 Reset Control Symbol Processing

The SREP Physical Layer supports the exchange of link-request/reset control symbols (reset) between itself and the Internal Switch Port.

Note that for reset control symbols to be exchanged, the PORT_OK bit in both the SREP Physical Layer and the Internal Switch Port must have a status of 1.

8.3.2.1 Sending a Reset Request

To send a reset from the SREP Physical Layer to the Internal Switch Port, write 0b011 to the CMD field of the "SREP Link Maintenance Request CSR".

8.3.2.2 Receiving a Reset Request

If the SELF_RST bit is 1 in the "SREP Mode CSR", then a "Bridge Reset" is performed. If the SELF_RST bit is 0 in the "SREP Mode CSR", then a reset is not performed immediately.

If the reset request should be ignored, then the RCS_EN bit in the "SREP Interrupt Event Enable Register" should be set to 0.

If software should be notified of the reset request using the RST_IRQ_b pin, then the RCS_EN bit in the "SREP Interrupt Event Enable Register" should be set to 1.

8.4 Data Integrity Checking

Data integrity checking is performed on both control symbols and packets. Additionally, error detection data is generated to ensure that data is not corrupted inside the SREP.

8.4.1 Packet Data Integrity Checking

Packets have two locations where CRC can occur. The first location is 80 bytes into the packet. The second location is at the end of the packet. Packets 80 bytes or smaller have only one CRC, packets larger than 80 bytes have two 16-bit CRC codes. The SREP checks the CRC of a packet only at the end of the packet.

Checking the packets CRC can be disabled by setting the ERR_DIS bit to 1 in the standard Port n Control CSR register, which resides in the MAC.

8.4.2 Data Path ECC

An Error Correction Code (ECC) is generated for every 8 byte quantity received from the link. If a packets size is not a multiple of 8 bytes, the ECC computed assumes that those bytes not part of the packet are 0.

The ECC is stored with the data every time the data is buffered. The ECC code is transferred in parallel with the 8 byte quantities, ensuring that single bit errors in the data stream are corrected, and double bit errors are detected.

ECC errors can be detected when a packet is being transferred to the Physical Layer. Data is updated to fix any correctable ECC errors detected before transmission. Uncorrectable ECC errors result in the packet being STOMPed and discarded.

ECC checking at the RapidIO physical layer is controlled by the "SREP ISF ECC Control Register". It is also possible to insert correctable and uncorrectable ECC errors in the SREP data stream using this register for test/verification purposes. For more information on Data Path ECC and how it affects Logical layer operation, see "ECC Error in Data Path".

8.5 RapidIO Physical Layer Flow Control

RapidIO flow control makes use of three concepts: priority based reordering, watermarks, and buffer release management.

The SREP Physical Layer supports flow control for packets exchanged between itself and the Internal Switch Port.

8.5.1 SREP Physical Layer Watermarks

The watermarks in the SREP Physical Layer are similar to those in the Tsi57x switches. Watermarks for transfer of packets from the Internal Switch Port into the SREP are located in the "SREP R2I Watermarks Register".

Flow control is generated for a given priority of traffic based on a programmable number of free buffers using watermarks. Watermarks can be programmed for priority 0, 1 and 2 packets. Priority 3 packets are accepted when there are free buffers. For information on the rules and examples for programming watermarks, see "Rules for Programming Watermarks".

This hierarchy of watermarks ensures that packets of lower priority cannot consume all buffers and prevent packets of higher priority from passing them. For example, if all buffers are filled, then at least one of the buffers must be occupied by a packet of priority 3. Since priority 3 is the highest priority in the system, the priority 3 packet should be given the first opportunity to make forward progress.

The default watermark values are 1 for priority 2, 2 for priority 1, and 3 for priority 0. This maximizes the number of buffers that can accept lower priority packets, which in turn maximizes the throughput of these packets.

In some systems, it is necessary to guarantee maximum throughput for a burst (continuous sequence) of packets at the same priority. In a congested system, it is possible that only one buffer is available for these packets. This can restrict throughput on the RapidIO transmit port, since while one packet in the burst is being transmitted and is awaiting acknowledgment, another packet in the burst cannot be accepted or transmitted. Watermarks can guarantee that two buffers are available for these packets. When two buffers are available, while one packet is transmitted and awaits acknowledgement another packet can be accepted. This leads to an increase in throughput for packets in the burst.

The Internal Switch Port and the SREP signals which priority of packets can be accepted based on watermarks. This allows the Internal Switch Port and the SREP to, at their discretion, select another packet for transmission that has a higher probability of being accepted.

8.5.1.1 Rules for Programming Watermarks

The following rules applied when are used in the SREP Physical Layer:

- No watermark is associated with Priority 3 packets
- The number of Free Buffers is computed as: Total Buffer - buffers occupied by RapidIO requests.
- A Priority x packet is accepted in the buffer if the number of Free Buffers is greater than the programmed watermark of the associated Priority (PRIOxWM). For example, when PRIO1WM is programmed to “3”, a Priority 1 packet is accepted only when there are 4 or more free buffers.
- The three programmed watermarks (PRIO0WM, PRIO1WM and PRIO2WM) have to contain values where PRIO0WM > PRIO1WM > PRIO2WM > 0 at all times.
- The watermarks have to be set according to the following rules:
  - PRIO2WM >= 1
  - PRIO1WM >= 2
  - PRIO0WM >= 3
  - PRIO2WM < PRIO1WM < PRIO0WM

Violating any one of the watermarks rules can create Deadlock situations in the system.

This hierarchy of watermarks ensures that packets of lower priority can never consume all buffers and prevent packets of higher priority from being transmitted. With the correct setting of the watermarks, the port must have received at least one Priority 3 packet if all buffers are full.

8.5.1.2 Generic Watermark Examples

Table 24: Examples of Use of Watermarks

Packet Buffers Available	Example 1: PRIO2WM = 1 PRIO1WM = 2 PRIO0WM = 3	Example 2: PRIO2WM = 2 PRIO1WM = 4 PRIO0WM = 5
	Packet Priority that can be Accepted	Packet Priority that can be Accepted
8	0, 1, 2, 3	0, 1, 2, 3
7	0, 1, 2, 3	0, 1, 2, 3
6	0, 1, 2, 3	0, 1, 2, 3
5	0, 1, 2, 3	1, 2, 3
4	0, 1, 2, 3	2, 3
3	1, 2, 3	2, 3
2	2, 3	3
1	3	3
0	None	None

Two examples are given in Table 8.5.1.2. The first example describes the default setting of the 3 watermarks. This maximizes the number of buffers that can accept lower priority packets, which in turn maximizes the throughput of these priorities. The second example describes a customized setting that favors the Priority 3 and 2 traffic at the expense of the throughput of Priority 1 and 0 packets.

8.5.2 Buffer Release Management

Buffer release management is a feature that allows the delays the reporting of free buffers. When the reporting of buffer releases is delayed, this has the effect of creating an opportunity for lower priority packets to be processed. Buffer release management can be used with watermarks to give lower priority packets an opportunity to make forward progress even under congested conditions.

Buffer release management does not guarantee that lower priority packets make forward progress in congested conditions. It just guarantees that they have an opportunity to do so.

Buffer release management is controlled by two registers, one for each direction of packet transfer. Buffer release management for transfers from the Internal Switch Port to the SREP Physical Layer are controlled by the “SREP R2I Buffer Release Control Register”. Transfers from SREP Physical Layer to the Internal Switch Port are controlled by “SREP I2R Buffer Release Control Register”.

The buffer release management registers have an identical programming model, which uses the following fields:

- REL_MGMT_EN – Enable or disable buffer release management
- REL_STOP – Stop reporting buffer releases when the number of free buffers reaches this number
- REL_RES – Resume reporting buffer releases when the number of free buffers reaches this number. REL_RES must be greater than REL_STOP.
- REL_TO – The maximum period in which buffer releases are not be reported. If this period expires, then buffer releases are reported depending on the setting of REL_TO_MODE.
- REL_TO_MODE – Determines when buffer release management becomes active after a timeout.

For example, suppose the priority 0, 1 and 2 RapidIO receive watermarks are set to their default values of 3, 2, and 1. The system designer requires that priority 0 packets be given the opportunity to progress even under high congestion scenarios. REL_STOP is set at 0, which stops free buffer reporting once all the buffers are filled. If REL_RES is set at 5, then free buffer reporting resume once five buffers are released. To free 5 buffers, at least 2 buffers that can hold priority 0 packets must be processed. Note that there are no guarantees that two priority 0 packets are processed, just that the opportunity exists for priority 0 packets to be received and processed.

To prevent unwarranted delays in high priority traffic due to buffer release management, the REL_TO field limits the amount of time buffer release management is active. In the previous example, the REL_TO field could be set to the time it takes to transmit 5 packets. This ensures that once buffer release management is activated, at most it delays packet transmission for the time required to transmit five maximum sized packets.

REL_TO_MODE is relevant under congestion scenarios when REL_TO has expired and the buffer fill level has not reached the REL_RES value. If REL_TO_MODE is 0, then buffer release management does not delay reporting of buffer fill levels until the buffer fill level reaches the REL_RES value. This allows higher priority traffic to get through without delay, which may starve lower priority traffic until the congestion eases. If REL_TO_MODE is 1, then buffer release management resumes delaying reportage of buffer fill levels and restarts the timeout period if the buffer fill level increases to the REL_STOP value. This delays higher priority traffic in favor of allowing lower priority traffic an opportunity to make forward progress. The mode chosen depends on the system performance characteristics required.

8.6 Physical Layer Debug Capabilities

Part 8 of the *RapidIO Interconnect Specification (Revision 1.3)* requires that the logical layer error information registers be writable to allow the creation of error conditions for software test purposes. For more information on these capabilities, see “Debug Support” and the *RapidIO Interconnect Specification (Revision 1.3)*.

9. SREP Transport Layer

Topics discussed include the following:

9.1 Overview

The SREP RapidIO Transport Layer interface has the following features:

- Supports 8- and 16-bit destination IDs
- Accepts a stream of packets from the Physical Layer, and creates a stream of packets which the Tsi620 RapidIO Logical Layer can accept.
- Filters packets based on packet type, destination ID, and source ID
- Supports 'promiscuous mode', where packets are accepted without checking the destination ID.
- Routes packets to the correct Logical Layer interface based on packet type and address
- Arbitrates between different Logical Layer sources of packets, and passes the stream of packets to the Physical layer
- Supports Part 3: Common Transport Layer Specification of the *RapidIO Interconnect Specification (Revision 1.3)* for an endpoint
- Supports Part 8: Error Management Extensions of the *RapidIO Interconnect Specification (Revision 1.3)* Transport Layer errors

9.2 8/16-bit Destination ID Support

The SREP Transport Layer supports reception of packets that have small (8-bit) and large (16-bit) destination IDs. The SREP Transport Layer allows the operation of mixed systems that make use of packets with both 8 and 16-bit destination IDs.

8-bit destination IDs are handled as distinct from 16-bit destination IDs. There are separate controls for 8 and 16-bit destination IDs.

The sourceID for all requests generated by the SREP come from the "SREP Base Device ID CSR". The sourceID/destinationID values for all responses are generated by swapping the sourceID/destinationID values of the received request packet.

Destination ID checking works off of two ranges - the Primary destination ID, programmed into the RapidIO standard register "SREP Base Device ID CSR", and the Secondary destination ID range, programmed using IDT specific registers. If destination ID checking is enabled, packets are accepted if they match either a Primary or Secondary range.

After the SREP is reset, it does not check the sourceID or destinationID of packets it receives. Packets from all sourceIDs and destinationIDs are accepted.

Packets that are not accepted according to the destination ID checking are discarded and an Illegal Target event is noted (see "Transport Layer Events").

9.2.1 Primary Destination ID Checking

The SREP can be programmed to check that the destination ID of each packet received matches the destination ID(s) programmed into the "SREP Base Device ID CSR".

To allow reception of any packet with an 8-bit destination ID, set the SM_TT_EN field to 0b00 in the "SREP Destination ID Checking Control Register". The setting of SMC_EN in the "SREP Destination ID Checking Control Register" has no effect when SM_TT_EN is 0b00.

To discard and note an error if any packet is received with an 8-bit destination ID, set the SM_TT_EN field to 0b01 in the "SREP Destination ID Checking Control Register". The setting of SMC_EN in the "SREP Destination ID Checking Control Register" has no effect when SM_TT_EN is 0b01.

To enable checking 8-bit destination IDs against the DEST_ID field of the "SREP Base Device ID CSR", set the SM_TT_EN field to 0b10 in the "SREP Destination ID Checking Control Register". When SM_TT_EN is 0b10, the setting of SMC_EN has an effect. For information on how the SMC_MASK and SMC_VALUE fields can be used to accept a range of 8-bit destination IDs, see "Secondary Destination ID Management".

To allow reception of any packet with a 16-bit destination ID, set the LG_TT_EN field to 0b00 in the "SREP Destination ID Checking Control Register". The setting of LMC_EN in the "SREP Destination ID Checking Control Register" has no effect when LG_TT_EN is 0b00.

To discard and note an error for any packet with a 16-bit destination ID, set the LG_TT_EN field to 0b01 in the "SREP Destination ID Checking Control Register". The setting of LMC_EN in the "SREP Destination ID Checking Control Register" has no effect when LG_TT_EN is 0b01.

To enable checking 16-bit destination IDs against the LG_DEST_ID field of the "SREP Base Device ID CSR", set the LG_TT_EN bit to 0b10 in the "SREP Destination ID Checking Control Register". When LG_TT_EN is 0b10, the setting of LMC_EN has an effect. For information on how the LMC_MASK and LMC_VALUE fields in the "SREP Large Secondary Destination ID Checking Control Register" can be used to accept a range of 16-bit destination IDs, see "Secondary Destination ID Management".

For a summary of the use of these control bits discussed, see Tables 25 and 26.

9.2.2 Secondary Destination ID Management

The SREP also allows checking of packet destIDs against ranges of destination IDs using a mask and value approach. If the bit-wise AND of the mask and the packets destination ID matches the value of the bit-wise AND of the mask and the value, then the packet is accepted.

For example, suppose that an endpoint should accept multicast packets with 16-bit destination IDs ranging from 0x1080 through 0x108F. The mask should be set to 0xFFF0, and the value should be set to 0x1080. In this way, any 16-bit destination ID whose most significant 12 bits are 0x108 is accepted.

To check 8-bit destination IDs against a mask and value, perform the following steps:

1. Set the SMC_MASK field of the "SREP Destination ID Checking Control Register" to have a 1 for every bit in the destination ID that should be checked.
2. Set the SMC_VALUE field of the "SREP Destination ID Checking Control Register" to have the required value which destination IDs should match.
3. Set the SMC_EN bit to 1 in the "SREP Destination ID Checking Control Register" to enable checking of 8-bit destination IDs.

To check 16-bit destination IDs against a mask and value, perform the following steps:

1. Set the LMC_MASK field of the "SREP Large Secondary Destination ID Checking Control Register" to have a 1 for every bit in the destination ID that should be checked.
2. Set the LMC_VALUE field of the "SREP Large Secondary Destination ID Checking Control Register" to have the required value which destination IDs should match.
3. Set the LMC_EN bit to 1 in the "SREP Destination ID Checking Control Register" to enable checking of 16-bit destination IDs.

Table 25: Relationship of SM_TT_EN and SMC_EN

SM_TT_EN	SMC_EN	Behavior
0b00	N/A	Accept all packets with 8-bit destination IDs
0b01	N/A	Do not accept any packets with 8-bit destination IDs.
0b10	0	Accept packets with an 8-bit destination ID that matches the value programmed in the "SREP Base Device ID CSR".
	1	• Accept packet with an 8-bit destination ID that matches the value programmed in the "SREP Base Device ID CSR". • Also accept packets with an 8-bit destination ID that matches the mask/value programmed in the "SREP Destination ID Checking Control Register".
0b11	N/A	Reserved. Do not accept any packets with 8-bit destination IDs.

Table 26: Relationship of LG_TT_EN and LGC_EN

LG_TT_EN	LMC_EN	Behavior
0b00	N/A	Accept all packets with 16-bit destination IDs
0b01	N/A	Do not accept any packets with 16-bit destination IDs.
0b10	0	Accept packets with a 16-bit destination ID that matches the value programmed in the "SREP Base Device ID CSR".
	1	• Accept packet with an 16-bit destination ID that matches the value programmed in the "SREP Base Device ID CSR". • Also accept packets with a 16-bit destination ID that matches the mask/value programmed in the "SREP Large Secondary Destination ID Checking Control Register".
0b11	N/A	Reserved. Do not accept any packets with 16-bit destination IDs.

The SREP can be configured to reject accesses destined for all destination IDs. In this state, the SREP does not respond to any transactions.

9.2.3 Destination ID Initialization

The "SREP Base Device ID CSR" is used as the source of all RapidIO requests issued by the SREP. This register is initialized based on the settings of two signals, SP_HOST and I2C_SA[6:0].

The following destination ID values are stated in the *RapidIO Interconnect Specification (Revision 1.2)*, Part 7 Interoperability Specification.

If SP_HOST is 0, this indicates that the SREP cannot act as a host in this system. The 8/16-bit destination ID should therefore be either 0xFF/0xFFFF or 0xFE/0xFFFE. I2C_SA[6] determines the least significant bit of the DEST_ID and LG_DEST_ID fields of the "SREP Base Device ID CSR".

If SP_HOST is 1, this indicates that the SREP will act as a host in this system. I2C_SA[6:0] determines the least significant 7 bits of the DEST_ID and LG_DEST_ID fields of the "SREP Base Device ID CSR". The most significant bits of these fields are initialized to 0.

The I2C_SA[6:0] value can be used as an input to software to determine what the destination ID should be for the endpoint.

9.3 Register Access Source ID Management

In reliable systems, it is necessary to restrict register access to a control processor in order to prevent faults in the system from propagating. The SREP supports a filtering function similar to that used for secondary destination ID management to enforce restrictions on the source ID of register write transactions. The filtering function is used in three registers:

- "SREP Register Access Source ID Checking Control Register"
- "SREP Register Access Small Source ID Checking Control Register"
- "SREP Large Register Access Source ID Checking Register"

These three registers are writable from any RapidIO source ID.

Register accesses must also be accepted based on the destination ID checking described in "8/16-bit Destination ID Support".

Similar to the secondary destination ID filtering function, the register access source ID filtering function allows separate filtering for 8 and 16-bit source IDs. All or none of the 8- or 16-bit source IDs can be accepted, based on the setting of the SREG_CTL and LREG_CTL fields in the "SREP Register Access Source ID Checking Control Register".

To select a range of 8-bit source IDs to be accepted, perform the following steps:

1. Set "SREP Register Access Small Source ID Checking Control Register".SREG_MASK to have a 1 for every bit which should be checked against the value in "SREP Register Access Small Source ID Checking Control Register".SREG_VALUE.
2. Set "SREP Register Access Small Source ID Checking Control Register".SREG_VALUE to the correct value to be checked.
3. Set the "SREP Register Access Source ID Checking Control Register".SREG_CTL to be 'S8'.

When SREG_MASK is 0, this corresponds to allowing all 8-bit source IDs to write to registers.

Similarly, to check a range of 16-bit source IDs, set the bit fields in the "SREP Register Access Small Source ID Checking Control Register" according to the following:

1. Set "SREP Large Register Access Source ID Checking Register".LREG_MASK to have a 1 for every bit which should be checked against the value in "SREP Large Register Access Source ID Checking Register".LREG_VALUE.
2. Set "SREP Large Register Access Source ID Checking Register".LREG_VALUE to the correct value to be checked.
3. Set "SREP Register Access Source ID Checking Control Register".LREG_CTL to be 'S6'.

When LREG_MASK is 0, this corresponds to allowing all 16-bit source IDs to write to registers.

9.4 Packet Transfer and Filtering

The SREP Transport Layer transfers packets to different logical layer queues based on their packet type. Maintenance Read Requests and Maintenance Write Requests are sent to the Register Request Queue. All NREAD, NWRITE and NWRITE_R requests are sent to the Logical Layer, where the address is checked against the "SREP Local Configuration Space Base Address CSR". The NREAD/NWRITE/NWRITE_R requests are sent to the Register Request Queue if the address matches the address in the "SREP Local Configuration Space Base Address CSR". If the address does not match the address in the "SREP Local Configuration Space Base Address CSR", the NREAD/NWRITE/NWRITE_R requests are routed to the R2I Data Buffers/Header Queues.

All response packets (Maintenance Read Response, Maintenance Write Response, NREAD response, NWRITE_R response) are routed to the R2I Data Buffers/Header Queues for handling. The arrival of responses is communicated to the I2R Request Queue to confirm that the transaction is completed. This packet routing is captured in Table 27.

Table 27: Routing of Packets to Logical Layer Handlers

Packet Type	Transaction Type (TTYPE)	Register Request Queue/ Data Buffers	R2I Data Buffers/ Header Queue	Port-Write Buffer	Doorbell Buffers	I2R Header Queue
FTYPE 2	NREAD (4)	X	X			
FTYPE 5	NWRITE (4) NWRITE_R (5)	X X	X X			
FTYPE 6	SWRITE (N/A)		X			
FTYPE 8	Read Request (0) Write Request (1) Port-Write Request (4)	X X		X		
	Read Response (2) Write Response (3)		X			X X
FTYPE 10	Doorbell Request				X	
FTYPE 13	Response, no payload (0) Response, payload (8)		X			X X

9.5 Bridge ISF to RapidIO Arbitration and Flow Control

The Logical I/O and Register Transaction layers present 4 sources of packets to the I2R transport layer. These sources are:

- I2R Request Queue
- Register Response Queue

- Port-Write Queue
- R2R Queue

The transport layer arbitrates between these different packet sources, and passing the packets to the Physical Layer Retry buffers. The transport layer must select packets to be passed to the Physical Layer Retry buffers that meet the watermarks and buffer release management requirements (see “RapidIO Physical Layer Flow Control”).

The minimum packet priority that the Retry Buffers can accept is communicated to the different packet sources. Each packet source requests transmission of the oldest packet that meets the minimum packet priority requirement of the Retry Buffers. The transport layer performs round robin arbitration among the different packet sources requesting transmission.

9.6 Transport Layer Events

There are few transport layer events in the SREP since there is little transport layer functionality. Multiple transport layer errors can occur within the same packet. They have event precedence as follows:

1. Packet Stomped
2. R2I Packet CRC Error
3. Illegal Target
4. Packet TTL Expired

9.6.1 Illegal Target Event

An Illegal Target event occurs when a RapidIO packet is received for a destination ID which the SREP is configured to reject.

For more information on control of acceptable destination IDs, see “8/16-bit Destination ID Support” and “Register Access Source ID Management”.

The Illegal Target packet is dropped. Information about the Illegal Target packet is captured in the Logical/Transport Layer error information registers. For more information on the behavior of the Logical/Transport Layer error information registers, see “RapidIO Logical/Transport Error Information Registers”.

9.6.2 R2I Stomped Packet Event

A Stomped Packet event occurs when a RapidIO packet being transferred from the physical layer to the transport/logical layer has an ‘STOMP’ status associated with it.

The physical layer can pass packets through to the transport/logical layer before the packets CRC is checked. Additionally, as part of the RapidIO transmission protocol, a packet may be STOMPed at any time by the transmitter. Either a bad CRC value, or a STOMP request, may be received by the physical layer as a packet is being transferred from the Physical layer to the Transport/Logical layer. As part of finishing the transfer of the packet, the Physical layer indicates that the packet being transferred should be STOMPed.

The physical layer transfers packets without dropping packets, including automatic retransmission attempts in the case of CRC corruption or packet STOMPing on the physical link. For this reason, the STOMPed packet is dropped by the transport layer. Information about the Stomped packet is captured in the Logical/Transport Layer error information registers. For more information on the behavior of the Logical/Transport Layer error information registers, see “RapidIO Logical/Transport Error Information Registers”.

9.6.3 R2I CRC Error Event

An R2I CRC Error event occurs when the CRC of a RapidIO packet being received from the physical layer is incorrect as checked by the Transport/Logical layer.

The physical layer can pass packets through to the transport/logical layer before the packets CRC is checked. A bad CRC value may therefore be passed through from the Physical layer to the Transport/Logical layer. If the bad CRC was detected at the physical layer, the physical layer STOMPs the packet.

If a CRC error is detected, but the physical layer has not STOMPed the packet, this indicates that the packet was corrupted during the transfer from the Physical layer to the Transport/Logical layer. In this case, the packet is discarded.

The packet is not retransmitted from the Physical layer to the Transport/Logical layer. When the packet is discarded by the Transport/Logical Layer, the packet is lost to the system.

Information about the corrupted packet is captured in the Logical/Transport Layer error information registers. For more information on the behavior of the Logical/Transport Layer error information registers, see “RapidIO Logical/Transport Error Information Registers”.

9.6.4 Packet TTL Expired

A Packet TTL event occurs when the SREP receives a packet from the physical layer whose time-to-live period has expired.

RapidIO Switches enforce a “time to live” on packets to allow system designers to engineer end-to-end timeouts in systems. It is therefore possible for the SREP to receive a packet whose time-to-live interval has expired from the physical layer of devices which incorporate the SREP.

When a packet whose time-to-live period has expired is received by the SREP, the packet is discarded. No error response is generated. Information about the packet discard is captured in the Physical layer registers of the MAC attached to the SREP. No status information is latched by the SREP.

10. SREP Logical Interface

Topics discussed include the following:

10.1 Overview

The SREP Logical I/O block has the following features for sending Bridge ISF transactions to RapidIO:

- Supports the following Bridge ISF commands as an Bridge ISF target:
  - Memory Write Burst
  - Memory Write Block
  - Memory Read Block
  - Memory Read Word
  - SYNC
- Supports all Bridge ISF byte lane combinations for Memory Write Burst and Memory Read Word
- Supports Segmentation And Reassembly (SARing) of Bridge ISF requests up to 128 data phases (1024 bytes or less, subject to alignment constraints) in size.
- Bridge ISF transactions are sent to RapidIO on the basis of address ranges controlled by a Doorbell BAR and a maximum of 8 Bridge ISF-to-RapidIO BARs with 256 Bridge ISF-to-RapidIO Lookup Table (LUT) entries
- BARs can overlap with each other. BARs are decoded in a strict priority order to determine which BAR controls what address range.

- Each Bridge ISF-to-RapidIO BAR can be assigned a configurable number of I2R Lookup Table entries
- I2R LUTs can be shared between I2R BARs
- Supports origination of NREAD, NWRITE, NWRITE_R, and SWRITE RapidIO Logical I/O transactions
- Supports origination of DOORBELL transactions
- Supports origination of 4 byte RapidIO Maintenance Read and Maintenance Write transactions
- Bridges RapidIO responses with DONE and ERROR status to Bridge ISF for NREAD transactions
- Supports RapidIO responses with DONE, ERROR and RETRY status for DOORBELL transactions.

The SREP Logical I/O block has the following features for sending RapidIO transactions to the Bridge ISF:

- Allows SREP registers, as defined by "Register Map", to be accessed through RapidIO Maintenance Read/Write Transactions using 4 byte-aligned transactions which are 4 bytes in size.
- Allows SREP and Tsi620 registers to be accessed through RapidIO NREAD, NWRITE and NWRITE_R packets using 4-byte aligned transactions which are 4 bytes in size or less.
- Supports up to 32 outstanding register requests
- Accepts Maintenance read and register access NREAD accesses from RapidIO regardless of sourceID.
- Supports reception of NREAD, NWRITE, NWRITE_R, and SWRITE RapidIO Logical I/O transactions
- Supports all payload sizes for NREAD, NWRITE, NWRITE_R and SWRITE defined the *RapidIO Interconnect Specification (Revision 1.3)*
- Processes RapidIO to Bridge ISF transactions on the basis of address ranges controlled by the RapidIO standard Register BAR and a maximum of 8 RapidIO-to-Bridge ISF BARs with 256 RapidIO-to-Bridge ISF Lookup Table (R2I LUT) entries
- BARs can overlap with each other. BARs are decoded in a strict priority order to determine which BAR controls what address range.
- R2I LUTs can be shared between R2I BARs
- Supports the following Bridge ISF commands as a Bridge ISF source:
  - Memory Write Block
  - Memory Read Block
  - SYNC
- Supports buffering of two port-write packets, each up to 32 bytes in size
- Supports reception and buffering of up to 32 DOORBELL transactions

The SREP Logical I/O block has the following RapidIO features:

- Supports 256 outstanding RapidIO transaction IDs

- RapidIO packets originated can have the CRF bit set
- RapidIO packets received can have the CRF bit set. The CRF bit does not affect the operation of the SREP Logical I/O block.
- Supports 34-bit RapidIO addressing
- Supports Part 1: Logical I/O Transactions of the *RapidIO Interconnect Specification (Revision 1.3)*
- Supports Part 2: Message Passing Logical Specification of the *RapidIO Interconnect Specification (Revision 1.3)*
- Supports Part 8: Error Management Extensions of the *RapidIO Interconnect Specification (Revision 1.3)*
- Supports transmission of port-write packets in accordance with Part 8: Error Management Extensions of the *RapidIO Interconnect Specification (Revision 1.3)*
- Transport/Logical I/O functionality can operate at speeds up to 156.25 MHz. This is sufficient to support all RapidIO link speeds and widths up to 4x at 3.125 Gbaud.

The SREP Logical I/O block has the following Bridge ISF features:

- Bridge ISF Interface can operate at speeds up to 156.25 MHz
- Bridge ISF Interface is synchronous with RapidIO Logical functionality

The SREP Logical I/O block has the following Data Integrity features:

- BARs and LUTs are protected by parity
- An end-to-end ECC protection scheme ensures that data is not corrupted during transfers between RapidIO and the Bridge ISF.
- Allows users to place an upper bound on the amount of time a transaction can exist within the SREP. This ensures that late responses cannot be matched with an incorrect request.

10.1.0.1 Features Not Supported

The SREP RapidIO Interface does not support the following RapidIO features:

- Reception or generation of RapidIO Logical I/O ATOMIC operations
- Register accesses of any size greater than 4 bytes using RapidIO Maintenance Read/Write Transactions, or RapidIO NREAD, NWRITE and NWRITE_R packets
- Generation or reception of RapidIO Message, Flow Control or Data Streaming transactions

10.2 RapidIO Logical Layer Protocol Support

The SREP supports transmission and reception of NREAD, NWRITE, NWRITE_R and SWRITE logical layer packets for memory-mapped input/output purposes. In this documentation, the NREAD, NWRITE, NWRITE_R and SWRITE packets are collectively called the Logical I/O packets. DOORBELL packets can also be sourced and received by the SREP.

The SREP supports all valid payload sizes for Logical I/O packets. Logical I/O packets all require that their payloads be a multiple of 8 bytes. NREAD, NWRITE, and NWRITE_R packets allow less than 8 bytes to be transferred in a single packet. The RapidIO specification restricts which bytes within the 8 byte payload can be valid. Valid byte lane combinations are documented in Table 28. In this table, an 'X' represents which byte(s) can be transferred.

Table 28: Valid NREAD, NWRITE, and NWRITE_R Byte Lane Combinations

Number of Bytes	0	1	2	3	4	5	6	7
1	X							
1		X						
1			X					
1				X				
1					X			
1						X		
1							X	
1								X
2	X	X						
2			X	X				
2					X	X		
2							X	X
3	X	X	X					
3						X	X	X
4	X	X	X	X				
4					X	X	X	X
5	X	X	X	X	X			
5				X	X	X	X	X
6	X	X	X	X	X	X		
6			X	X	X	X	X	X
7	X	X	X	X	X	X	X	
7		X	X	X	X	X	X	X

The operation of each of these packet types is described in the following sections.

10.2.1 Maintenance Transactions

The SREP can receive and originate RapidIO Maintenance transactions. All possible Maintenance transactions (Read, Write, Read Response, Write Response, Port-Write) are supported.

Maintenance Read and Write transactions require a response from the RapidIO link partner. Maintenance Write responses do not require a response to be sent over the Bridge ISF.

Port-Writes may be originated by the SREP, and do not require responses. Two Port-Writes received by the SREP will be buffered.

10.2.2 NWRITE

The SREP can receive and originate RapidIO NWRITE requests. All possible NWRITE sizes are supported.

NWRITE packets do not require logical layer responses from the receiver. Therefore, once an NWRITE packet is acknowledged by the RapidIO link partner, the NWRITE packet is discarded from SREP's RapidIO buffers. The SREP does not need to compose a response for an NWRITE request.

Similarly, when an NWRITE packet is transferred over the Bridge ISF, the buffer associated with the NWRITE is cleared.

10.2.3 NWRITE_R

The SREP can receive and originate RapidIO NWRITE_R requests. All valid sizes of NWRITE_R requests are supported.

NWRITE_R packets require logical layer responses from the receiver. The response must be received within the timeout period controlled by the "SREP Response Timeout Control CSR". For more information on RapidIO Response Timeout Events, see "Logical I/O Packet Response Timeout Events".

The SREP sends an NWRITE_R response for an NWRITE_R request it receives once the NWRITE_R is transferred over the Bridge ISF or to the register response queue. For this reason, a response with ERROR status for an NWRITE_R request is sent only when an error is detected for the NWRITE_R request within the SREP. If the NWRITE_R request is successfully bridged and transferred over the Bridge ISF, a response with DONE status is sent. Errors may occur during further processing of the NWRITE_R bridged transaction, which causes an error notification after the RapidIO response transaction is sent.

The NWRITE_R request is discarded once the NWRITE_R request is transferred.

The priority of a response packet for an NWRITE_R request is one more than the priority of the original request.

10.2.4 S_WRITE

The SREP can receive and originate RapidIO SWRITE requests. All SWRITE requests are a multiple of 8 bytes.

As with an NWRITE packet, SWRITE packets do not require logical layer responses from the receiver. Therefore, once an SWRITE packet is acknowledged by the RapidIO link partner, the SWRITE packet is discarded from the SREP's RapidIO buffers. For information on when SWRITE packets can be generated, see "Bridge ISF Memory Write Block Segmentation".

Similarly, once the SWRITE packet received from the RapidIO link partner is transferred over the Bridge ISF, the SWRITE packet can be discarded.

10.2.5 NREAD

The SREP can receive and originate RapidIO NREAD requests. All possible NREAD sizes are supported.

NREAD requests require that a response packet be sent. The SREP discards an NREAD request that it originated only when it received the matching logical layer response packet. This is true for NREAD requests received from the Bridge ISF and from RapidIO. The response must be received within the timeout period controlled by the "SREP Response Timeout Control CSR". For more information on RapidIO Response Timeout Events, see "Logical I/O Packet Response Timeout Events".

The RapidIO priority of a response packet for an NREAD request is one more than the priority of the original request.

10.2.6 DOORBELLs

The SREP can receive and originate RapidIO DOORBELL requests.

DOORBELL transactions are originated from Bridge ISF write transactions. DOORBELLs require a response packet to be received before the request can be judged complete. DOORBELL response packets support a RETRY status, which is a logical layer request to transmit the DOORBELL again. The SREP supports retransmission of DOORBELL transactions on receipt of a response packet with a RETRY status.

DOORBELL transactions can be received from RapidIO. Doorbell requests are buffered until they can be serviced by software. For more information on the reception of DOORBELL requests, see "RapidIO Doorbell Request Handling".

10.3 Register Access Support

The SREP registers can be accessed using Maintenance Read and Write transactions from RapidIO. The SREP registers can also be accessed using NREAD, NWRITE and NWRITE_R transactions when the registers are memory mapped. Additionally, SREP registers can be accessed from PCI when PCI BAR0 is enabled and configured.

10.3.1 Maintenance Read and Write Support

The SREP Register Transaction block can receive RapidIO Maintenance Read and Write requests.

The SREP Register Transaction block supports Maintenance requests to SREP registers that are 4 bytes in size. Maintenance Read and Write requests that are not 4 bytes causes an Illegal Register Access event to be asserted. For more information on Illegal Register Access events, see "RapidIO Illegal Register Access" and "RapidIO Unsupported Transaction Events".

The SREP processes the Maintenance Read or Write received against registers contained in the SREP.

Maintenance Read or Write requests cannot access registers in other Tsi620 blocks. To access registers in other blocks, use NREAD/NWRITE/NWRITE_R packets.

Maintenance Read and Write transactions are processed in the order in which they are received.

On completion of the register access, the SREP sends an appropriate maintenance read and write response packet.

The priority of a response packet for a Maintenance Read/Write request is ONE more than the priority of the original request.

Reads of registers that are reserved in the SREP memory map return 0. Writes to registers that are reserved in the SREP memory map complete without error.

Maintenance read accesses are not subject to the sourceID restrictions described in "Register Access Source ID Management". They are, however, subject to the destinationID restrictions described in "8/16-bit Destination ID Support".

Maintenance write accesses are subject to the sourceID and destinationID restrictions described in "8/16-bit Destination ID Support" and "Register Access Source ID Management".

10.3.2 Accessing Registers Using Logical I/O Transactions

The registers are mapped to a location in the 34 bit RapidIO Logical I/O address space using the "SREP Local Configuration Space Base Address CSR"..

When the "SREP Local Configuration Space Base Address CSR" overlaps with the address space controlled by an R2I BAR, the "SREP Local Configuration Space Base Address CSR" takes precedence. The size of the address space controlled by the LCS BAR is 256 KB for Tsi620.

The "SREP Local Configuration Space Base Address CSR" is disabled after the SREP is reset. It can be enabled in two ways:

- By writing to the "SREP Local Configuration Space Base Address CSR". This clears the LCS_DIS bit to 0 in the "SREP R2I RapidIO Miscellaneous Control CSR"
- By setting to the LCS_DIS bit to 0 in the "SREP R2I RapidIO Miscellaneous Control CSR"

The "SREP Local Configuration Space Base Address CSR" can be disabled by writing 1 to the LCS_DIS bit of the "SREP R2I RapidIO Miscellaneous Control CSR".

The SREP registers are compacted into a 4 KB contiguous memory block when accessed using NREAD/NWRITE/NWRITE_R transactions in the Tsi620 memory map (see "Register Map").

To access registers, the SREP Logical Layer checks the address of RapidIO NREAD, NWRITE, and NWRITE_R transactions against the address configured in the "SREP Local Configuration Space Base Address CSR". If the address matches the address space supported, then the transaction is serviced using the Register Request Queue/Data Buffers.

Logical I/O register requests (NREAD/NWRITE/NWRITE_R) are subject to the destinationID restrictions described in "8/16-bit Destination ID Support".

Logical I/O NWRITE and NWRITE_R register requests are also subject to the sourceID restrictions described in "Register Access Source ID Management"

The SREP supports NREAD/NWRITE/NWRITE_R requests to SREP and Tsi620 registers that are 4 bytes or less, and aligned to a 4-byte address boundary. Transactions that are greater than 4 bytes result in an Illegal Register Access event being detected.

RapidIO SWRITE transactions cannot access SREP registers because SWRITE transactions are a multiple of 8 bytes.

The SREP does not check that it received a multiple of 8 bytes for an SWRITE. All SWRITEs are assumed to be at least 8 bytes (see "RapidIO Illegal Transaction Events").

Registers in Tsi620 are defined in little-endian format. However, RapidIO is a big-endian protocol. All RapidIO register accesses are handled in big-endian format when received from RapidIO. The SREP register bus slave byte-word swaps all register access requests, and performs appropriate address munging. The SREP register bus master byte-word swaps all register access data that it sends/receives, and performs appropriate address munging.

10.3.3 Restrictions on Register Access

Some registers in the SREP must be accessed 4 bytes at a time when accessed using NREAD/NWRITE/NWRITE_R. These are registers are normally protected by parity. They include the following:

- "SREP Local Configuration Space Base Address CSR"
- "SREP R2I Base Address Register x LUT Control CSR"
- "SREP R2I Base Address Register x Lower"
- "SREP R2I Upper LUT Entry Translation Address Register"
- "SREP R2I Lower LUT Entry Translation Address Register"
- "SREP I2R Base Address Register x LUT Entry CSR"
- "SREP I2R Base Address Register x Upper"

- "SREP I2R Base Address Register x Lower"
- "SREP I2R Doorbell BAR Upper"
- "SREP I2R Doorbell BAR Lower"
- "SREP I2R Upper LUT Entry Translation Register"
- "SREP I2R Lower LUT Entry Translation Address Register"
- "SREP I2R LUT Translation Parameters Register"

10.4 Bridging Logical I/O Requests to the Bridge ISF

The SREP's main function is to convert RapidIO packets to Bridge ISF transactions, and vice versa. This section describes converting RapidIO packets to Bridge ISF transactions (see "Bridging ISF Requests to RapidIO").

NREAD, NWRITE, and NWRITE_R packets can access registers (see "Register Access Support") instead of being sent to Bridge ISF.

Doorbell requests may also be received. Doorbell requests are not sent to Bridge ISF; they are stored in their own buffer (see "RapidIO Doorbell Request Handling").

RapidIO Logical I/O reads and writes are converted to Bridge ISF read and write transactions. The bridging operation uses the RapidIO Address to control the following attributes of the Bridge ISF transaction:

- Address translation between RapidIO 34 bit address space and Bridge ISF 64-bit address space
- Enforcement of read and write access privileges
- Routing of the RapidIO request to the correct Bridge ISF port

Packets with the Critical Request Flow (CRF) bit set are processed no differently from packets with the CRF bit cleared.

Table 29 defines the translation between RapidIO transactions and Bridge ISF transactions. Several kinds of the RapidIO transactions received cannot be converted to Bridge ISF transactions, including Maintenance Read and Write transactions, Maintenance Port-Write transactions, and Doorbells. The handling of Maintenance Read and Write transactions is described in "Maintenance Read and Write Support". Maintenance Port-Write transaction handling is described in "Maintenance Port-Write Support". The handling of Doorbell transactions is described in "RapidIO Doorbell Request Handling".

Table 29: RapidIO Packet Mapping to Bridge ISF Transaction Types

RapidIO Transaction	RapidIO Ftype	RapidIO Transaction Type	Bridge ISF Transaction
Maintenance Read Request	0x8	0x0	N/A
Maintenance Write Request	0x8	0x1	N/A
Maintenance Read Response	0x8	0x2	Bridge ISF Response with Data, or Bridge ISF Error Response
Maintenance Write Response	0x8	0x3	N/A
Maintenance Port-Write	0x8	0x4	N/A
Logical I/O NREAD	0x2	0x4	Bridge ISF Memory Read Block
Logical I/O NWRITE	0x5	0x4	Bridge ISF Write Block
Logical I/O NWRITE_R	0x5	0x5	Bridge ISF Write Block
Logical I/O SWRITE	0x6	N/A	Bridge ISF Write Block
Doorbell	0xA	N/A	N/A
Response without Data	0xD	0x0	Bridge ISF Error Response for Bridge ISF Read Block, Bridge ISF Read Burst requests. Note: No Bridge ISF response is associated with Bridge ISF transactions that are translated to NWRITE_R packets
Response with Data	0xD	0x8	Bridge ISF Response with Data for Bridge ISF Read Block, Bridge ISF Read Burst requests

10.4.1 Base Address Register (BARs) and Lookup Table (LUT) Operation

The SREP selects which RapidIO addresses it supports based on the settings of the Logical I/O Base Address Register and the Local Configuration Space Base Address Register (LCS BAR) (“SREP Local Configuration Space Base Address CSR”).

- The 8 Logical I/O Base Address Register sets, each collectively called an R2I BAR, define RapidIO address ranges for which Logical I/O packets are accepted.
- The LCS BAR defines the base address of the RapidIO address range that accesses Tsi620 registers using RapidIO Logical I/O transactions. For more information on accessing Tsi620 registers from RapidIO, see “Accessing Registers Using Logical I/O Transactions”.

The address ranges controlled by R2I BARs and LCSR BAR can overlap with each other. The overlap is resolved as follows:

- Logical I/O transactions with addresses that fall within the LCS BAR are handled as register accesses to the Tsi620 registers.
- The lowest numbered R2I BAR into which the address of a Logical I/O transaction falls determines the translation of the RapidIO transaction, if any, to a Bridge ISF transaction.

Parity errors in the LCS BAR or any of the R2I BARs, stop the processing of NREAD/NWRITE/NWRITE_R requests by the SREP (see “RapidIO-to-Bridge ISF Parity Error”).

If the LCS BAR or any of the R2I BARs has a parity error, decoding of Logical I/O accesses is undefined.

Only Maintenance transactions are guaranteed to access register space when parity errors are detected in the LCS BAR or any of the R2I BARs (see “RapidIO-to-Bridge ISF Parity Error”).

RapidIO requests with addresses that fall outside of all enabled BARs causes an ‘Out of Bounds Request’ event to be detected (see “RapidIO OOB Request Event”).

The minimum size of each R2I BAR is 4 KB. The maximum size of each R2I BAR is 16 GB.

Each R2I BAR is divided into a configurable number of equally sized LUT entries. The size of each LUT entry is a power of 2, starting with 4 KB. The attributes that translate Logical I/O requests to Bridge ISF requests are associated with a LUT entry. There are 256 LUT entries for the RapidIO-to-Bridge ISF direction, organized as a table.

The smallest LUT entry size supported is 4 KB. The maximum LUT entry size supported is 1 GB.

Each BAR has the following fields:

- Bar Enable – Controls whether or not the BAR is active
- Address – the start of the RapidIO address range in which the BAR accepts a Logical I/O request
- Size – a value specifying the size of the RapidIO address range supported by this BAR

- LUT Index – This is the index of the first LUT entry that controls the translation of Logical I/O transactions to Bridge ISF transactions.
- Num LUTs – The number of LUT entries, beginning with the LUT Index, into which the BAR address range is divided.

To meet the restrictions on RapidIO-to-Bridge ISF LUT size, the following must be true:

- Size >= Num LUTs (LUT size must be at least 4 KB)
- Size <= 18 + Num LUTs (LUT size must be less than or equal to 1 GB)
- Num LUTS <= 8 (cannot use more LUTs than exist)

The starting address of the BAR is a multiple of the Size.

BAR address bits that would otherwise cause the BAR to begin at an address that is not a multiple of Size are ignored.

For example, suppose that the SREP must be programmed to accept Logical I/O transactions in the RapidIO address range of 0x1_0000_0000 to 0x1_2000_0000. The transactions access 16 different bounded buffers, each at its own address within the Bridge ISF address space. For this example, BAR 1 and 16 LUT entries starting at index 128 are arbitrarily chosen to use this configuration requirement.

- Set "SREP R2I Base Address Register x LUT Control CSR" to 0x0080_1104
- Set "SREP R2I Base Address Register x Lower" to 0x0000_0401

It is a programming error if "SREP R2I Base Address Register x LUT Control CSR".FIRST_LUT, added to "SREP R2I Base Address Register x LUT Control CSR".NUM_LUTS, is greater than 255.

The operation of the SREP is undefined when this programming error is present.

Each LUT entry controls the following attributes:

- Bridge ISF Translation Address – This is the first 64-bit Bridge ISF translation address in the block of addresses controlled by the LUT entry that the RapidIO address should be translated to.
- Bridge ISF PORT – The Bridge ISF port number that transactions should be sent to
- RD_EN – control of read access privilege on this LUT
- WR_EN – control of write access privilege on this LUT

For example, suppose that LUT table entry 0x9C should be programmed to allow read and write access at a Bridge ISF address of 0x1234_5678_0000_0000 for Bridge ISF port number, ISF_PORT = 0b1. The following registers and values produce the mapping described above:

- Set "SREP R2I LUT and Parity Control Register" to 0x0000_009C
- Set "SREP R2I Upper LUT Entry Translation Address Register" to 0x1234_5678

- Set "SREP R2I Lower LUT Entry Translation Address Register" to 0x0000_0013

Note that if multiple LUT entries are to be programmed or read sequentially, the Auto-increment feature of the "SREP R2I LUT and Parity Control Register" can be used to avoid setting the LUT index for each LUT entry.

If the AUTO_INC bit in the "SREP R2I LUT and Parity Control Register" is set to 1, the LUT_IDX value increments by 1 when the "SREP R2I Lower LUT Entry Translation Address Register" is read or written. When the "SREP R2I Lower LUT Entry Translation Address Register" is read or written and the LUT_IDX value is 0xFF, the LUT_IDX value wraps to 0.

LUT entries can be associated with more than one BAR.

There are some additional controls in the R2I LUT registers that are not directly related to the Bridge ISF:

- "SREP R2I Lower LUT Entry Translation Address Register".RD_EN - If this bit is 0, Bridge ISF Reads to this LUT entry cause an error (see "RapidIO Read Denied Event").
- "SREP R2I Lower LUT Entry Translation Address Register".WR_EN - If this bit is 0, Bridge ISF Writes to this LUT entry cause an error (see "RapidIO Write Denied Event").

Translation of RapidIO requests to Bridge ISF transactions, and receiving RapidIO responses, depend on the availability of Bridge ISF resources. Status bits in the "SREP R2I Event Status Register" indicate the availability of these resources, as follows.

- NO_BISF_TID is 1 when all Bridge ISF transaction IDs are in use. Bridge ISF transaction IDs are needed when a RapidIO transaction is translated to a Bridge ISF transaction that requires a response (Read). RapidIO transactions that do not require Bridge ISF responses (Writes) can continue to be processed when all Bridge ISF transaction IDs are in use.

10.4.1.1 RapidIO-to-Bridge ISF Global Controls

The only global control parameter that affects RapidIO-to-Bridge ISF transactions is Bridge ISF transaction priority.

Bridge ISF transaction priority for transactions sent from RapidIO to the Bridge ISF is set globally in the "SREP R2I ISF Request Priority Control Register" and the "SREP R2I ISF Response Priority Control Register".

10.4.1.2 RapidIO to Bridge ISF Address Translation

The Bridge ISF address used for a RapidIO packet depends on combining the RapidIO address specified by the packet, with the Bridge ISF Translation Address specified by the LUT entry. The portion of the RapidIO address and the Bridge ISF Translation Address used depends on the size of the LUT entry. The lower part of the Bridge ISF address is the offset of the RapidIO address within the LUT entry. The Bridge ISF address bits that are outside of the LUT entry form the remainder of the address.

For example, suppose that RapidIO address 0x1_2345_6780 falls in a 64-KB LUT entry. The Bridge ISF Translation Address for this LUT entry is specified as 0x FEDC_BA98_7654_3000. The Bridge ISF address used for this transaction is 0xFEDC_BA98_7654_6780. The least significant 16 bits of the RapidIO address are used as the lower part of the Bridge ISF since this is the offset within the 64 KB LUT entry. The upper 48 bits of the Bridge ISF translation address are used, since these are outside of the LUT entry.

10.4.2 RapidIO to Bridge ISF Packet Segmentation and Reassembly

All RapidIO packets fit into one Bridge ISF transaction. A RapidIO request packet maps to one Bridge ISF transaction.

10.4.3 RapidIO Request Packet Values in Responses

RapidIO request packets that require responses have two fields in them, the destination ID and source ID, which identify where the packet should be sent to and where a response should be sent. The SREP swaps the destination ID and the source ID of the request packet when composing a RapidIO response packet.

RapidIO request packets received by the SREP have a Critical Request Flow (CRF) bit in them which may or may not be set. The response packet has the same CRF bit setting that the original request had.

10.5 Bridging ISF Requests to RapidIO

Bridge ISF requests are either reads or writes, with a maximum data size of 128 Bridge ISF data phases (1024 bytes or less). The Bridge ISF requests can be mapped to different RapidIO packet types, as shown in Table 30.

Table 30: Bridge ISF Transaction Mapping to RapidIO Packet Types

Bridge ISF Transaction	RapidIO Transaction	RapidIO Ftype	RapidIO Transaction Type
Bridge ISF Memory Read Word	Maintenance Read	0x8	0x0
	Logical I/O NREAD	0x2	0x4
Bridge ISF Memory Read Block	Maintenance Read	0x8	0x0
	Logical I/O NREAD	0x2	0x4
Bridge ISF Write Burst	Maintenance Write	0x8	0x1
	Logical I/O NWRITE	0x5	0x4
	Logical I/O NWRITE_R	0x5	0x5
	Doorbell	0xA	N/A

Table 30: Bridge ISF Transaction Mapping to RapidIO Packet Types (Continued)

Bridge ISF Transaction	RapidIO Transaction	RapidIO Ftype	RapidIO Transaction Type
Bridge ISF Write Block	Maintenance Write	0x8	0x1
	Logical I/O NWRITE	0x5	0x4
	Logical I/O NWRITE_R	0x5	0x5
	Logical I/O SWRITE	0x6	N/A
	Doorbell	0xA	N/A
Bridge ISF Response with Data	Response	0xD	0x8
Bridge ISF Response without Data	Response	0xD	0x0

Programming the RapidIO Ftype or Transaction Type fields to reserved values causes undefined behavior, and may prevent ISF transactions from being accepted. These values are in the RD_FTYPE, RD_SUBTYPE, WR_FTYPE, and WR_SUBTYPE fields of the "SREP I2R LUT Translation Parameters Register".

Translating Bridge ISF transactions to RapidIO maintenance and Logical I/O packets requires several protocol specific parameters to be specified, including:

- Packet Priority (separate for Read and Write requests)
- RapidIO Address Translation
- Critical Request Flow flag
- Destination ID
- Hop Count (for Maintenance packets)

The RapidIO protocol supports 34, 50 and 66 bit addressing. The address size is set globally for a system. Address size is controlled by the "SREP Processing Element Logical Layer Control CSR".

The SREP only supports 34 bit RapidIO addressing.

The hop count value in Maintenance Read and Maintenance Write transactions sent by the SREP is 1 less than the value programmed into the HOP_COUNT field of the "SREP I2R Upper LUT Entry Translation Register". This is done to allow the hop count of maintenance transactions sent to the SREP in the Tsi620 to be the same hop count as maintenance transactions originated by the SREP.

A HOP_COUNT field value of 0 is a programming error.

Every Bridge ISF request has a 64-bit address associated with it. The 64-bit address is mapped to RapidIO packet type and parameters through the use of the Bridge ISF-to-RapidIO Doorbell BAR, Bridge ISF-to-RapidIO Base Address Registers (I2R BARs) and Bridge ISF-to-RapidIO Lookup Table entries (I2R LUTs).

To transmit RapidIO requests, the MAST_EN bit of the "SREP General Control CSR" must be set to 1.

The values of the MAST_EN bit and the HOST bit in the "SREP General Control CSR" after the SREP is reset are determined by the latched values of the SP_MAST_EN and SP_HOST "Power-up Configuration Signals".

The I2R BARs select a range of Bridge ISF addresses for which Bridge ISF requests are accepted. The address ranges controlled by BARs can overlap with each other. Accesses that hit in the Doorbell BAR are translated to RapidIO Doorbell packets. Otherwise, the lowest numbered I2R BAR for an address range is selected to control the translation for that address range.

I2R BARs with parity errors are handled as unreadable and unwriteable (see "Bridge ISF-to-RapidIO Parity Error").

Each I2R BAR, except the Doorbell BAR, is linked to a range of I2R LUTs. The address space at which an I2R BAR responds is divided equally among the I2R LUT entries associated with the I2R BAR. Each LUT entry in turn specifies the parameters for the RapidIO packets to be created for Bridge ISF requests. The LUT entry allows parameters for reads to be different from those for writes.

The I2R BARs and I2R LUTs are programmed through the following registers:

- "SREP I2R LUT and BAR Parity Control Register"
- "SREP I2R Base Address Register x LUT Entry CSR"
- "SREP I2R Base Address Register x Upper"
- "SREP I2R Base Address Register x Lower"
- "SREP I2R Upper LUT Entry Translation Register"
- "SREP I2R Lower LUT Entry Translation Address Register"
- "SREP I2R LUT Translation Parameters Register"

The "SREP I2R LUT and BAR Parity Control Register" controls which I2R BAR and/or LUT entry is being programmed through the remaining registers.

The I2R BAR LUT entry usage, 64-bit Bridge ISF starting address, and BAR size and is programmed through the "SREP I2R Base Address Register x LUT Entry CSR", "SREP I2R Base Address Register x Upper", and "SREP I2R Base Address Register x Lower". The I2R BAR must be enabled by setting the "SREP I2R Base Address Register x Lower".BAR_EN bit to 1.

The starting address of each I2R BAR is a multiple of the BAR size. BAR address bits that would otherwise cause the BAR to begin at an address that is not a multiple of BAR size are ignored.

If the "SREP I2R LUT and BAR Parity Control Register".AUTO_INC bit is 1, then when the "SREP I2R LUT Translation Parameters Register" is read or written, the LUT_IDX field in the "SREP I2R LUT and BAR Parity Control Register" increments. When the LUT_IDX field reaches its maximum value, it rolls over back to 0. The AUTO_INC bit supports sequential programming/reading of the I2R LUTs without having to set the "SREP I2R LUT and BAR Parity Control Register" for every I2R LUT entry.

The RapidIO protocol parameters (Hop count, translation address, Critical Request Flow bit, priority, and destination ID size) are set through the use of the LUT registers. Note that the Critical Request Flow bit, and the RapidIO packet priority, can be set separately for RapidIO reads and writes.

The sourceID of all RapidIO request packets originated by the SREP is contained in the "SREP Base Device ID CSR". The DEST_ID or LG_DEST_ID fields are used based on the setting of the TT_CODE field of the "SREP I2R Lower LUT Entry Translation Address Register".

The reset value of the "SREP Base Device ID CSR" is controlled by the I2C_SA[6:0] and SP_HOST Power-up Options Signals.

The sourceID of all RapidIO response packets originated by the SREP is the destID of the request packet.

There are some additional controls in the LUT registers that are not directly related to the RapidIO protocol:

- "SREP I2R Lower LUT Entry Translation Address Register".RD_EN - If this bit is 0, Bridge ISF Reads to this LUT entry cause an error (see "Read Denied Event").
- "SREP I2R Lower LUT Entry Translation Address Register".WR_EN - If this bit is 0, Bridge ISF Writes to this LUT entry cause an error (see "Write Denied Event").
- "SREP I2R Upper LUT Entry Translation Register".PFTCH - This bit controls the Segmentation And Reassembly (SAR) algorithm used for Bridge ISF Read requests. This bit is discussed further in "Bridge ISF-to-RapidIO Request Segmentation And Reassembly".

Translation of Bridge ISF transactions into RapidIO requests, and receiving RapidIO responses, depend on the availability of RapidIO resources. Status bits in the "SREP R2I Event Status Register" indicate the availability of these resources as follows.

- NO_DECOMP_BUFF is 1 when no buffers are available for handling responses for decomposed Bridge ISF transactions. For more information on when decomposed response buffers are required, see "Bridge ISF-to-RapidIO Request Segmentation And Reassembly".
- NO_RIO_TID is 1 when no RapidIO Transaction IDs (TIDs) are available to send out RapidIO requests that require responses. RapidIO TIDs are required when a Bridge ISF transaction is being translated to a RapidIO NWRITE_R, NREAD, Maintenance Read, Maintenance Write, or DOORBELL request. Note that requests that do not require responses (NWRITE, SWRITE, Port-Writes) can continue to be transmitted as these do not require RapidIO TIDs.
- NUF_DECOMP is 1 when sufficient decomposed transaction buffers are available to allow the transaction at the head of the I2R queue to be decomposed, and 0 when there are not enough decomposed transaction buffers in the same situation. Note that NUF_DECOMP is only relevant when NUF_VALID is 1.

10.5.1 Bridge ISF Transaction Conversion to RapidIO Doorbells

The Bridge ISF address at which the Doorbell BAR responds is controlled by the following registers:

- "SREP I2R Doorbell BAR Upper"
- "SREP I2R Doorbell BAR Lower"

The Bridge ISF Doorbell BAR is 4 KB large. Transactions that hit the Bridge ISF Doorbell BAR are bridged to RapidIO Doorbell transactions.

When Bridge ISF transactions are bridged to DOORBELL requests, the destination ID of the DOORBELL request is part of the data written (see Table 31).

Table 31: Bridge ISF Write Data Used in Doorbell Packets

Bridge ISF Data Byte	Use
3	Most significant byte of a 16-bit destination ID, if the TT_CODE selects 16-bit destination IDs. If TT_CODE selects 8-bit destination IDs, this byte is not used. For more information on how TT_CODE is specified, see Table 32.
2	Least significant byte of a 16-bit destination ID, if the TT_CODE selects 16-bit destination IDs. If TT_CODE selects 8-bit destination IDs, this byte contains the 8-bit destination ID. For more information on how TT_CODE is specified, see Table 32.
1	Most significant byte of the DOORBELL data.
0	Least significant byte of the DOORBELL data.

Bridge ISF Write Burst transactions allow for non-contiguous byte enables in a write. The SREP detects an Unsegmentable Request error when data written to the "SREP I2R Doorbell BAR Upper"/"SREP I2R Doorbell BAR Lower" has non-contiguous byte enables.

The remaining Doorbell packet parameters are a function of the address at which the write occurred.

Table 32: Bridge ISF Write Address Used in Doorbell Packets

Bridge ISF Address Bits	Use
2:0	Must be 0 to ensure proper alignment of doorbell data.
4:3	Priority of the Doorbell. Note that a value of 0b11 in these bits is illegal, and causes an ISF Unsegmentable Request event.
6:5	TT code of the Doorbell. 0b00 = 8-bit destination ID 0b01 = 16-bit destination ID 0b10-0b11 = Reserved, causes in an ISF Unsegmentable Request event.

Table 32: Bridge ISF Write Address Used in Doorbell Packets (Continued)

Bridge ISF Address Bits	Use
7	CRF bit setting for the Doorbell
8:11	Must be 0.

The sourceID of all RapidIO Doorbell packets originated by the SREP is contained in the "SREP Base Device ID CSR". The DEST_ID or LG_DEST_ID fields are used based on the setting of the TT code value of the originating transaction (see Table 32).

Doorbell packets must be acknowledged with a response packet. The response packet can have a Done or Error status, just like Logical I/O packets. The response packets for doorbells may also have a Retry status. When a Retry response is received, the doorbell packet must be retransmitted.

To detect the situation where Doorbell packets are endlessly retried, the "SREP Response Timeout Control CSR" specifies a response timeout. The timeout for Doorbell packets is the time in which either a Done or Error status must be received in a response packet. Responses with a status of 'Retry' do not halt the response timeout. If the timeout expires, a RapidIO Response Timeout event is detected. The timeout for Doorbell packets is twice the interval programmed in the "SREP Response Timeout Control CSR" (see "Logical I/O Packet Response Timeout Events".

10.5.2 Bridge ISF-to-RapidIO Request Segmentation And Reassembly

There are four Bridge ISF request transactions that must be bridged to RapidIO, as captured inTable 33.

Table 33: Transaction Request Type Encoding

Transaction	Description
Memory Read Word	Read 8 or fewer bytes, controlled by byte lanes
Memory Read Block	Read up to 128 Bridge ISF data phases (1024 or fewer contiguous bytes)
Memory Write Burst	Write up to 128 Bridge ISF data phases (1024 or fewer bytes with byte lanes for every 8 byte data phase)
Memory Write Block	Write up to 128 Bridge ISF data phases (1024 or fewer bytes with contiguous byte lanes)

10.5.2.1 Bridge ISF Addressing/Alignment and Transaction Decomposition

The size of the address specified by a Bridge ISF request is 64-bits. Read and write request addresses are byte aligned to any byte boundary. By default, the address is aligned with the '0' location to the right (little endian) and unless otherwise specified this is the default orientation.

Bridge ISF request data payloads are 64-bit aligned, but if the address starts at a byte location within the 64-bit aligned address (that is, addr[2:0] is not equal to 0) the data in the first data phase that is outside the address start byte position is driven as “do not care.” The address of the remaining data proceeds linearly through the address space.

It is illegal to specify a size and starting address for a Bridge ISF request that requires more than 32, 64-bit data transfers for 256 byte payloads, or more than 128, 64-bit data transfers for 1024 byte payloads.

Bridge ISF requests must be mapped to RapidIO requests in a manner consistent with RapidIO restrictions on byte lane combinations (see “RapidIO Logical Layer Protocol Support”). A Bridge ISF transaction must be decomposed into multiple RapidIO transactions if one of the following conditions is true:

- The Bridge ISF transaction has a byte lane combination that does not match any byte lane combination supported by RapidIO
- The Bridge ISF transaction address is not aligned to an 8-byte boundary either at the start or end of the transaction, and crosses at least one 8-byte boundary
- The Bridge ISF transaction is greater than 256 bytes

In these cases, R2I buffers must be reserved to manage multiple responses, and in the case of a Bridge ISF Read Block and Read Word, compose a single Bridge ISF response transaction. Bridge ISF Read Block transactions require up to 4 R2I buffers to be reserved, depending on the size and alignment of the Bridge ISF request. Bridge ISF Read Word only requires 1 R2I buffer to be reserved. Write Block and Write Burst transactions require one buffer to be reserved when the transaction is decomposed into RapidIO NWRITE_R packets. When Write Block and Write Burst transactions are decomposed into NWRITE or SWRITE packets, no R2I buffers are required.

The number of Bridge ISF buffers reserved for decomposed transactions is controlled by the DECOMP field of the “SREP R2I Watermarks Register”. If a request must be decomposed but there are insufficient R2I DECOMP buffers free, the decomposition of that request will wait until sufficient buffers are free.

It is a programming error to set the DECOMP field value to 0. The minimum legal value for the DECOMP field is 1.

R2I buffers reserved for decomposed transactions cannot be used for RapidIO requests, or for responses to non-decomposed transactions.

If a Bridge ISF transaction must be decomposed, but there are insufficient decomposition buffers reserved by the “SREP R2I Watermarks Register”.DECOMP field setting, the Bridge ISF transaction cannot be decomposed. The Bridge ISF transaction will incur an I2R Transaction Time-to-Live Expired event (see “I2R Transaction Time-to-Live Expired”).

If no decomposition buffers are available, the NO_DECOMP_BUFF bit is set in the “SREP R2I Event Status Register”.

If insufficient decomposition buffers are available, the NUF_DECOMP bit is 0 and the NUF_VALID bit is 1 in the “SREP R2I Event Status Register”.

10.5.2.2 Bridge ISF Memory Read Word Segmentation

A Bridge ISF Memory Read Word request may be bridged to a RapidIO Maintenance Read request, or to multiple RapidIO NREAD requests.

The hop count value in Maintenance Read and Maintenance Write transactions sent by the SREP is 1 less than the value programmed into the HOP_COUNT field of the "SREP I2R Upper LUT Entry Translation Register". This is done to allow the hop count of maintenance transactions sent to the SREP in the Tsi620 to be the same hop count as maintenance transactions originated by the SREP.

A HOP_COUNT field value of 0 is a programming error.

If the Bridge ISF Memory Read Word request is bridged to a RapidIO Maintenance Read request, the Bridge ISF Memory Read Word request must have byte lanes and sizes that describe a 4 byte read at a 4-byte aligned address. The bytes to be read must be contiguous. A single RapidIO Maintenance Read request is issued. Attempting to bridge Bridge ISF Memory Read Word requests that are not 4 bytes at a 4-byte aligned address to RapidIO Maintenance Read requests causes an Bridge ISF Unsegmentable Request event (see "Bridge ISF Unsegmentable Request").

The PFTCH bit in the "SREP I2R Upper LUT Entry Translation Register" must not be set when the Bridge ISF transaction is translated to a RapidIO Maintenance Read request.

Bridge ISF Memory Read Word requests are bridged to a varying number of RapidIO NREAD requests based on the byte lanes in the Bridge ISF Memory Read Word request. If the byte lanes match a single natively supported RapidIO byte lane combination, then a single RapidIO NREAD request is issued. If the byte lanes do not match a single natively supported RapidIO byte lane combination, then the minimum number of RapidIO NREAD requests are issued for the byte lane combination. The NREAD responses are reassembled into one Bridge ISF response. Examples of this are displayed in Table 34.

Table 34: Bridge ISF Memory Read Word Decomposition Examples

Bridge ISF byte lanes (Active when 1)	NREAD Transaction(s) byte lanes (Active When 0)
0b1111_1111	0b0000_0000
0b0011_0000	0b1100_1111
0b1011_1111	0b0111_1111 0b1100_0000
0b1011_1000	0b0111_1111 0b1100_1111 0b1111_0111
0b1010_1010	0b0111_1111 0b1101_1111 0b1111_0111 0b1111_1101

10.5.2.3 Bridge ISF Memory Read Block Segmentation

A Bridge ISF Memory Read Block request may be bridged to a RapidIO Maintenance Read request, or to multiple RapidIO NREAD requests.

The hop count value in Maintenance Read and Maintenance Write transactions sent by the SREP is 1 less than the value programmed into the HOP_COUNT field of the "SREP I2R Upper LUT Entry Translation Register". This is done to allow the hop count of maintenance transactions sent to the SREP in the Tsi620 to be the same hop count as maintenance transactions originated by the SREP.

A HOP_COUNT field value of 0 is a programming error.

If the Bridge ISF Memory Read Block request is bridged to a RapidIO Maintenance Read request, the Bridge ISF Memory Read Block request must have byte lanes and sizes that describe a 4-byte read at a 4-byte aligned address. The bytes to be read must be contiguous. A single RapidIO Maintenance Read request is issued. Attempting to convert Bridge ISF Memory Read Block requests that are not 4 bytes at a 4-byte aligned address to RapidIO Maintenance Read requests causes an Bridge ISF Unsegmentable Request event (see "Bridge ISF Unsegmentable Request").

The PFTCH bit in the "SREP I2R Upper LUT Entry Translation Register" must not be set when the Bridge ISF transaction is translated to a RapidIO Maintenance Read request.

A Bridge ISF Memory Read Block request may be bridged to multiple RapidIO NREAD requests, depending on the setting of "SREP I2R Upper LUT Entry Translation Register".PFTCH.

If the PFTCH bit is set to 1, this means that it is safe to read additional memory to service the request. In this case, a RapidIO NREAD requests are issued which start at the Bridge ISF Memory Read Block address, rounded down to the nearest 8 byte boundary. The size of the RapidIO NREAD requests is the lowest multiple of 32 bytes which completely covers the Bridge ISF Memory Read Block request size. In the case of where the Bridge ISF Memory read is greater that 256 bytes, multiple RapidIO NREAD requests are issued.

The NREAD responses are assembled into a single Bridge ISF response. Note that the order in which NREAD responses are received may not match the order in which the NREAD requests were issued.

All bytes that are not required to service the original Bridge ISF Memory Read Block are dropped. Examples of Bridge ISF Memory Read Block Prefetchable Segmentation are in Table 35.

Table 35: Bridge ISF Memory Read Block Prefetchable Segmentation Examples

Bridge ISF Start Address	Bridge ISF Size (bytes)	RapidIO Start Address(es)	RapidIO Size(s) (bytes)
0x06F8	128	0x06F8	128
0x06FF	128	0x06F8	160
0x06FC	252	0x06F8	256

Table 35: Bridge ISF Memory Read Block Prefetchable Segmentation Examples (Continued)

Bridge ISF Start Address	Bridge ISF Size (bytes)	RapidIO Start Address(es)	RapidIO Size(s) (bytes)
0x06FC	129	0x06F8	160
0x0704	191	0x700	224
0x0400	1016	0x400 0x500 0x600 0x700	256 256 256 256
0x0404	800 (0x320)	0x400 0x500 0x600 0x700	256 256 256 64

If the PFTCH bit is set to 0, it is not safe to read additional memory to service the request. In this case, RapidIO NREAD requests must be issued that correspond to the requested memory. This generates more RapidIO requests as compared to the case when the PFTCH bit is set to 1.

It is recommended that the PFTCH bit for I2R LUT entries be set to 1 when possible to improve performance.

The algorithm that segments a Bridge ISF Memory Read Block request when the PFTCH bit is set to 0 is to issue an NREAD request to read an 8 byte boundary, and then issue the largest request that is smaller than the number of bytes remaining. This results in the minimum number of NREAD requests being issued. Note that the maximum number of NREAD requests issued is 8.

The NREAD responses are assembled into a single Bridge ISF response. Note that the order in which NREAD responses are received may not match the order in which the NREAD requests were issued.

Examples of Bridge ISF Memory Read Block Non-Prefetchable Segmentation are in Table 36.

Table 36: Bridge ISF Memory Read Block Non-Prefetchable Segmentation Examples

Bridge ISF Start Address	Bridge ISF Size (bytes)	RapidIO Start Address(es)	RapidIO Size(s) (bytes)
0x06F8	128	0x06F8	128
0x06F9	128	0x06F8 0x0700 0x0760 0x0770 0x0778	7 (byte lanes) 96 16 8 1 (byte lanes)

Table 36: Bridge ISF Memory Read Block Non-Prefetchable Segmentation Examples (Continued)

Bridge ISF Start Address	Bridge ISF Size (bytes)	RapidIO Start Address(es)	RapidIO Size(s) (bytes)
0x06FC	252 (0xFC)	0x06F8 0x0700 0x07E0 0x07F0	4 (byte lanes) 224 16 8
0x0402	1023 (0x3FF)	0x400 0x408 0x508 0x608 0x708 0x7E0 0x7F0 0x7F8	6 (byte lanes) 256 256 256 224 16 8 1 (byte lanes)
0x0404	800 (0x320)	0x400 0x408 0x508 0x608 0x708 0x718 0x720	4 (byte lanes) 256 256 256 16 8 4 (byte lanes)

10.5.2.4 Bridge ISF Memory Write Burst Segmentation

Bridge ISF memory write burst transactions have byte lane values associated with every data phase of the transaction. This complicates the conversion of Bridge ISF memory write burst transactions to RapidIO, as up to four RapidIO transactions may be required for every Bridge ISF data phase.

In some PCI devices, Bridge ISF memory write burst transactions are created even though the byte lanes for such transactions are contiguous, as in a Bridge ISF memory write block transaction. To handle memory write burst transactions as if they were memory write block transactions, set the BST_2_BLK bit in the "SREP I2R Upper LUT Entry Translation Register" to 1.

If BST_2_BLK is 1, Bridge ISF Memory Write Burst transactions are SARed based on the first data phase (head), the middle data phases (body), and the last data phase (tail). If BST_2_BLK is 1, then the head and body of Bridge ISF Memory Write Burst transactions are segmented as if they were Bridge ISF Memory Write Block transactions (see "Bridge ISF Memory Write Block Segmentation"). The data written is determined based on the starting address and size of the transaction. Byte enables are ignored for all data phases except the last one (the tail), which is SARed as a Bridge ISF Write Burst transaction (byte enables control what packets are sent).

Single data phase Bridge ISF Memory Write Burst transactions are SARed as if they were the tail of a packet. Byte enables control what packets are sent.

A Bridge ISF Byte Enables Discontiguous event is detected if a Write Burst transaction with discontiguous byte enables is handled as a BST_2_BLK transaction. For more information on Bridge ISF Byte Enables Discontiguous events, see "Bridge ISF Byte Enables Discontiguous Event".

A Bridge ISF memory write burst transaction may be bridged to RapidIO Maintenance Write, NWRITE, NWRITE_R or DOORBELL transactions. RapidIO Doorbell transactions are created if the Bridge ISF memory write block transaction address falls in the 4-K window with a starting address that is the I2R Doorbell BAR (see "SREP I2R Doorbell BAR Upper" and "SREP I2R Doorbell BAR Lower"). All other packet types are determined based on the value of the WR_FTYPE and WR_SUBTYPE fields in the "SREP I2R LUT Translation Parameters Register" (see Table 38).

Table 37: RapidIO NWRITE, NWRITE_R and SWRITE Packet Selection

WR_FTYPE	WR_SUBTYPE	Write < 8 bytes	Write Multiple of 8 bytes
5 (WRITE)	4 (NWRITE)	NWRITE	NWRITE
	5 (NWRITE_R)	NWRITE_R	NWRITE_R
6 (SWRITE)	0	NWRITE	SWRITE
8 (MAINTENANCE)	1	Maintenance Write (4 byte only)	N/A

If the Bridge ISF memory write burst transaction consists of a single 4 byte request to a 4 byte aligned address, the Bridge ISF memory write burst can be bridged to a RapidIO Maintenance Write. Note that the byte lanes must be contiguous. Attempting to convert Bridge ISF Memory Write Burst requests that are not 4 bytes at a 4-byte aligned address to RapidIO Maintenance Write requests results in Bridge ISF Unsegmentable Request events (see "Bridge ISF Unsegmentable Request").

If the Bridge ISF memory write burst transaction consists of a 4 byte request to an 8 byte aligned address, the Bridge ISF memory write burst can be bridged to a DOORBELL transaction. Otherwise, the Bridge ISF transaction fails (see "Bridge ISF Unsegmentable Request").

If the Bridge ISF memory write burst transaction is bridged to RapidIO NWRITE or NWRITE_R, then at least one NWRITE or NWRITE_R transaction is issued for every data phase of the Bridge ISF memory write burst transaction.

The segmentation of each data phase of the Bridge ISF memory write burst transaction matches that was completed for the Bridge ISF memory read word transaction (see "Bridge ISF Memory Read Word Segmentation"). The minimum number of NWRITE or NWRITE_R transactions are issued, consistent with the byte lane combinations supported by RapidIO.

10.5.2.5 Bridge ISF Memory Write Block Segmentation

Bridge ISF memory write block transactions can be bridged to RapidIO Maintenance Write, NWRITE, NWRITE_R, SWRITE, and DOORBELL transactions. RapidIO Doorbell transactions are created if the Bridge ISF memory write block transaction address falls in the 4-K window with starting address that is the I2R Doorbell BAR (see "SREP I2R Doorbell BAR Upper" and "SREP I2R Doorbell BAR Lower"). All other packet types are determined based on the value of the WR_FTYPE and WR_SUBTYPE fields in the "SREP I2R LUT Translation Parameters Register" (see Table 38).

The hop count value in Maintenance Read and Maintenance Write transactions sent by the SREP is 1 less than the value programmed into the HOP_COUNT field of the "SREP I2R Upper LUT Entry Translation Register". This is done to allow the hop count of maintenance transactions sent to the SREP in the Tsi620 to be the same hop count as maintenance transactions originated by the SREP.

A HOP_COUNT field value of 0 is a programming error.

Table 38: RapidIO NWRITE, NWRITE_R and SWRITE Packet Selection

WR_FTYPE	WR_SUBTYPE	Write < 8 bytes	Write Multiple of 8 bytes
5 (WRITE)	4 (NWRITE)	NWRITE	NWRITE
	5 (NWRITE_R)	NWRITE_R	NWRITE_R
6 (SWRITE)	0	NWRITE	SWRITE
8 (MAINTENANCE)	1	Maintenance Write (4 byte only)	N/A

If the Bridge ISF memory write block transaction consists of a single 4 byte request to a 4 byte aligned address, the Bridge ISF memory write block can be bridged to a RapidIO Maintenance Write. Note that the byte lanes must be contiguous. If translation to RapidIO Maintenance Writes is selected, but the write request is not 4 bytes or is not aligned to a 4-byte address, the Bridge ISF transaction fails (see "Bridge ISF Unsegmentable Request").

Single data phase Bridge ISF memory write block segmentation when bridging to NWRITE and NWRITE_R packets is similar to that described for "Bridge ISF Memory Read Word Segmentation". The minimum number of NWRITE or NWRITE_R transactions are issued, consistent with the byte lane combinations supported by RapidIO.

The minimum number of Multiphase Bridge ISF memory write block segmentation when bridging to NWRITE, NWRITE_R and SWRITE transactions is similar to that of Bridge ISF memory read block non-prefetchable segmentation.

- First, an NWRITE or NWRITE_R RapidIO packet is issued to align the remaining bytes to be written to an 8 byte boundary. This is called the head of the transaction. If the starting address of the transaction is 8 byte aligned, a head packet does not need to be issued.
- Subsequent RapidIO packets may be either NWRITE, NWRITE_R or SWRITEs. These are called the body of the transaction.
- At the end, if there are fewer than 8 bytes left to write, a final NWRITE or NWRITE_R packet is issued. This is called the tail packet of the transaction.

RapidIO NWRITE, NWRITE_R and SWRITE packets are allowed to be smaller than the RapidIO size field indicates, so body packets are issued with the maximum possible number of 8 byte quantities in them.

When Bridge ISF Write Burst transactions are SARed as Bridge ISF Write Block transactions, a tail packet is sent (for more information on Bridge ISF Write Burst transaction SARing, see "Bridge ISF Memory Write Burst Segmentation").

NWRITE_R response packets are counted to ensure that all responses are received before the Bridge ISF memory write block request is completed.

Examples of Bridge ISF Memory Write Block Segmentation are in Table 39.

Table 39: Bridge ISF Memory Write Block Segmentation Examples

Bridge ISF Start Address	Bridge ISF Size (bytes)	RapidIO Start Address(es)	RapidIO Packet Size(s) (bytes)	RapidIO Packet Type Allowed				
				Mnt Wr	Door-bell	NWR	NW_R	SWR
0x6F8	3, 2, 1	0x6F8	3, 2, 1	No	No	Yes	Yes	No
0x6F8	4	0x6F8	4	Yes	Yes	Yes	Yes	No
0x6FA	4	0x6F8	2 (byte lanes) 2 (byte lanes)	No No	No No	Yes Yes	Yes Yes	No No
0x06F8	128	0x06F8	128	No	No	Yes	Yes	Yes

Table 39: Bridge ISF Memory Write Block Segmentation Examples (Continued)

Bridge ISF Start Address	Bridge ISF Size (bytes)	RapidIO Start Address(es)	RapidIO Packet Size(s) (bytes)	RapidIO Packet Type Allowed				
				Mnt Wr	Door-bell	NWR	NW_R	SWR
0x06F9	128	0x06F8 0x0700 0x07F0	7 (byte lanes) 112 1 (byte lanes)	No No No	No No No	Yes Yes Yes	Yes Yes Yes	No Yes No
0x06FC	252 (0xFC)	0x06F8 0x0700	4 (byte lanes) 248	No No	No No	Yes Yes	Yes Yes	No Yes
0x0404	800 (0x320)	0x400 0x408 0x508 0x608 0x708 0x720	4 (byte lanes) 256 256 256 24 4 (byte lanes)	No No No No No No	No No No No No No	Yes Yes Yes Yes Yes Yes	Yes Yes Yes Yes Yes Yes	No Yes Yes Yes Yes No

When a RapidIO NWRITE, NWRITE_R or SWRITE packet is issued that is a multiple of 8 bytes, the RapidIO packet Size and WDPTR values indicate that the packet is 256 bytes (Size is 0xF, WDPTR is 0x1).

10.5.3 Bridge ISF-to-RapidIO Error Responses

In some circumstances Bridge ISF requests result in Bridge ISF error responses without transmitting a RapidIO request (see "Bridge ISF Error Conditions"). The Bridge ISF error responses are pushed into I2I Queue. The I2I Queue then adds its requests to the R2I Queues according to the RapidIO priority encoded in the ERROR_RESP field of the "SREP I2R Miscellaneous CSR".

10.6 Maintenance Port-Write Support

The SREP supports the origination and reception of Port-Write transactions, which are another variation of a maintenance transaction.

10.6.1 Receiving Port-Writes

Two received port-write transactions can be buffered by the SREP. The contents of the port-writes can be read from the following SREP registers:

- "SREP Port-Write Receive Status Register"
- "SREP Port-Write 0 Receive Buffer n Registers"
- "SREP Port-Write 1 Receive Buffer n Registers"

The "SREP Port-Write Receive Status Register" contains two status bits, PWn_RXD and PWn_ERR, for each port-write. PWn_RXD is set to 1 when a port write is accepted into port-write buffer n. If the port-write was of an illegal size (not 4 bytes or a multiple of 8 bytes, or greater than 32 bytes), then the PWn_ERR bit is set.

The PWn_SIZE field in the "SREP Port-Write Receive Status Register" defines how many of the port-write buffer registers have valid contents. PW_SIZE indicates the amount of data that was received, not the value of the packet Size field. If the port-write was an illegal size, the PW_SIZE field indicates the last 8 byte quantity which is at least partially valid. For example, if a port-write of 20 bytes was received, then the PW_SIZE field would have a value of 0x3. For port-writes that are greater than 32 bytes, PW_SIZE indicates a 32 byte port-write was received.

When both port-write receive buffers are available, port-write 0 buffers are used. Port-writes received while both PW0_RXD and PW1_RXD are set to 1 are discarded, causing the PW_DISC bit to be set to 1.

To discard a buffered port-write, write a 1 to the PWn_RXD bit in the "SREP Port-Write Receive Status Register" register. The PW_DISC and PWn_ERR bits are cleared when the PWn_RXD bit is cleared.

The ability to free up one port-write buffer at a time enables reception of one port write while another port-write is being transferred from the buffer registers.

The receipt of a port-write is signaled when the PW_RX_EN bit is set to 1 in the "SREP Interrupt Event Enable Register".

10.6.2 Software Testing of Port-Write Reception

The ability to fake port-write reception is used to verify software without receiving a port write. To fake the reception of a port-write, perform the following steps:

1. Disable port-write interrupt reporting by writing 0 to the PW_RX_EN bit in the "SREP Interrupt Event Enable Register".
2. Set the "SREP Port-Write 0 Receive Buffer n Registers" and "SREP Port-Write 1 Receive Buffer n Registers" to have the required data for the port-write(s).
3. Set the PWn_SIZE fields in the "SREP Port-Write Receive Status Register" to the appropriate value, and set the PWn_FAKE bit, in a single write to the "SREP Port-Write Receive Status Register". This causes the PWn_RXD bit to be set. The PWn_SIZE field is valid at this point.
4. Enable port-write interrupt reporting by writing 1 to the PW_RX_EN bit in the "SREP Interrupt Event Enable Register".

After the last step, an interrupt will be seen in the "SREP Interrupt Status Register" which can be handled by software.

A Port-Write packet can be received from the system while data is written to the "SREP Port-Write 0 Receive Buffer n Registers", and thus, will overwrite the port-write(s) being composed by software.

10.6.3 Event Notification Port-Write Transmission

The SREP can originate port-write packets in the standard format described by the RapidIO Error Management Extensions specification. For more information on how to enable events detected by the SREP to trigger transmission of port-write packets, see "Event Notification and Register Hierarchy".

The SREP can generate port-writes for events in other blocks. For more information on the event notification function, see "SREP Event Notification".

The Port-Write packet does not have a guaranteed delivery and does not have an associated response (see *RapidIO Interconnect Specification (Revision 1.2)*). Depending on system design, a port write may need to be sent repeatedly until cleared. The SREP continues to send port writes at an interval, programmable through the "SREP Port-Write Parameters Register" register. Port-writes are sent until the

- All enabled events are cleared
- The PORT_W_PEND bit in the "SREP Error and Status CSR" is cleared

The data in the Port-write message is generated based on the current state of the SREP's registers.

To send a port write, the destination ID must be set in the "SREP Port-Write Target Device ID CSR" and priority of the port-write packet must be set in the "SREP Port-Write Parameters Register".

The 16 byte data payload of the maintenance Port-Write packet contains the contents of several CSRs, the port that encountered the error condition, and implementation-specific information. The layout of the port-write packet is shown in Table 91.

The payload of the maintenance port-write packet is defined by the *RapidIO Interconnect Specification (Revision 1.2) RapidIO Error Management Extensions*.

Table 40: Port-Write Physical and Transport Layer Fields

Field Name	Value
TT	Determined by LARGE_DESTID field of "SREP Port-Write Target Device ID CSR" LARGE_DESTID = 0 means TT = 0b00 LARGE_DESTID = 1 means TT = 0b01
Source ID	Determined by TT value and DEST_ID or LG_DEST_ID field of "SREP Base Device ID CSR" If TT = 0b00, SourceID = DEST_ID If TT = 0b01, SourceID = LG_DEST_ID
Destination ID	Determined by TT value and DESTID_MSB and DESTID_LSB fields of "SREP Port-Write Target Device ID CSR" If TT = 0b00, Destination ID = DESTID_LSB If TT = 0b01, Destination ID = DESTID_MSB and DESTID_LSB
Hop Count	Always 0xFF

Table 40: Port-Write Physical and Transport Layer Fields (Continued)

Field Name	Value
CRF	Always 0
Priority	Determined by PW_PRIORITY field of "SREP Port-Write Parameters Register".

10.6.4 SREP Port-Write Transmit Trigger Register Transmission

To facilitate software testing, the SREP supports the transmission of port-writes triggered by software. The "SREP Port-Write Transmit Trigger Register" can send a port-write when hardware-triggered transmission of port-writes is disabled. To disable transmission of hardware-triggered port-writes, ensure that all port-write event sources are disabled in the "SREP Port-Write Event Enable Register".

The behavior of the SREP is not defined when transmission of port-writes from software is attempted when port-write transmission can be triggered by hardware.

When the "SREP Port-Write Transmit Trigger Register" is written with a PW_PEND value of 1, this causes a port-write to be transmitted at the earliest possible moment. Port-write transmission does not wait for the expiry of the "SREP Port-Write Parameters Register". If the "SREP Port-Write Parameters Register" time-period does expire before the port-write triggered by the "SREP Port-Write Transmit Trigger Register", the port-write transmitted uses the data in the "SREP Port-Write Transmit Trigger Register".

A port-write triggered by the "SREP Port-Write Transmit Trigger Register" is sent once.

If a write to "SREP Port-Write Transmit Trigger Register" sets the PW_PEND value to 0, then any pending port-write transmission request from this register is de-asserted.

10.7 RapidIO Doorbell Request Handling

The SREP can accept up to 32 doorbell requests into a queue, send an interrupt once the first doorbell is accepted, and present information about the doorbells for software processing. To accept doorbell requests into the queue, the DB_RX_EN bit in the "SREP Doorbell Receive Control Register" must be set to 1.

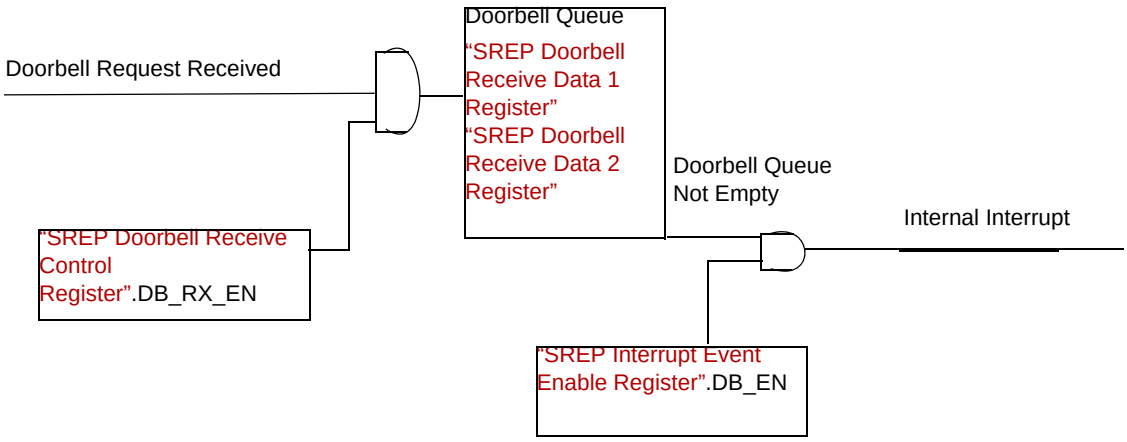
If a Doorbell request is received when DB_RX_EN is 0, a response with a status of Error is sent.

If doorbell requests can be accepted, then up to 32 doorbell requests can be stored in the doorbell queue in a first-in first-out order. If a doorbell request is received when all entries in the doorbell queue are occupied, a response packet with a RETRY status is generated for the doorbell request. The doorbell request is discarded. The sourceID of all RapidIO Doorbell response packets is the destID of the Doorbell request packet.

If doorbell requests are received into the doorbell queue, then doorbell interrupts can be generated. To enable generation of a doorbell interrupt when at least one doorbell event is in the queue, set DB_EN bit to 1 in the “SREP Interrupt Event Enable Register”.

Doorbell interrupts can be cleared by reading all valid doorbell requests from the queue, or by disabling the transmission of interrupts. The relationship of the various doorbell interrupt enables and interrupt lines is shown in Figure 24.

Figure 24: RapidIO Doorbell Interrupt Control Diagram



To handle a doorbell interrupt, perform the following steps:

1. Read the “SREP Doorbell Receive Data 1 Register”. If the “SREP Doorbell Receive Data 1 Register”.DB_VALID bit is 0, there are no more doorbell events in the queue.
2. If “SREP Doorbell Receive Data 1 Register”.DB_VALID is 1, read “SREP Doorbell Receive Data 2 Register”. This completes the handling of the doorbell event, and presents the next doorbell event in the “SREP Doorbell Receive Data 1 Register” if there is one.
3. Repeat these steps until no more valid doorbell events are detected.

If the “SREP Doorbell Receive Data 1 Register” is read twice in a row, the same data is returned. Only when the “SREP Doorbell Receive Data 2 Register” is read is a doorbell request popped from the head of the queue. At this point, the next (if any) doorbell request is presented in the “SREP Doorbell Receive Data 1 Register” and the“SREP Doorbell Receive Data 2 Register”.

If the doorbell request queue is full, then the “SREP Doorbell Receive Data 1 Register”.DB_FULL bit is 1.

If at least one Doorbell request is retried since the last time the “SREP Doorbell Receive Data 1 Register” was read, then the “SREP Doorbell Receive Data 1 Register”.DB_RTRY bit is set to 1.

Note that it is possible to receive more doorbell requests as doorbell events are being processed. Do not assume that the maximum number of doorbell events that can presented sequentially is 32.

When "SREP Doorbell Receive Data 2 Register" is read, the doorbell event is removed from the doorbell queue.

10.8 Transaction Ordering and Synchronization

The SREP supports the transaction ordering requirements of the *RapidIO Interconnect Specification (Revision 1.3)* Part 1, section 2.3.1. The ordering of transactions is defined in the context of a request flow, which is a sequence of transactions with the same source, destination, and priority. The transaction ordering requirements are:

- Logical I/O write transactions in the same flow must be completed at the Logical layer in the order that they were received.
- Logical I/O write transactions may be completed ahead of NREAD transactions in the same flow.
- Responses to Logical I/O transactions may be issued in a different order than the requests were received in.
- The physical and transport layers ensure that no reordering of write requests occurs in the SREP.
- A read request must be completed after all writes issued ahead of it in the same flow
- Maintenance transactions are not part of any flow, and so may be completed out of order with respect to Logical I/O requests.

Responses for Bridge ISF-to-RapidIO requests may be issued in any order, subject to the transaction ordering rules. Similarly, responses for RapidIO-to-Bridge ISF requests may be issued in any order.

The SREP completes a write request once it is issued to the Bridge ISF or to the address mapped register window. Note that the Bridge ISF target of the transaction completes the write after the Bridge ISF request is issued by the SREP.

The SREP completes a read request once a response is returned from the Bridge ISF or the address mapped register window. The read response for a Bridge ISF/address mapped register window is returned asynchronously from other read and write requests issued to different Bridge ISF targets. This behavior has impacts on the RapidIO control of the SREP.

For example, assume that a processor connected through RapidIO to the SREP needs to program a BAR and associated LUT entries, and then to access the Bridge ISF target associated with the BAR. The processor issues memory-mapped register writes to the SREP to configure the BAR and associated LUT entries. The processor must ensure that the register write transactions have taken effect before issuing any accesses to the newly programmed BAR address space.

The method for ensuring that accesses to the newly programmed BAR have taken effect is by issuing a read of the last register programmed. The read can be either a Maintenance Read or a Logical I/O NREAD transaction. The response to the read must be received by the processor before the processor issues any requests to the newly configured BAR.

In another example, a processor has performed reads and writes to a Bridge ISF target through RapidIO, and now needs to read a memory-mapped status register which will indicate the operation just completed. Before the status register read can be issued, the reads and writes to the Bridge ISF target must be completed.

If the last access to the Bridge ISF target is a read, the processor can wait for the reads completion before reading the memory-mapped status register. If the last access to the Bridge ISF target is a write, then a read must be performed after the write.

10.8.1 I2R Transaction Synchronization

Some Bridge ISF-to-RapidIO requests do not result in Bridge ISF responses, but do have RapidIO responses. Examples are Bridge ISF Write Burst or Bridge ISF Write Block transactions that are bridged to RapidIO Doorbell, NWRITE_R, and Maintenance Write requests. It is useful for the originating entity to know when these RapidIO transactions are completed for synchronization purposes.

Completion can mean receipt of a RapidIO response with a DONE or ERROR status, or that there is a timeout waiting for a RapidIO response.

NWRITE, NWRITE_R and Maintenance Write requests can be synchronized by issuing a Read request to the address that was written. Due to RapidIO ordering rules, the Read request is only complete after the Write request completes.

NWRITE_R, Maintenance Write and Doorbell requests can be synchronized by checking two status bits (DB_NO_ACK and NMWR_NO_ACK) in the "SREP R2I Event Status Register". These bits can be polled to indicate when outstanding doorbell requests, or NWRITE_R/Maintenance Write requests, are completed.

The DB_NO_ACK and NMWR_NO_ACK status bits do not cause information to be latched in the RapidIO Logical/Transport Error Information Registers, and cannot cause RIO_LOG events to occur in the "SREP Interrupt Status Register".

10.8.2 Synchronization of Data Path Control Register Changes

Some systems may require changes to data-path related control registers during normal system operation. Changing data-path related registers while there are transactions outstanding may result in unpredictable operation.

When data-path related control register values are changed during normal operation, the procedure in this section must be followed.

The data-path registers, which are subject to the procedure defined in this section, are between the "SREP R2I Base Address Register x LUT Control CSR" and "SREP R2R Queue Buffer Release Control Register", inclusive. Registers that require this procedure have a reference to this section as part of their description.

Additionally, the "SREP ISF Response Timeout Register" is subject to the procedure described in this section.

Before data-path related registers can be changed, all data path transactions must be completed, and new data path transactions must be prevented from starting. The procedure to ensure that all outstanding transactions are completed is as follows:

1. Set the MAST_EN bit in "SREP General Control CSR" to 0. This prevents the SREP from issuing more RapidIO requests.
2. Wait for all outstanding I2R RapidIO requests to be completed. To compute the time interval, see "Transaction End-to-End Time-to-Live".
3. Disable packet reception in the MAC. This prevents new RapidIO requests from being received.
4. Wait for all outstanding R2I RapidIO requests to be completed. To compute the time interval, see "Transaction End-to-End Time-to-Live".

Once all outstanding transaction are completed, and no new transactions have started, it is possible to change registers in the data path through the following means:

1. Perform a register write, using a RapidIO Maintenance Write or a register access from another block in the Tsi620, to change the register control value.
2. Perform a register read, using a RapidIO Maintenance Write or a register access from another block in the Tsi620, to synchronize the register control value change.
3. Repeat steps 1 and 2 until all register changes are completed.

After all register changes are completed, normal data-path operation can be resumed by setting "SREP General Control CSR".MAST_EN to 1 and/or enabling packet reception in the MAC.

10.8.3 NWRITE_R Response Completion

The response for an NWRITE_R RapidIO request is sent immediately after the NWRITE_R request is placed in the R2I Bridge ISF Queue. A successful response will be sent for the NWRITE_R, regardless of whether errors occur on the PCI bus.

To ensure that the NWRITE_R has completed successfully, perform these operations:

1. Perform an NREAD at the same address as the NWRITE_R. This will force completion of the NWRITE_R on the PCI bus.
2. Check that no PCI errors have occurred. This can be done by reading PCI error status registers, or by checking that an interrupt/MSI has not been received on the PCI bus. For more information, refer to "Error Handling".

10.9 Logical Layer Flow Control

RapidIO flow control makes use of three concepts: priority based reordering, watermarks, and buffer release management.

Logical layer flow control occurs in the RapidIO to Bridge ISF direction, and in the Bridge ISF to RapidIO direction. In the RapidIO to Bridge ISF direction, the logical layer flow control determines when transactions can be sent from the R2I Header Queue/Data Buffers to the Bridge ISF R2I Request Queue. In the Bridge ISF to RapidIO direction, logical layer flow control determines when transactions can be received from Bridge ISF into the I2R Header Queue/Data Buffers, and when Error and NWRITE_R responses can be accepted into the R2R Response Queue.

The watermark and release buffer management control registers for the Bridge ISF R2I Request queue are:

- "SREP R2I ISF Watermarks Control Register"
- "SREP R2I ISF Buffer Release Control Register"

The watermarks for the Bridge ISF I2R Request Queue are hard coded, so no control register exists for them. The buffer release management control registers for the Bridge ISF I2R Request queue is:

- "SREP I2R Buffer Release Control Register"

The "SREP I2R Buffer Release Control Register" is disabled when the "SREP General Control CSR".MAST_EN bit is set to 0.

The watermark and release buffer management control registers for the R2R Queue are:

- "SREP R2R Queue Watermarks Control Register"
- "SREP R2R Queue Buffer Release Control Register"

The values in the watermark and buffer release control registers must obey the relationships described in "Rules for Programming Watermarks" and "Buffer Release Management".

The behavior of I2R flow control is undefined if the values in any of the I2R flow control registers mentioned above have programming errors.

10.9.1 Rules for Programming Watermarks

The following rules apply to selection of watermark values for the SREP:

- No watermark is associated with Priority 3 packets;
- A Priority x packet is accepted in the buffer if the number of free buffers is greater than the programmed watermark of the associated Priority (PRIOxWM). For example, when PRIO1WM is programmed to "3", a Priority 1 packet is accepted only when there are 4 or more free buffers.
- The 3 programmed watermarks (PRIO0WM, PRIO1WM and PRIO2WM) have to contain values where PRIO0WM > PRIO1WM > PRIO2WM > 0 at all times.
- The watermarks have to be set such that:
  - PRIO2WM >= 1;
  - PRIO1WM >= 2;
  - PRIO0WM >= 3;

— PRIO2WM < PRIO1WM < PRIO0WM

Violating any one of the watermarks rules can create Deadlock situations in the system.

This hierarchy of watermarks ensures that packets of lower priority may never consume all buffers and prevent packets of higher priority from being transmitted. With the correct setting of the watermarks, the port must have at least one Priority 3 packet to transmit if all buffers are full.

10.9.2 Buffer Release Management

Buffer release management is a feature that allows the reporting of increases in the number of buffers (buffer releases) to be delayed. When the reporting of buffer releases is delayed, this has the effect of creating an opportunity for lower priority packets to be processed. Buffer release management can be used with watermarks to give lower priority packets an opportunity to make forward progress even under congested conditions.

Buffer release management does not guarantee that lower priority packets can make forward progress in congested conditions. It only guarantees that they will have an opportunity to progress.

The buffer release management registers have an identical programming algorithm, which uses the following fields:

- REL_MGMT_EN – Enable or disable buffer release management
- REL_MGMT_STOP – Stop reporting buffer releases when the number of free buffers reaches this number
- REL_MGMT_RES – Resume reporting buffer releases when the number of free buffers reaches this number. REL_MGMT_RES must be greater than REL_MGMT_STOP
- REL_MGMT_TO – The maximum period in which buffer releases are not be reported. If this period expires, then buffer releases are reported depending on the setting of REL_TO_MODE.
- REL_TO_MODE – Determines when buffer release management becomes active after a timeout.

For example, suppose the priority 0, 1 and 2 RapidIO receive watermarks are set to their default values of 3, 2, and 1. The system designer requires that priority 0 packets have the opportunity to progress even under high congestion scenarios. REL_STOP is set at 0, which stops free buffer reporting once all the buffers are filled. If REL_RES is set at 5, then free buffer reporting resumes once 5 buffers are released. To free 5 buffers, at least 2 buffers that can hold priority 0 packets must be processed. Note that there are no guarantees that two priority 0 packets will be processed, just that the opportunity exists for priority 0 packets to be received and processed.

In the previous example, buffer release management is not invoked until the buffer is full.

To prevent unwarranted delays in high priority traffic due to buffer release management, the REL_TO field limits the amount of time buffer release management is active. In the previous example, the REL_TO field could be set to the time it takes to transmit 5 packets. This ensures that once buffer release management is activated, at most it delays packet transmission for the time required to transmit 5 maximum sized packets.

REL_TO_MODE is relevant under congestion scenarios when REL_TO has expired and the buffer fill level has not reached the REL_RES value. If REL_TO_MODE is 0, then buffer release management does not delay reporting of buffer fill levels until the buffer fill level reaches the REL_RES value. This allows higher priority traffic to get through without delay, which may starve lower priority traffic until the congestion has eased. If REL_TO_MODE is 1, then buffer release management resumes delaying reportage of buffer fill levels if the number of free buffers ever reaches REL_MGMT_STOP. This delays higher priority traffic in favor of allowing lower priority traffic an opportunity to make forward progress. The mode chosen depends on the required system performance characteristics.

It is recommended that the REL_STOP level be set less than the watermark level of the RapidIO packet priority being protected, and that the REL_RES value be set at greater than the watermark level of the RapidIO packet priority being protected. It is further recommended that only one watermark occur between REL_RES and REL_STOP, to prevent starvation of the packet priority being protected by packets of higher priorities.

10.9.2.1 R2I Watermark and Buffer Release Management Register Value Restrictions

The following rules restrict the relationship between the Buffer Release Management and Watermark registers:

- The “SREP R2I Buffer Release Control Register” REL_MGMT_RES field value must be greater than the value in the PRIO0WM field of the “SREP R2I Watermarks Register”
- The “SREP R2I Buffer Release Control Register” REL_MGMT_RES field value, added to the number of buffers reserved for decomposed responses (DECOMP field of the “SREP R2I Watermarks Register”) must be less than 31.
- The “SREP R2I Buffer Release Control Register” REL_MGMT_STOP field value must be less than the PRIO2WM field value in the “SREP R2I Watermarks Register”.
- The “SREP R2I ISF Buffer Release Control Register” REL_MGMT_RES field value must be greater than the value in the PRIO0WM field of the“SREP R2I ISF Watermarks Control Register”
- The “SREP R2I ISF Buffer Release Control Register” REL_MGMT_STOP field must be less than the value of the PRIO2WM field of the “SREP R2I ISF Watermarks Control Register”

10.9.2.2 I2R Watermark and Buffer Release Management Register Value Restrictions

The following rules restrict the relationship between the Buffer Release Management and Watermark registers:

- The “SREP I2R Buffer Release Control Register” REL_MGMT_RES field value must be greater than 3
- The “SREP I2R Buffer Release Control Register” REL_MGMT_STOP field value must be less than 2

- The "SREP B2S Buffer Release Control Register" REL_MGMT_RES field value must be greater than the value in the PRIO0WM field of the "RapidIO Port x RapidIO Watermarks Register"
- The "SREP R2I ISF Buffer Release Control Register" REL_MGMT_STOP field must be less than the value of the PRIO2WM field of the "RapidIO Port x RapidIO Watermarks Register"

10.9.3 Cut-through Operation

One approach to handling transactions, referred to as store-and-forward, is to receive the entire transaction before starting to process the transaction. To reduce the latency of transactions, arbitration and transmission for the next stage of transaction processing may begin before a complete transaction is received. This mode of operation is called cut-through.

The SREP operates in store-and-forward mode in the RapidIO-to-Bridge ISF direction. RapidIO packets are cut-through from the Physical Layer to the Logical Layer. Once the complete packet is received, the packet can begin transmission over the Bridge ISF.

The SREP operates in cut-through mode in the Bridge ISF-to-RapidIO direction. In the Bridge ISF-to-RapidIO direction, is possible for data to be received from the Bridge ISF at a rate that is faster or slower than the RapidIO port can transmit it. If the Bridge ISF is faster than RapidIO, then the RapidIO port has data to transmit. If the Bridge ISF is slower than RapidIO, the RapidIO port may not have data to transmit, resulting in wasted bandwidth on the RapidIO link. Store-and-forward operation in the Bridge ISF-to-RapidIO direction is used in the MAC.

10.9.4 RapidIO R2I Buffer Reservation

The SREP has 32 R2I Buffers and Header Queue entries. These buffers are allocated among the following transaction types:

- RapidIO Logical I/O Requests, and responses to undecomposed Bridge ISF requests
- Responses for decomposed Bridge ISF requests
- I2I Transactions

10.9.4.1 R2I Buffers for I2I Transactions

I2I transactions are error responses for Bridge ISF requests (see "Bridging ISF Requests to RapidIO").

The R2I Buffers/Queues have two sources of packets - the SREP RapidIO Interface, and the I2I Queue.

I2I transactions are passed into the first available R2I buffer. Buffer availability is decided based on the RapidIO priority of the I2I transaction, and the number of free R2I buffers. When a buffer is available for an I2I transaction, the I2I transaction is enqueued before any R2I transactions of equal or lower priority.

10.9.4.2 RapidIO R2I Buffer Reservation for Decomposed Response Packets

Some I2R transactions must be decomposed into multiple RapidIO requests, which result in multiple response packets to complete the transaction. The separate RapidIO responses must be assembled into a single Bridge ISF response. For more information on I2R decomposition, see "Bridge ISF-to-RapidIO Request Segmentation And Reassembly".

All R2I Buffers necessary to compose the Bridge ISF response are reserved when the first I2R RapidIO request packet is issued for a decomposed Bridge ISF request. I2R requests that map to a single RapidIO request do not require SREP buffers to be reserved specifically for them.

Buffers are reserved for decomposed Bridge ISF requests using the DECOMP field of the "SREP R2I Watermarks Register".

It is a programming error to set the DECOMP field value to 0. The minimum legal value for the DECOMP field is 1.

Buffers reserved for decomposed Bridge ISF requests are not available for RapidIO requests, or for RapidIO responses to undecomposed Bridge ISF requests.

10.9.5 Register Transaction Flow Control

Register transaction flow control makes use of the same reordering, watermark, and release buffer management controls. For an explanation of reordering, watermarks, and release buffer management, see "RapidIO Physical Layer Flow Control".

Register transactions are serviced on a first-come-first-serve basis. NWRITE transactions do not require responses, and so can be serviced when they reach the head of the Register Request Queue. Register transactions that require a response (Maintenance Read/Write, NREAD, NWRITE_R) cannot be serviced until there is an entry available in the Register Response Queues.

Reordering, watermarks and buffer release management are applied to the Register Response queues on the basis of the RapidIO priority of the register responses. No reordering, watermarks, or buffer release management are required for the Register Request queue.

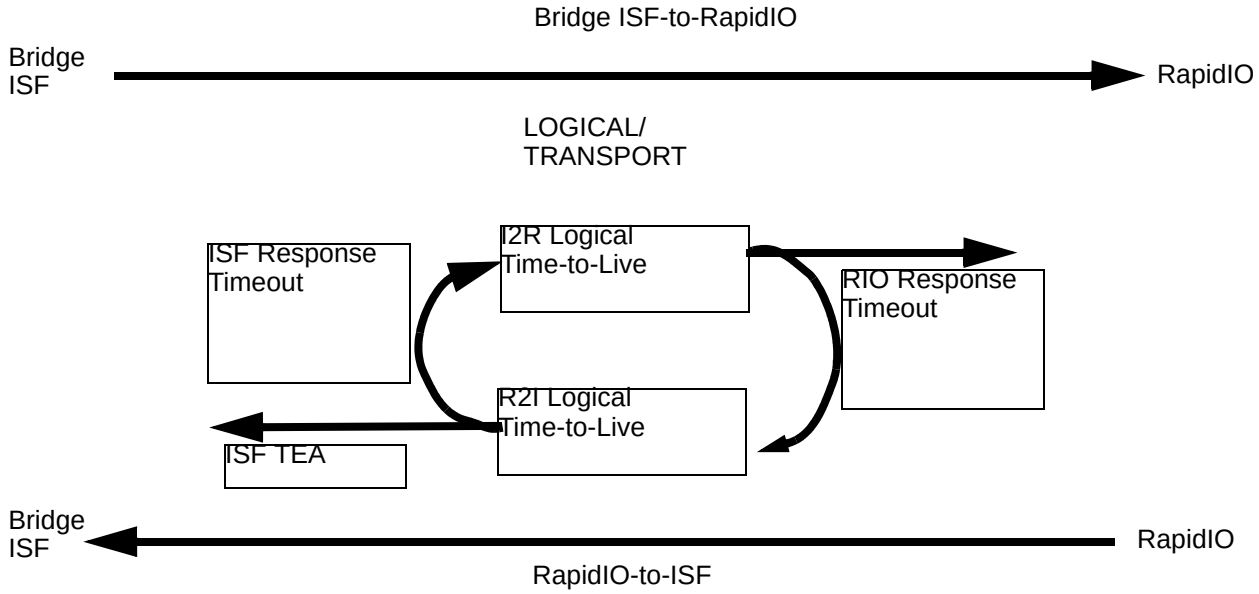
There is no ordering relationship maintained between RapidIO transactions that are serviced by the Register Request Queue/Buffer, and RapidIO transactions that are sent to the Bridge ISF. Software must ensure that register changes are complete before issuing RapidIO-to-Bridge ISF transactions that depend on those register changes, and vice-versa (see "Transaction Ordering and Synchronization").

10.10 Transaction End-to-End Time-to-Live

One of the interesting issues in packet based networks that operate a request/response protocol is how to make timeouts operate robustly. An endpoint making a request must be certain that, after a period of time, no response is returned for a request in order to initiate error handling. Additionally, the network must ensure that no response is returned after the timeout period. Otherwise, when the transaction tag for the timed-out request is used by a new request and a late response for the timed-out request arrives, the late response is matched up with the incorrect request. Debugging issues like this are difficult in small systems, and bordering on impossible with large systems.

The SREP allows other endpoints to place an upper bound on the amount of time before a request must have timed out. At each stage of a transactions progress through the SREP, the time the packet is buffered is tracked. If the time exceeds a programmed maximum, the packet is discarded. The different stages of a transactions progress, and the associated timing function, is shown in Figure 25.

Figure 25: SREP End-to-End Timeout Tracking



The end-to-end timeout for a Bridge ISF-to-RapidIO transaction is composed of the I2R Logical Time-to-Live period, added to the RIO Response Timeout, the R2I Logical Time-to-Live, and the Bridge ISF Transfer Error Acknowledge period.

The end-to-end timeout for a RapidIO-to-Bridge ISF transaction is composed of the R2I Logical Time-to-Live, the Bridge ISF Response Timeout, and the I2R Logical Time-to-Live.

The end-to-end timeout for a RapidIO-to-register transaction is composed of the amount of time required to process the register request and the I2R Logical Time-to-Live.

The R2I Logical Time-to-Live is controlled by the "SREP R2I Transaction Time-To-Live Register". This controls how long a request or response can be buffered in the R2I Data Buffers/Header Queue. If time-to-live expires for a transaction in these buffers, the transaction is dropped. If the transaction was a response to a Bridge ISF request, no Bridge ISF response is sent, so the requestor must time out the Bridge ISF request. For more information on R2I Time-to-Live Expired Events, see "RapidIO Logical Layer Time-to-Live Expired".

Note that there is no Time-to-Live control on the Register Access Request queue, or on the Doorbell Rx Queue. Register accesses complete in a predictable amount of time, so no timeout needs to be specified. The Doorbell Rx Queue similarly does not require a timeout.

The I2R Logical Time-to-Live is controlled by the "SREP I2R Transaction Time-To-Live Register". This controls how long a request or response can be buffered in the I2R Data Buffers/Header Queue, the Register Response Queue, and the R2R Queue. If a RapidIO response packet times out in the I2R Data Buffers/Header Queue, Register Response Queue or the R2R Queue, the response packet is dropped. There are individual status bits for each queue mentioned in this paragraph. For more information, see "R2R Time-to-Live Expired", "Register Access Response Time-to-Live Expired", and "I2R Transaction Time-to-Live Expired".

10.11 Logical I/O Packet Events

The SREP can detect the logical I/O packet error events specified in the *RapidIO Interconnect Specification (Revision 1.3)* Part 8: Error Management Extensions Specification, as well as a number of implementation-specific error events.

Note that some RapidIO logical I/O packet error events have implications for transactions bridged from Bridge ISF-to-RapidIO, and others are RapidIO specific. The detection of an error event normally results in the completion of a transaction with an error response. The error response may be sent on the Bridge ISF or RapidIO.

All logical I/O packet errors detected results in information being latched in the RapidIO error status registers (for more information on the information latched, see “Event Capture”).

10.11.1 Precedence of Logical I/O Packet Errors

Multiple errors may be detected in the same logical I/O packet, or for different errors to be detected at the same time on different packets. The SREP checks for errors in a specific order. The order in which the errors are detected is as follows:

1. Response Timeout
2. Packet TTL Expired
3. R2I Packet Stomped
4. R2I Packet CRC Error
5. Illegal Target
6. Unsupported Transaction
7. Illegal Transaction
8. Unexpected Response
9. Error Response
10. Doorbell Error Response
11. Spoof Response
12. R2I BAR Parity Error
13. OOB RIO Request
14. Illegal Register Access
15. R2I LUT Parity Error
16. RIO Write Denied
17. RIO Read Denied
18. LUT Entry Boundary Crossing

An ‘Illegal Target’, ‘R2I Packet CRC Error Detected’ or ‘Packet Stomp’ event is detected by the Transport layer before a Logical I/O packet event is detected. Additionally, the Transport layer may detect R2I Packet CRC events and Packet Stomped events at the same time (see “Transport Layer Events”).

10.11.2 Logical I/O Packet Response Timeout Events

The *RapidIO Interconnect Specification (Revision 1.3)* requires a timeout to be enforced for responses to Logical I/O request transactions. Responses are required for Doorbell, NREAD, NWRITE_R, and Maintenance Read and Maintenance Write transactions.

The hop count value in Maintenance Read and Maintenance Write transactions sent by the SREP is 1 less than the value programmed into the HOP_COUNT field of the "SREP I2R Upper LUT Entry Translation Register". This is done to allow the hop count of maintenance transactions sent to the SREP in the Tsi620 to be the same hop count as maintenance transactions originated by the SREP.

A HOP_COUNT field value of 0 is a programming error.

The timeout is programmed using the standard "SREP Response Timeout Control CSR". The TVAL field of the "SREP Response Timeout Control CSR" controls the timeout interval, which must be tracked for every transaction. Each packet has a 3 bit 'packet counter' associated with it, which counts down from 7 when a RapidIO request is issued. Once a 'packet counter' reaches 0, and the 'timeout counter' (described next) reaches 0, a response timeout is detected. A 'tick' is the unit interval for decrementing the 'packet counter'.

The 'timeout counter' drives the decrementing of the 'packet counters'. The 'timeout counter' counts down from TVAL. The timeout period for a packet varies depending on when the packet was received with respect to the timeout counter value. If the packet was received when the "timeout counter" was close to zero, the first 'tick' of the packet counter is much shorter than expected. The 'timeout counter' must expire 8 times before a packet is timed out.

A write to the "SREP Response Timeout Control CSR" takes effect immediately by reloading the 'timeout counter'. The timeout period is unpredictable when the 'timeout counter' is reloaded.

Response timeouts can be disabled by writing 0 to the "SREP Response Timeout Control CSR". Disabling response timeouts means that transactions will wait forever for responses. Resource exhaustion caused by transactions waiting for responses that will never come will prevent any new requests from being issued.

The 'timeout counter' decrements at the rate of the SREP Bridge ISF reference clock, divided by 4.

- If the reference clock is operating at 156.25 MHz, increasing TVAL by 1 adds 204.8 nsec to the timeout interval.
- If the reference clock is operating at 125 MHz, increasing TVAL by 1 adds 256 nsec to the timeout interval.

The response timeout operation for RapidIO Doorbell requests is twice as long as that for other RapidIO packets. This is completed to account for receiving Doorbell RETRY responses. The ‘packet counter’ for Doorbell requests is allowed to expire twice. Before the ‘packet counter’ expires once, RETRY responses are handled by re-sending the Doorbell request. After the ‘packet counter’ has expired once, any RETRY response is dropped. The Doorbell request is not resent if a RETRY response is received after the ‘packet counter’ has expired once. After the ‘packet counter’ for a Doorbell request has expired twice, no responses for retried Doorbell requests are expected to be received. A Response Timeout event is detected.

When RapidIO request packet has not received a response in the timeout interval, a Response Timeout event is detected. Response Timeout events cause the L_RESP_TO bit in the “SREP Logical and Transport Layer Error Detect CSR” to be set. Information about the request packet is latched in the Logical/Transport Layer Error Information registers, starting with the “SREP Logical and Transport Layer Address Capture CSR”.

For more information about the operation of the Logical/Transport Layer Error Information registers, see “Event Capture”.

A Response Timeout event causes the request packet to be dropped from the buffers of the SREP. a Bridge ISF error response may be sent to complete the Bridge ISF transaction that was bridged to the RapidIO request packet. If the I_ERESP_DIS bit in the “SREP I2R Miscellaneous CSR” is set to 0, a Bridge ISF error response is sent for Response Timeout Events. If the I_ERESP_DIS bit is set to 1, a Bridge ISF error response is not sent for Response Timeout Events.

10.11.2.1 Disabling Response Timeouts

By default, detection of response timeouts is enabled. However, for lab debug purposes, it is sometimes useful to disable timeouts for responses.

To disable response timeout events, set the TVAL field to 0 in the “SREP Response Timeout Control CSR”. Conversely, when the TVAL field is non-zero, RapidIO logical layer response timeouts are enabled.

10.11.2.2 Creating Response Timeout Events

A response timeout event can be created by bridging a read request from Bridge ISF to RapidIO for which no RapidIO entity responds.

10.11.3 RapidIO Logical Layer Time-to-Live Expired

An R2I Packet Time-to-Live Expired Event is detected when a RapidIO packet is buffered in the R2I Data Buffer/Header Queue for a period longer than that programmed in the “SREP R2I Transaction Time-To-Live Register”.

Logical layer time-to-live events are separate from transport layer time-to-live events. For more information on Transport Layer time-to-live events, see “Packet TTL Expired”.

Checking for R2I Packet Time-to-Live Expired Events is enabled when the “SREP R2I Transaction Time-To-Live Register”.TVAL field is non-zero.

Each transaction in the R2I Data Buffer/Header Queue has a time-to-live value associated with it. If the time-to-live value is 0 at the time that the transaction is selected to be put into the Bridge ISF Request Queue, an R2I Packet Time-to-Live Expired Event is detected.

Detection of a R2I Packet Time-to-Live Expired Event causes the L_R2I_TTL bit in the "SREP Logical and Transport Layer Error Detect CSR" to be set to 1. When the R2I_TTL_EN bit is set to 1 in the "SREP Logical and Transport Layer Error Logging Enable CSR", assertion of L_R2I_TTL causes the packet information to be latched in the Logical I/O Status registers, starting at "SREP Logical and Transport Layer Address Capture CSR".

For more information about the operation of the Logical/Transport Layer Error Information registers, see "Event Capture".

The packet for which a time-to-live expired event is detected is discarded. If the packet was a RapidIO request, no error response is sent to the RapidIO requestor. If the packet was a RapidIO response, no error response is sent to the Bridge ISF requestor.

To create an R2I Packet Time-to-Live Expired Event, perform the following steps:

1. Set the "SREP R2I ISF Watermarks Control Register" to 0x00080808, which prevents the transfer of all RapidIO transactions.
2. Set the "SREP R2I Transaction Time-To-Live Register".TVAL field to a non-zero value.
3. Send a valid RapidIO NREAD/NWRITE/NWRITE_R/SWRITE transaction, which will be bridged to Bridge ISF to the SREP.
4. Wait for a period of time that allow the time-to-live period to expire.
5. Set the "SREP R2I ISF Watermarks Control Register" to 0x00010203, which allows the RapidIO transaction to be transmitted.

An R2I Packet Time-to-Live Expired Event should be detected at this point.

10.11.4 RapidIO Unsupported Transaction Events

The 'Unsupported Transaction' event means that the SREP has received a packet from RapidIO that it does not know how to process.

Each RapidIO packet has an FTYPE value that defines the request/response format of the packet. The SREP does not support all defined FTYPEs. Each RapidIO packet also has a Transaction Type field (also called TTYPE or TYPE), which further refines the type of request/response that the packet represents. For some FTYPEs, the SREP does not support all defined Transaction values. A summary of those FTYPE/Transaction combinations that are not supported by the SREP is contained in Table 41.

Packets that have an unsupported FTYPE are dropped without a response.

Request packets that have a supported FTYPE, but whose Transaction type is not supported, cause an Error response to be generated. The request is dropped. This is explained in Table 41.

Reception of an unsupported packet causes the L_UNSUP_TRANS bit in the "SREP Logical and Transport Layer Error Detect CSR" to be set to 1. When the UNSUP_TRANS_EN bit is set to 1 in the "SREP Logical and Transport Layer Error Logging Enable CSR", assertion of L_UNSUP_TRANS causes the packet information to be latched in the Logical/Transport Layer Error Information registers.

For more information about the operation of the Logical/Transport Layer Error Information registers, see "Event Capture".

Table 41: Unsupported Requests/Responses Detected

Packet Type	Field Name	Unsupported Requests/Responses	RapidIO Error Response	Bridge ISF Error Response
REQUEST FTYPE = 2	Transaction	0b1100 = ATOMIC increment 0b1101 = ATOMIC decrement 0b1110 = ATOMIC set 0b1111 = ATOMIC clear	Yes	No
WRITE FTYPE = 5	Transaction	0b1100 = ATOMIC swap 0b1101 = ATOMIC compare-and-swap 0b1110 = ATOMIC test-and-swap	No	No
FTYPE 0, 7, 9, 11, 15	FTYPE	0, 1, 3, 4, 7, 9, 11, 12, 14, 15	No	No
FTYPE 8, 13	Status	0b1100-0b1111 (Conditional)	No	Yes
FTYPE 8, when TTYPE = 2 or 3 (Maintenance Read response and Maintenance Write response)	Status	0b0011 (Retry)	No	Yes for read, No for write
FTYPE 13	Status	0b0011 (Retry) when the Request FTYPE is not 10 (Doorbell)	No	Yes

Response packets with a status value of 0b0001 through 0b0111, and 0b1001 through 0b1011, causes an Unsupported Transaction Event to be detected. a Bridge ISF response with an Error status is sent if the outstanding transaction requires a response. The RapidIO request is handled as completed with an 'ERROR' response.

Response packets with status values of 0b1100-0b1111 by default cause an Unsupported Transaction event to be detected, and are handled as described in Table 41. If a system makes use of these implementation-specific status values, it is possible to process them as DONE or ERROR status values through the "SREP R2I RapidIO Miscellaneous Control CSR". Implementation-specific status values interpreted as 'ERROR' responses result in an 'Error Response' event being detected. In both cases, the reception of the response packet completes the RapidIO request (see "RapidIO Error Response Events").

Reception of an FTYPE 8 with a status of Retry is an unsupported response. If the original Bridge ISF request was a read, a Bridge ISF error response is sent to complete the transaction.

Reception of an FTYPE 13 with a status of Retry is only an unsupported transaction if the originating request is not a Doorbell. If the original Bridge ISF request was a read, a Bridge ISF error response is sent to complete the transaction.

To create an unsupported request/response event, configure a RapidIO device to send a RapidIO packet as described in Table 41 to the SREP.

10.11.5 RapidIO Illegal Transaction Events

An 'Illegal Transaction' event occurs when a RapidIO packet is received that is supported, but has reserved values for defined fields. A summary of the fields and field values that are detected as illegal is in Table 42.

Reception of an illegal transaction causes the L_ILL_TRANS bit in the "SREP Logical and Transport Layer Error Detect CSR" to be set to 1. When the ILL_TRANS_EN bit is set to 1 in the "SREP Logical and Transport Layer Error Logging Enable CSR", assertion of L_ILL_TRANS causes the packet information to be latched in the Logical/Transport Layer Error Information registers.

For more information about the operation of the Logical/Transport Layer Error Information registers, see "Event Capture".

Packets that cause an illegal transaction event are dropped. If the illegal transaction is a supported FTYPE that requires a response, then an error response packet is generated.

Table 42: Illegal Field Values Detected

Protocol Layer	RapidIO Packet Bit Field Name	Bit Field Description	Reserved Values	RapidIO Error Resp	Bridge ISF Error Resp
Logical	Transaction, when FTYPE = 2	The specific transaction within the format class	0b0000-0b0011, 0b0101-0b1011	No	N/A
Logical	Transaction, when FTYPE = 5	The specific transaction within the format class	0b0000-0b0011 0b0110-0b1011 0b1111	No	N/A
Logical	Status, when FTYPE = 8 or 13	Reserved values of status	0b0001-0b0111, 0b1001-0b1011	N/A	Yes, for BISF reads
Logical	Transaction, when FTYPE = 8	The specific transaction within the format class	0b0101-0b1111	No	N/A
Logical	Transaction, when FTYPE = 13	The specific transaction within the format class	0b0001-0b0111 0b1001-0b1111	No	Yes, for BISF reads

Table 42: Illegal Field Values Detected (Continued)

Protocol Layer	RapidIO Packet Bit Field Name	Bit Field Description	Reserved Values	RapidIO Error Resp	Bridge ISF Error Resp
Logical	Status	The success or failure of a request	0b0001-0b0010 0b0100-0b0110 0b1000-0b1011	No	Yes, for BISF Read
Logical	wdptr/size combinations	Specify the size of an NWRITE/NWRITE_R	0b0 / 0b1101 0b0 / 0b1110 0b1 / 0b1110 0b0 / 0b1111	No for NWRITE, Yes for NWRITE_R	N/A
Logical	Priority, when FTYPE = 2 OR when FTYPE = 10 OR when FTYPE = 5 and TTYPE = 5 OR when FTYPE = 8 and TTYPE = 0, 1	Specifies the priority of a transaction (NREAD, NWRITE_R, Maintenance Read/Write Requests, Doorbells)	3 It is not possible to make a request at priority 3 when the request requires a response.	Yes Note: Priority of the response is 3.	N/A

To create an illegal transaction event, configure a RapidIO device to send a RapidIO packet as described in Table 42 to the SREP.

10.11.6 RapidIO Unexpected Response

An ‘Unexpected Response’ event is detected when a response packet is received for a RapidIO transaction ID that is not in use.

RapidIO requests and responses are matched through the use of an 8-bit field in each packet called a Transaction ID. Transaction IDs are managed by the SREP. Each request packet is transmitted with a transaction ID unique to that packet. A response packet must have the same transaction ID as the request. Once a response is received, or the response times out, the transaction ID can be reused. An ‘Unexpected Response’ event is detected when a response packet is received with a Transaction ID that does not match any request tracked by the SREP.

Reception of an unexpected response packet causes the L_UNEXP_RESP bit in the “SREP Logical and Transport Layer Error Detect CSR” to be set to 1. When the UNEXP_RESP_EN bit is set to 1 in the “SREP Logical and Transport Layer Error Logging Enable CSR”, assertion of L_UNEXP_RESP causes the packet information to be latched in the Logical I/O Status registers, starting at “SREP Logical and Transport Layer Address Capture CSR”.

For more information about the operation of the Logical/Transport Layer Error Information registers, see “Event Capture”.

An unexpected response packet is dropped. No other fault handling needs to be completed for an unexpected response packet.

Creation of an Unexpected Response event can be completed by writing to the Logical I/O Error Management registers.

10.11.7 RapidIO Error Response Events

An ‘Error Response’ event is detected when a response packet with a Status of ‘ERROR’ is received for an existing RapidIO request.

Response packets for Maintenance Reads, Maintenance Writes, NWRITE_R and NREAD packets all have a 4 bit field called ‘Status’ in them. Status has two defined values, one for ERROR which means that the request could not be completed successfully, and one for DONE which means that the request was completed successfully and the requested data, if any, is returned.

The hop count value in Maintenance Read and Maintenance Write transactions sent by the SREP is 1 less than the value programmed into the HOP_COUNT field of the “SREP I2R Upper LUT Entry Translation Register”. This is done to allow the hop count of maintenance transactions sent to the SREP in the Tsi620 to be the same hop count as maintenance transactions originated by the SREP.

A HOP_COUNT field value of 0 is a programming error.

Detection of an Error Response event causes the L_ERR_RESP bit in the “SREP Logical and Transport Layer Error Detect CSR” to be set to 1. When the ERR_RESP_EN bit is set to 1 in the “SREP Logical and Transport Layer Error Logging Enable CSR”, assertion of L_ERR_RESP causes the information for the request packet (not the error response) to be latched in the Logical I/O Status registers, starting at “SREP Logical and Transport Layer Address Capture CSR”.

For more information about the operation of the Logical/Transport Layer Error Information registers, see “Event Capture”.

As error response event causes the request packet to be dropped from the buffers of the SREP. a Bridge ISF error response is sent to complete a Bridge ISF Read Word or Read Burst transaction that was bridged to the RapidIO request packet. No Bridge ISF error response needs to be sent for Bridge ISF Write Block or Write Burst transactions.

Creation of an Error Response event can be completed by bridging a Bridge ISF READ to a RapidIO NREAD packet that is routed to the SREP. When no RapidIO-to-Bridge ISF BARs are enabled, and the L_OOB error is not enabled, the SREP sends a RapidIO ERROR response back to itself to complete the RapidIO transaction. When the L_ERR_RESP_EN bit is set in the “SREP Logical and Transport Layer Error Logging Enable CSR”, reception of the error response causes the detection of an L_ERR_RESP event.

10.11.8 RapidIO Doorbell Error Response Events

A “Doorbell Error Response” event is detected when a response packet with a Status of ‘ERROR’ is received for an existing RapidIO Doorbell request.

Response packets for Doorbell packets all have a 4-bit field called ‘Status’ in them. Status has three defined values:

- ERROR – The request could not be completed successfully

- RETRY – The request could not be completed at this time. Attempt the request again.
- DONE – The request was completed successfully.

Detection of a Doorbell Error Response event causes the L_DB_ERR_RESP bit in the "SREP Logical and Transport Layer Error Detect CSR" to be set to 1. When the DB_ERR_RESP_EN bit is set to 1 in the "SREP Logical and Transport Layer Error Logging Enable CSR", assertion of L_DB_ERR_RESP causes the packet information to be latched in the Logical I/O Status registers, starting at "SREP Logical and Transport Layer Address Capture CSR".

For more information about the operation of the Logical/Transport Layer Error Information registers, see "Event Capture".

A doorbell error response event causes the request packet to be dropped from the buffers of the SREP. No error response is sent on Bridge ISF since the Bridge ISF transaction that triggered the doorbell does not require a response.

Creation of a Doorbell Error Response event can be completed by disabling reception of doorbell requests in the SREP by setting the DB_RX_EN bit in the "SREP Doorbell Receive Control Register" to 0. If a Bridge ISF WRITE is bridged to a RapidIO Doorbell packet that is routed to the SREP, the SREP sends a response packet with an Error status back, which triggers the Doorbell Error Response event.

10.11.9 RapidIO OOB Request Event

An 'Out Of Bounds Request' event is detected when an NREAD, NWRITE, NWRITE_R or SWRITE packet is received with an address that does not match the LCS BAR ("SREP Local Configuration Space Base Address CSR") or any enabled R2I BAR.

NREAD, NWRITE, NWRITE_R and SWRITE packet all contain an Address field that identifies the RapidIO 34, 50 or 66 bit address that the packet should affect. Each BAR identifies a range of RapidIO addresses that the SREP must respond to. The "SREP Local Configuration Space Base Address CSR" identifies the RapidIO address space at which NREAD, NWRITE, and NWRITE_R requests can access the SREP's registers. For more information on how the BARs and "SREP Local Configuration Space Base Address CSR" are used, see "Bridging Logical I/O Requests to the Bridge ISF".

Reception of an OOB Request causes the L_OOB bit in the "SREP Logical and Transport Layer Error Detect CSR" to be set to 1. When the OOB_EN bit is set to 1 in the "SREP Logical and Transport Layer Error Logging Enable CSR", assertion of L_OOB causes the packet information to be latched in the Logical I/O Status registers, starting at "SREP Logical and Transport Layer Address Capture CSR".

For more information about the operation of the Logical/Transport Layer Error Information registers, see "Event Capture".

A OOB Request event causes the request packet to be dropped from the buffers of the SREP. If the request requires a response (that is, NREAD, NWRITE_R), an error response is sent to complete the RapidIO request.

Creation of an Error Response event can be completed by bridging a Bridge ISF READ to a RapidIO NREAD packet that is routed to the SREP. When no RapidIO-to-Bridge ISF BARs are enabled, and the L_OOB error is enabled, the receipt of the request causes the detection of an L_ERR_RESP event.

10.11.10 RapidIO Illegal Register Access

An Illegal Register Access event is detected in three situations:

- An attempt is made to read or write registers with a RapidIO Maintenance request that is greater than 4 bytes or not aligned to a 4-byte address.
- An attempt is made to read or write registers with a RapidIO NREAD/NWRITE/NWRITE_R request that is greater than 4 bytes, not aligned to a 4-byte address, or both.
- An attempt is made to write registers from a source ID that is not enabled.

The SREP supports Maintenance Read and Maintenance Write requests to access to SREP registers that are 4 bytes and occur at a 4-byte aligned address. An Illegal Register Access event is detected when a Maintenance Read or Maintenance Write request is not 4 bytes, or does not occur at a 4-byte aligned address.

The SREP also supports NREAD, NWRITE, and NWRITE_R requests that are 4 bytes or less at a 4-byte aligned address. An Illegal Register Access event is detected if the NREAD, NWRITE or NWRITE_R request has a size greater than 4 bytes, or attempt to access registers at a non-4-byte aligned address.

Since SWRITE packets only support accesses that are a multiple of 8 bytes, SWRITE packets cannot be used to access the SREP registers.

The SREP allows users to restrict write accesses to registers based on the source ID of the register transactions. Separate controls exist for 8-bit and 16-bit destination IDs (for more information, see “Register Access Source ID Management”).

Reception of an Illegal Register Access causes the L_BAD_REG_ACC bit in the “SREP R2I Event Status Register” to be set to 1. When the BAD_REG_ACC_EN bit is set to 1 in the “SREP R2I Event Status Logging Enable Register”, assertion of L_BAD_REG_ACC causes the packet information to be latched in the Logical I/O Status registers, starting at “SREP Logical and Transport Layer Address Capture CSR”.

For more information about the operation of the Logical/Transport Layer Error Information registers, see “RapidIO Logical/Transport Error Information Registers”.

For more information on the programming of register access source ID controls, see “Register Access Source ID Management”.

An Illegal Register Access event causes the request packet to be dropped from the buffers of the SREP. If the request requires a response (that is, NREAD, NWRITE_R, Maintenance Read, Maintenance Write), an error response is sent to complete the RapidIO request.

Creation of an Illegal Register Access event can be completed by bridging a Bridge ISF READ of a size greater than 4 bytes to a RapidIO NREAD packet that is routed to the SREP. The NREAD request must have an address that hits in the "SREP Local Configuration Space Base Address CSR". Reception of the request causes an L_BAD_REG_ACC event to be detected.

It is also possible to create an Illegal Register Access event by bridging a Bridge ISF WRITE that is 4 bytes and aligned to a 4-byte address to a write transaction that is looped back to the SREP, when the SREP Destination ID programmed in the "SREP Base Device ID CSR" is not allowed to write to the registers (see "Register Access Source ID Management").

10.11.11 RapidIO Spoof Response Event

A Spoof Response event is detected when a response is received that has a source ID that is different from the destination ID for the request, when the type of response does not match the type of request, or amount of data in the response does not match the amount of data requested.

The RapidIO protocol requires that responses must contain the source and destination IDs of the request, in reverse order. Destination IDs may swap the source and destination IDs, or they may put their own destination ID into a response as the source.

The RapidIO protocol does not define required behavior for destination/source ID swapping under error conditions.

Note that the source ID is also judged to be different if the TT code of the response packet does not match the TT code of the request packet.

A 'Spoof Response' event is also detected when a response packet is received that does not match the request packet type. These situations are:

- Maintenance Read Request does not receive maintenance read response
- Maintenance Write Request does not receive maintenance write response
- NREAD request does not receive a Type 13 response
- NWRITE_R request does not receive a Type 13 response

A 'Spoof Response' event is also detected when a response packet is received with an amount of data that does not match the amount of data requested. This can occur in two cases:

- The response to an NREAD request has an amount of data different from that of the NREAD request.
- The response to an NWRITE_R or Doorbell request has any data, regardless of the status of the response.

When the amount of data received differs from the amount of data expected, this indicates that a late response was received for a Transaction ID that has timed out and has since been reused.

The transaction timeout strategy/design for the system must be corrected to avoid this error.

Reception of an appropriate Maintenance Read/Write response or Type 13 response with a status of ‘Error’ will never result in detection of a SPOOF response. For information on the events related to reception of a response with a status of ‘Error’, refer to “RapidIO Error Response Events” or a “RapidIO Doorbell Error Response Events”.

Reception of a Spoof Response event cause the L_SPOOF bit in the “SREP Logical and Transport Layer Error Detect CSR” to be set to 1. When the SPOOF_EN bit is set to 1 in the “SREP Logical and Transport Layer Error Logging Enable CSR”, assertion of L_SPOOF causes the packet information to be latched in the Logical I/O Status registers, starting at “SREP Logical and Transport Layer Address Capture CSR”.

A Spoof Response event causes the response packet to be dropped. No change occurs to the outstanding request. No other handling is performed for a Spoof Response in anticipation of the ‘real’ responses eventual arrival.

Creation of a Spoof Response event can be completed by bridging a Bridge ISF READ to a RapidIO NREAD packet that is routed to a RapidIO endpoint, that will return a response based on its own destination ID, not the routing information in the packet. The RapidIO endpoint must be configured to accept the packet, even though the destination ID does not match that of the endpoint. Reception of the response to this request causes an L_SPOOF event to be detected.

10.11.12 RapidIO-to-Bridge ISF Parity Error

The RapidIO-to-Bridge ISF LUT and BAR entries bridge RapidIO transactions to the Bridge ISF. For more information on the operation of the BARs and LUTs, see “Bridging Logical I/O Requests to the Bridge ISF” and “Bridging ISF Requests to RapidIO”.

Note that the “SREP Local Configuration Space Base Address CSR” (the LCS BAR) is also protected by parity.

The LCS BAR (“SREP Local Configuration Space Base Address CSR”), R2I BARs (“SREP R2I Base Address Register x LUT Control CSR” and “SREP R2I Base Address Register x Lower”) and LUTs (“SREP R2I Upper LUT Entry Translation Address Register” and “SREP R2I Lower LUT Entry Translation Address Register”) support a parity based error detection capability. The parity used is odd parity - the number of 1s in the data, added to the parity bit, results in an odd number of 1s. Each register in each BAR and LUT entry has a parity bit associated with it. The parity bit is updated when the registers are written. Parity is checked when a BAR or LUT entry is read for packet bridging purposes, and when the registers are read.

To disable LUT parity checking for the R2I LUTs, set the LUT_PAR_DIS bit in the “SREP R2I LUT and Parity Control Register”.

To disable BAR parity checking for the R2I BARs, R2I BAR Control and the “SREP Local Configuration Space Base Address CSR”, set the BAR_PAR_DIS bit in the “SREP R2I LUT and Parity Control Register”.

BAR parity errors are checked for in two stages. If the access hits in the LCS BAR, and the LCS BAR does not have a parity error, then the access is processed as a register request. If the access hits in the LCS BAR, and the LCS BAR does have a parity error, then the transaction is discarded and it receives an error response.

The second stage involves checking parity on the R2I BARs. If there is a parity error in any of the R2I BARs, then the transaction is discarded and it receives an error response.

When a parity error is detected, a parity error event is asserted by setting the L_R2I_PERR bit in the "SREP Logical and Transport Layer Error Detect CSR". Additional parity error information about the BAR parity error is latched in the "SREP R2I BAR and LUT Parity Error Status Register".

If a BAR parity error is not detected, and the RapidIO packet falls in a range supported by an R2I BAR, a RapidIO-to-Bridge ISF LUT entry is read. If the LUT entry used has bad parity, a parity error event is asserted by setting the L_R2I_PERR bit in the "SREP Logical and Transport Layer Error Detect CSR".

No access is performed for a RapidIO transaction whose handling depends on corrupted BARs or LUTs. In other words, the RapidIO transaction is handled as if it had accessed an area of memory that was not readable or writable. If the RapidIO transaction requires a response, an error response is sent to facilitate system debug.

When a LUT entry register is read using either a maintenance read request, an NREAD request, or from another block in the Tsi620, it is possible that the BAR or LUT entry has bad parity. If the LUT entry has bad parity on a register access, a parity error event is asserted by setting the L_R2I_PERR bit to 1 in the "SREP Logical and Transport Layer Error Detect CSR" and the LUT_PERR_RD and LUT_PERR bits are set in the "SREP R2I BAR and LUT Parity Error Status Register".

Accesses to LUT registers indicates successful completion regardless of parity errors.

When a BAR entry register is read and a parity error is detected, similar behavior occurs. The L_R2I_PERR bit is set in the "SREP Logical and Transport Layer Error Detect CSR" and the BAR_PERR_RD and BAR_PERR bits are set in the "SREP R2I BAR and LUT Parity Error Status Register".

The R2I BAR and/or LUT entry for which bad parity was detected is latched in the "SREP R2I BAR and LUT Parity Error Status Register". If more than one of the LCS BAR or R2I BAR entries cause parity errors to be detected, only the first BAR parity error is latched in the "SREP R2I BAR and LUT Parity Error Status Register".

The "SREP R2I BAR and LUT Parity Error Status Register" has a separate status bit for parity errors in the LCS BAR.

10.11.12.1 R2I LUT and BAR Parity Error Information Latching

When a BAR or LUT parity error is detected, the L_R2I_PERR bit is set in the "SREP Logical and Transport Layer Error Detect CSR". Unlike other errors, when the L_R2I_PERR status bit is set, the "SREP R2I BAR and LUT Parity Error Status Register" is locked. It is not necessary to set the L_R2I_PERR bit in the "SREP Logical and Transport Layer Error Logging Enable CSR" to lock the "SREP R2I BAR and LUT Parity Error Status Register".

When a parity error is detected, additional parity errors are ignored. Thus, a BAR parity error prevents the detection of a further BAR or LUT parity error until the L_R2I_PERR bit is cleared. The presence of a LUT parity error has the same effect.

If the L_R2I_PERR bit in the "SREP Logical and Transport Layer Error Logging Enable CSR" is set when a parity error is detected, then additional error information about the parity error is latched as identified in "RapidIO Logical/Transport Error Information Registers". All of the error information and status registers also are locked, and a RIO_LOG event occurs.

To generate a port-write or an interrupt when an R2I parity error occurs, the L_R2I_PERR bit in the "SREP Logical and Transport Layer Error Logging Enable CSR" must be set.

When the registers are locked for another enabled event, as described in "RapidIO Logical/Transport Error Information Registers", and the L_R2I_PERR bit in the "SREP Logical and Transport Layer Error Detect CSR" is not set, then parity error information is latched in the "SREP R2I BAR and LUT Parity Error Status Register". Additional parity error information is not latched in this case, regardless of the setting of the L_R2I_PERR bit in the "SREP Logical and Transport Layer Error Logging Enable CSR".

10.11.12.2 Testing R2I BAR Parity

It is possible to insert and verify BAR entry parity operation under software control. To insert a BAR entry parity error, perform the following:

1. Set the BAR_PAR_INV bit in the "SREP R2I LUT and Parity Control Register" to 1. This causes parity to be inverted when a BAR entry register is written.
2. Write a value to at least one of the "SREP R2I Base Address Register x LUT Control CSR" or "SREP R2I Base Address Register x Lower".
3. Repeat step 2 until all the BAR registers required for the test are corrupted.
4. Set the BAR_PAR_INV bit in the "SREP R2I LUT and Parity Control Register" to 0 to avoid corrupting BAR parity in further tests.

To cause a BAR parity error, ensure that the BAR_PAR_DIS bit is 0 in the "SREP R2I LUT and Parity Control Register" and then either send a Logical I/O transaction that uses a BAR with bad parity, or read a BAR register.

To clear the parity errors inserted, rewrite the BAR registers while the BAR_PAR_INV bit is set to 0.

10.11.12.3 Testing R2I LUT Entry Parity

It is possible to insert and verify LUT parity operation under software control. To insert a LUT entry parity error, perform the following:

1. Set the LUT_PAR_INV bit in the "SREP R2I LUT and Parity Control Register" to 1. This causes parity to be inverted when a LUT entry register is written.
2. Set the LUT_IDX field to the index of the LUT entry for which parity is to be corrupted.
3. Write a value to either the "SREP R2I Upper LUT Entry Translation Address Register" or "SREP R2I Lower LUT Entry Translation Address Register"

4. Repeat steps 2 and 3 until the LUT entries required for the test are corrupted.
5. Set the LUT_PAR_INV bit in the "SREP R2I LUT and Parity Control Register" to 0 to avoid corrupting LUT entry parity in further tests.

To cause a LUT entry parity error, ensure that the LUT_PAR_DIS bit is 0 in the "SREP R2I LUT and Parity Control Register" and then either send a Logical I/O transaction that uses a LUT entry with bad parity, or read a LUT entry register.

To clear the parity errors inserted, rewrite the LUT entries while the LUT_PAR_INV bit is set to 0.

10.11.13 RapidIO Read Denied Event

A Read Denied event is detected when an NREAD transaction attempts to access memory for which read privileges have not been granted.

The SREP supports denial of read access to RapidIO address space at the granularity of a LUT entry. The RD_EN bit in the "SREP R2I Lower LUT Entry Translation Address Register" controls whether or not an NREAD transaction is allowed to be processed. If an NREAD transaction attempts to access memory controlled by a LUT entry without read privileges enabled, a Read Denied Event is detected. For more information about the operation of the R2I LUTs, see "Bridging Logical I/O Requests to the Bridge ISF".

Detection of a Read Denied Event causes the L_NO_RD bit in the "SREP Logical and Transport Layer Error Detect CSR" to be set to 1. When the NO_RD_EN bit is set to 1 in the "SREP Logical and Transport Layer Error Logging Enable CSR", assertion of L_NO_RD causes the packet information to be latched in the Logical I/O Status registers, starting at "SREP Logical and Transport Layer Address Capture CSR".

For more information about the operation of the Logical/Transport Layer Error Information registers, see "Event Capture".

A Read Denied event causes the request packet to be dropped from the buffers of the SREP. An error response is also sent to notify the requestor of the error.

Creation of a Read Denied event can be completed by bridging a Bridge ISF READ to a RapidIO NREAD packet that is routed to the SREP. The NREAD request must have an address that hits in a BAR/LUT entry that has READ privileges disabled. The receipt of the request causes the detection of an L_NO_RD event.

10.11.14 RapidIO Write Denied Event

A Write Denied event is detected when an NWRITE, NWRITE_R or SWRITE transaction attempts to access memory for which write privileges have not been granted.

The SREP supports denial of write access to RapidIO address space at the granularity of a LUT entry. The WR_EN bit in the "SREP R2I Lower LUT Entry Translation Address Register" controls whether or not NWRITE, NWRITE_R and SWRITE transactions are allowed to be processed. If a write transaction attempts to access memory controlled by a LUT entry without write privileges enabled, a Write Denied Event is detected. For more information about the operation of the R2I LUTs, see "Bridging Logical I/O Requests to the Bridge ISF".

Detection of a Write Denied Event causes the L_NO_WR bit in the "SREP Logical and Transport Layer Error Detect CSR" to be set to 1. When the NO_WR_EN bit is set to 1 in the "SREP Logical and Transport Layer Error Logging Enable CSR", assertion of L_NO_WR causes the packet information to be latched in the Logical I/O Status registers, starting at "SREP Logical and Transport Layer Address Capture CSR".

For more information about the operation of the Logical/Transport Layer Error Information registers, see "Event Capture".

A Write Denied event causes the request packet to be dropped from the buffers of the SREP. An error response is also sent for NWRITE_R packets to notify the requestor of the error.

Creation of a Write Denied event can be completed by bridging a Bridge ISF WRITE to a RapidIO NWRITE packet that is routed to the SREP. The NWRITE request must have an address that hits in a BAR/LUT entry that has WRITE privileges disabled. The receipt of the request causes the detection of an L_NO_WR event.

10.11.15 RapidIO LUT Entry Boundary Crossing Event

A LUT Entry Boundary Crossing event is detected when any logical I/O packet crosses a LUT entry boundary.

RapidIO allows logical I/O packets to begin on any 8 byte aligned address, and to be up to 256 bytes. RapidIO-to-Bridge ISF translation occurs based on the address ranges control by BARs and LUT entries. It is therefore possible for a RapidIO transaction to begin in one LUT entry (the 'first' entry) and finish in another LUT entry (the 'second' entry).

Since the parameters in adjacent LUT entries are usually not consistent with each other, RapidIO transactions that span the address spaces controlled by different LUT entries can cause unintended behavior, such as reading or writing memory that does not exist or should not be accessed.

However, some applications make use of contiguous address spaces supported by adjacent LUT entries. Therefore, the SREP supports detection of RapidIO transactions that span LUT entry boundaries, but does not handle them as errors.

Packets that cross LUT boundaries are translated using the LUT entry for the packets' starting address. The parameters of the LUT entry that the packet finishes in, if any, are ignored.

A packet is judged to have crossed a LUT boundary if the packet's size field, added to the starting address, goes across a LUT boundary. This means that LUT Entry Boundary Crossing events may be falsely detected for writes, since a write may have less data that indicated by the size field.

SWRITE packets do not have a size. For purposes of detecting a LUT Entry Boundary Crossing event, all SWRITEs are assumed to have a size of 256 bytes. This may result in false detection of LUT Entry Boundary Crossing events.

Detection of a LUT Entry Boundary Crossing Event causes the L_LUT_BND bit in the "SREP R2I Event Status Register" to be set to 1. When the LUT_BND_EN bit is set to 1 in the "SREP R2I Event Status Logging Enable Register", assertion of L_LUT_BND causes the packet information to be latched in the Logical I/O Status registers, starting at "SREP Logical and Transport Layer Address Capture CSR".

For more information about the operation of the Logical/Transport Layer Error Information registers, see "Event Capture".

A LUT Entry Boundary Crossing event is used for status purposes - no error handling is triggered by a LUT Entry Boundary Crossing event.

Creation of a LUT Entry Boundary Crossing event can be completed by bridging a Bridge ISF READ or WRITE to a RapidIO NREAD/NWRITE packet that is routed to the SREP. The NREAD/NWRITE request must have an address/size combination that crosses a LUT entry boundary. receipt of the request causes the detection of an L_LUT_BND event.

10.12 Bridge ISF Error Conditions

In the Bridge ISF to RapidIO direction, the SREP can detect the logical layer error events specified in the *RapidIO Interconnect Specification (Revision 1.3)* Part 8: Error Management Extensions Specification, as well as a number of implementation-specific error events.

Note that some Bridge ISF logical layer error events have implications for transactions bridged from RapidIO-to-Bridge ISF, and others are RapidIO specific. The detection of an error event usually results in the completion of a transaction with an error response. The error response may be sent on the Bridge ISF or RapidIO.

All Bridge ISF Logical Layer errors detected results in information being latched in the Bridge ISF error status registers. For more information on the information latched, see "SREP Bridge ISF Event Information Registers".

To create Bridge ISF errors for software test purposes, perform the following steps:

1. Disable all Bridge ISF error events in the "SREP ISF Logical Error Logging Enable CSR"
2. Clear all Bridge ISF error events in the "SREP ISF Logical Error Detect CSR"
3. Disable Bridge ISF error event reporting in the "SREP Interrupt on ISF Event Control Register" and "SREP Port-Write Transmit on ISF Event Control Register"
4. Write data into the "SREP ISF Logical Error Upper Attributes Capture CSR", "SREP ISF Logical Error Middle Attributes Capture CSR", and "SREP ISF Logical Error Lower Attributes Capture CSR".
5. Enable the Bridge ISF error event.
6. Trigger the Bridge ISF error by writing to the "SREP ISF Logical Error Generate CSR"
7. Enable event reporting for the Bridge ISF error.

At this point, the Bridge ISF error should be reported to software.

10.12.1 Precedence of Bridge ISF Errors

The SREP checks for errors in a specific order. The order in which the errors are detected is listed as follows:

1. Transfer Error Acknowledge
2. ECC Error, R2I
3. ECC Error, RIO I2R
4. ECC Error, Bridge ISF I2R
5. I2R Transaction Time-to-Live Expired
6. R2R Time-to-Live Expired
7. Register Access Response Time-to-Live Expired
8. Bridge ISF Response Timeout
9. Unsupported Bridge ISF Transaction
10. Illegal Bridge ISF Response
11. Bridge ISF Error Response
12. Bridge ISF BAR Parity Error
13. Bridge ISF OOB Request Event
14. Bridge ISF Unsegmentable Request
15. I2R LUT Entry Parity Error
16. Bridge ISF Write Denied
17. Bridge ISF Read Denied
18. I2R LUT Entry Boundary Crossing
19. Bridge ISF Byte Enables Discontiguous

Transfer Error Acknowledge (TEA) and ECC errors are two Bridge ISF related errors that are described in this section. A TEA occurs when the Bridge ISF completes a requested transfer with an error acknowledgement (see “Transfer Error Acknowledge Event (TEA)”).

An error that affects the Logical Layer design usually is the ECC error. An ECC error occurs when a data corruption is detected internal to the SREP (see “ECC Error in Data Path” and “Data Path ECC”).

10.12.2 Bridge ISF Response Timeout Events

To facilitate robust system design, Bridge ISF response timeouts can be enforced on RapidIO NREAD transactions sent to the Bridge ISF.

The timeout is programmed using the standard “SREP ISF Response Timeout Register”. The TVAL field of the “SREP ISF Response Timeout Register” controls the timeout interval, which must be tracked for every transaction. Each transaction has a 3 bit ‘packet counter’ associated with it, which counts down from 7 when a Bridge ISF request is issued. Once a ‘packet counter’ reaches 0, and the ‘timeout counter’ (described next) reaches 0, a response timeout is detected. A ‘tick’ is the unit interval for decrementing the ‘packet counter’.

The ‘timeout counter’ drives the decrementing of the ‘packet counters’. The ‘timeout counter’ counts down from TVAL. The timeout period for a packet varies depending upon when the packet was received with respect to the timeout counter value. If the packet was received when the “timeout counter” was close to zero, the first ‘tick’ of the packet counter will be much shorter than expected. The ‘timeout counter’ must expire 8 times before a packet times out.

A write to the “SREP ISF Response Timeout Register” takes effect immediately by reloading the ‘timeout counter’. The timeout period is unpredictable when the ‘timeout counter’ is reloaded.

The “timeout counter” decrements at the rate of the SREP Bridge ISF reference clock, divided by 4.

- If the reference clock is operating at 156.25 MHz, increasing TVAL by 1 adds 204.8 nsec to the timeout interval.
- If the reference clock is operating at 125 MHz, increasing TVAL by 1 adds 256 nsec to the timeout interval.

When a RapidIO request packet has not received a response from the Bridge ISF in the timeout interval, a Bridge ISF Response Timeout event is detected. Response Timeout events cause the I_RESP_TO bit in the “SREP ISF Logical Error Detect CSR” to be set. Information about the request packet is latched in the Bridge ISF Error Information registers, starting with the “SREP ISF Logical Error Upper Attributes Capture CSR”.

For more information about the operation of the Bridge ISF Error Information registers, see “SREP Bridge ISF Event Information Registers”.

a Bridge ISF Response Timeout event causes the request packet to be dropped from the Bridge ISF buffers of the SREP. A RapidIO error response is sent if the ERESP_DIS bit is 0 in the “SREP R2I RapidIO Miscellaneous Control CSR”. If the ERESP_DIS bit is 1, then a RapidIO error response is not sent. A RapidIO error response is sent to complete the RapidIO NREAD transaction that was bridged to the Bridge ISF request packet.

For deterministic system behavior, ensure that the Bridge ISF Response Timeout interval is set to be at least twice what the expected maximum response time.

10.12.2.1 Disabling Response Timeouts

By default, detection of response timeouts is enabled. However, for lab debug purposes, it is sometimes useful to disable timeouts for responses.

To disable response timeout events, set the TVAL field to 0 in the “SREP ISF Response Timeout Register”. Conversely, when TVAL is not zero, response timeouts are enabled.

10.12.2.2 Creating Response Timeout Events

a Bridge ISF response timeout event can be created by bridging a read request from RapidIO to Bridge ISF for which the Bridge ISF entity will not respond. a Bridge ISF response timeout event will occur. Alternatively, the response timeout is set to a very short value. A request can be bridged to a Bridge ISF port that responds too slowly.

10.12.3 Bridge ISF Unsupported Transaction Events

The ‘Unsupported Transaction’ event means that the SREP has received a Bridge ISF request that it does not know how to process.

Each Bridge ISF request has a request command type. Many request commands are not supported by the SREP. A summary of those Bridge ISF request command types that are not supported by the SREP is contained in Table 43. Reception of a Bridge ISF request with an unsupported request command type causes an ‘Unsupported Transaction’ event to be detected.

Additionally, the ‘Type’ and ‘Mod’ Bridge ISF fields describe the format of response transactions. A description of the unsupported ‘Type’ and ‘Mod’ combinations is provided in Table 44. If the ‘Type’ and ‘Mod’ fields describe an unsupported response format, then an ‘Unsupported Transaction’ event is detected.

Requests whose command type is unsupported are dropped. Responses whose format is not supported are dropped.

Reception of an unsupported request command type causes the I_UNSUP_TRANS bit in the “SREP ISF Logical Error Detect CSR” to be set to 1. When the UNSUP_TRANS_EN bit is set to 1 in the “SREP ISF Logical Error Logging Enable CSR”, assertion of I_UNSUP_TRANS causes the packet information to be latched in the Bridge ISF Error Status registers, starting at “SREP ISF Logical Error Upper Attributes Capture CSR”.

For more information about the operation of the ISF Error Information registers, see “SREP Bridge ISF Event Information Registers”.

Table 43: Unsupported Transaction Request Type Encoding

Request Command Type (Cmd)	Request Command Description
0010	I/O Read
0011	I/O Write
0100	Memory Read Block right justified
0101	Memory Write Block right Justified
1010	Configuration Read
1011	Configuration Write
1100	Split Completion
1101	DAC – Dual Address Cycle

Table 44: Unsupported Transaction Type and Mod Encoding

Type[1]	Type[0]	Mod	Rsp Type
0	1	1	0b01 0b11
1	0	N/A	N/A

The only method to create a Bridge ISF Unsupported Transaction event is to write appropriate data to the Bridge ISF Logical Layer Error registers, and use the "SREP ISF Logical Error Generate CSR" to generate a Bridge ISF Unsupported Transaction event.

10.12.4 Bridge ISF Unexpected Response

An 'Unexpected Response' event is detected when a Bridge ISF response is received for a Bridge ISF Transaction Tag that is not currently in use.

The Tag field (also called a Transaction Tag or Identifier) is a 4-bit field that is assigned by the Bridge ISF when the transaction is assembled and enqueued to be sent to the Bridge ISF transaction destination. An 'Unexpected Response' event is detected when a response packet is received with a Tag that does not match any request tracked by the SREP Bridge ISF gasket.

Reception of an unexpected response packet causes the I_UNEXP_RESP bit in the "SREP ISF Logical Error Detect CSR" to be set to 1. When the UNEXP_RESP_EN bit is set to 1 in the "SREP ISF Logical Error Logging Enable CSR", assertion of L_UNEXP_RESP causes the packet information to be latched in the Logical I/O Status registers, starting at "SREP ISF Logical Error Upper Attributes Capture CSR".

For more information about the operation of the Bridge ISF Error Information registers, see "SREP Bridge ISF Event Information Registers".

An unexpected response packet is dropped. No other fault handling needs to be completed for an unexpected response packet.

The only method to create a Bridge ISF Unexpected Response event is to write appropriate data to the Bridge ISF Logical Layer Error registers, and use the "SREP ISF Logical Error Generate CSR" to generate a Bridge ISF Unexpected Response event.

10.12.5 Bridge ISF Error Response Events

An 'Error Response' event is detected when a response which indicates an error status is received for an existing Bridge ISF request.

The status of Bridge ISF response transactions is encoded using the Type and Mod fields. If the Type[0] bit is 1 (indicating a response transaction), and the Mod field is 1, then the response has no data for a read transaction. The lack of data indicates an error in the Bridge ISF target of the transaction.

Reception of an response transaction with an error status causes the I_ERR_RESP bit in the "SREP ISF Logical Error Detect CSR" to be set to 1. When the ERR_RESP_EN bit is set to 1 in the "SREP ISF Logical Error Logging Enable CSR", assertion of I_ERR_RESP causes the request information, not the error response, to be latched in the Bridge ISF Error Information registers.

For more information about the operation of the Bridge ISF Error Information registers, see "SREP Bridge ISF Event Information Registers".

A Bridge ISF error response event causes the Bridge ISF request packet to be dropped from the buffers of the SREP. A RapidIO Error response is sent to complete the RapidIO transaction that was bridged to the Bridge ISF request.

a Bridge ISF Error Response event can be created by bridging a RapidIO NREAD to a Bridge ISF target that will return an Error response for the Bridge ISF transaction. Another method to create a Bridge ISF Error Response event is to write appropriate data to the Bridge ISF Logical Layer Error registers, and use the "SREP ISF Logical Error Generate CSR" to generate a Bridge ISF Error Response event.

10.12.6 Bridge ISF OOB Request Event

An 'Out Of Bounds Request' event is detected when a valid Bridge ISF request is received with an address that does not match any enabled I2R BAR.

Bridge ISF requests contain an Address field that identifies the 64-bit address which the packet should affect. Each I2R BAR, and the I2R Doorbell BAR, identifies a range of Bridge ISF addresses that the SREP must respond to. For more information on how the BARs are used, see "Bridging ISF Requests to RapidIO".

To transmit RapidIO requests, the MAST_EN bit of the "SREP General Control CSR" must be set to 1.

The I2R BARs are not enabled until the MAST_EN bit of the "SREP General Control CSR" is set to 1. The receipt of a Bridge ISF request when the MAST_EN bit is set to 0 causes the following behavior:

- The Bridge ISF request is dropped
- a Bridge ISF error response is sent, if necessary
- a Bridge ISF OOB Request event is detected

Reception of an OOB Request causes the I_OOB bit in the "SREP ISF Logical Error Detect CSR" to be set to 1. When the OOB_EN bit is set to 1 in the "SREP ISF Logical Error Logging Enable CSR", assertion of L_OOB causes the packet information to be latched in the Logical I/O Status registers, starting at "SREP ISF Logical Error Upper Attributes Capture CSR".

For more information about the operation of the Bridge ISF Error Information registers, see "SREP Bridge ISF Event Information Registers".

A OOB Request event causes the request packet to be dropped from the buffers of the SREP. If the request requires a response (that is, a read request), an error response is sent to complete the Bridge ISF request. The error response is sent at the Bridge ISF priority programmed in the ERROR field of the "SREP R2I ISF Request Priority Control Register".

To generate a Bridge ISF OOB Request event, send a request on Bridge ISF to the SREP at a Bridge ISF address for which no Bridge ISF BAR is configured to respond, for example, when all Bridge ISF BARs are disabled. Another method to create a Bridge ISF Error Response event is to write appropriate data to the Bridge ISF Logical Layer Error registers, and use the "SREP ISF Logical Error Generate CSR" to generate a Bridge ISF OOB Request event.

10.12.7 Bridge ISF Unsegmentable Request

a Bridge ISF Unsegmentable Request event is detected when it is not possible to translate a Bridge ISF transaction to a RapidIO maintenance transaction or Doorbell transaction due to the size or alignment of the Bridge ISF transaction.

The SREP supports generation of RapidIO maintenance read and maintenance write requests that are 4 bytes at 4-byte aligned addresses. Bridge ISF requests support access sizes that are different from 4 bytes. Bridge ISF requests also support 4 byte accesses that are not aligned to a 4 byte aligned address. An Unsegmentable Request event is detected when a Bridge ISF transaction is configured to be translated to a RapidIO maintenance read or write when the Bridge ISF transaction is not 4 bytes or is not aligned to a 4-byte address.

a Bridge ISF Unsegmentable Request event is also detected when a Bridge ISF transaction hits in the Bridge ISF Doorbell BAR and at least one of the following is true:

- Bridge ISF Write request is not 4 bytes.
- The Bridge ISF Write request is not completed at an 8 byte aligned address boundary
- The Bridge ISF Write request occurs at an address that would result in the doorbell being sent at priority 3 (an illegal priority, since doorbells require RapidIO responses) or undefined destination ID size (TT code of 2 or 3).
- The Bridge ISF Memory Write Burst request does not have all four least significant bytes selected
- The Bridge ISF transaction is a Read request

Reception of a Bridge ISF Unsegmentable Request causes the I_UNSAR_REQ bit in the "SREP ISF Logical Error Detect CSR" to be set to 1. When the UNSAR_REQ_EN bit is set to 1 in the "SREP ISF Logical Error Logging Enable CSR", assertion of I_UNSAR_REQ causes the Bridge ISF request information to be latched in the Logical I/O Status registers, starting at "SREP ISF Logical Error Upper Attributes Capture CSR".

For more information about the operation of the Bridge ISF Error Information registers, see "SREP Bridge ISF Event Information Registers".

a Bridge ISF Unsegmentable Request event causes the Bridge ISF request packet to be dropped from the buffers of the SREP. If the request required a response (a Bridge ISF read), a Bridge ISF error response is sent to complete the Bridge ISF request.

To generate a Bridge ISF Unsegmentable Request event, send a request on Bridge ISF to the SREP of a size greater than 4 bytes, or at an address that is not a multiple of 4 bytes, or both, which hits in a LUT entry that will translate the request to a RapidIO Maintenance transaction. a Bridge ISF Unsegmentable Request can also be created by sending a similar request to the Bridge ISF Doorbell BAR. Another method to create a Bridge ISF Unsegmentable Request event is to write appropriate data to the Bridge ISF Logical Layer Error registers, and use the "SREP ISF Logical Error Generate CSR" to generate a Bridge ISF Unsegmentable Request event.

10.12.8 Bridge ISF-to-RapidIO Parity Error

The Bridge ISF-to-RapidIO BARs and LUTs convert Bridge ISF transactions to RapidIO. For more information on the operation of the BARs and LUTs, see "Bridging ISF Requests to RapidIO".

The Bridge ISF-to-RapidIO BARs are comprised of the following registers:

- "SREP I2R Base Address Register x LUT Entry CSR"
- "SREP I2R Base Address Register x Upper"
- "SREP I2R Base Address Register x Lower"
- "SREP I2R Doorbell BAR Upper"
- "SREP I2R Doorbell BAR Lower"

The Bridge ISF-to-RapidIO LUTs are comprised of the following registers:

- "SREP I2R Upper LUT Entry Translation Register"
- "SREP I2R Lower LUT Entry Translation Address Register"
- "SREP I2R LUT Translation Parameters Register"

The Bridge ISF-to-RapidIO BARs and LUTs support a parity based error detection capability. The parity used is odd parity - the number of 1s in the data, added to the parity bit, results in an odd number of 1s. Each register in each BAR and LUT entry has a parity bit associated with it. The parity bit is updated when the registers are written. Parity is checked when a BAR or LUT entry is read for packet bridging purposes, and when the registers are read .

To disable BAR parity checking for the Bridge ISF-to-RapidIO BARs, set the BAR_PAR_DIS bit in the "SREP I2R LUT and BAR Parity Control Register".

To disable LUT entry parity checking for the Bridge ISF-to-RapidIO LUTs, set the LUT_PAR_DIS bit in the "SREP I2R LUT and BAR Parity Control Register".

To transmit RapidIO requests, the MAST_EN bit of the "SREP General Control CSR" must be set to 1.

When a Bridge ISF transaction hits in any BAR, the parity of the Doorbell BAR and all I2R BARs is checked. If any BAR entry has incorrect parity, a parity error event is asserted by setting the I_I2R_PERR bit in the "SREP ISF Logical Error Detect CSR" and the BAR_PERR bit in the "SREP I2R BAR and LUT Parity Error Status Register". If the Doorbell BAR had a parity error, the DB_BAR bit is set and the BAR_IDX field is cleared to 0. If an I2R BAR had the parity error, the index of the BAR entry that failed is latched in the BAR_IDX field of the "SREP I2R BAR and LUT Parity Error Status Register". The Bridge ISF packet is not bridged to RapidIO. If the Bridge ISF transaction requires a Bridge ISF response, a Bridge ISF error response is sent to facilitate system debug.

If no BAR entry has a parity error, a Bridge ISF-to-RapidIO LUT entry is read. If the LUT entry used has bad parity, a LUT entry parity error event is asserted by setting the I_I2R_PERR bit in the "SREP ISF Logical Error Detect CSR" and the LUT_PERR bit in the "SREP I2R BAR and LUT Parity Error Status Register". It the LUT entry has a parity error, the Bridge ISF packet is not bridged to RapidIO. If the Bridge ISF transaction requires a Bridge ISF response, a Bridge ISF error response is sent to facilitate system debug.

When a BAR entry register is read using a maintenance read request, an NREAD request, or from another block in the Tsi620, it is possible that the BAR entry has bad parity. A BAR parity error event is asserted by setting the I_I2R_PERR bit to 1 in the "SREP ISF Logical Error Detect CSR". The BAR_PERR_RD and BAR_PERR bits are set in the "SREP I2R BAR and LUT Parity Error Status Register".

When a LUT entry register is read using either a maintenance read request, an NREAD request, or from another block in Tsi620, it is possible that the LUT entry has bad parity. A LUT parity error event is asserted by setting the I_I2R_PERR bit to 1 in the "SREP ISF Logical Error Detect CSR". The LUT_PERR_RD and LUT_PERR bits are set in the "SREP I2R BAR and LUT Parity Error Status Register".

The BAR and/or LUT entry for which bad parity was detected is latched in the "SREP I2R BAR and LUT Parity Error Status Register". If two different BAR entries cause parity errors to be detected, only the first is latched in the "SREP I2R BAR and LUT Parity Error Status Register".

10.12.8.1 I2R LUT and BAR Parity Error Information Latching

When a BAR or LUT parity error is detected, the L_I2R_PERR bit is set in the "SREP ISF Logical Error Detect CSR". Unlike other errors, when the L_I2R_PERR status bit is set, the "SREP I2R BAR and LUT Parity Error Status Register" is locked. It is not necessary to set the L_I2R_PERR bit in the "SREP ISF Logical Error Logging Enable CSR" to lock the "SREP I2R BAR and LUT Parity Error Status Register".

When a parity error is detected, further parity errors are ignored. Thus, a BAR parity error prevents detection of a further BAR or LUT parity error until the L_I2R_PERR bit is cleared. The presence of a LUT parity error has the same effect.

If the L_I2R_PERR bit in the "SREP ISF Logical Error Logging Enable CSR" is set when a parity error is detected, then additional error information about the parity error is latched as identified in "SREP Bridge ISF Event Information Registers". All of the error information and status registers are locked, and an ISF_LOG event occurs.

To generate a port-write or an interrupt when an R2I parity error occurs, the L_I2R_PERR bit in the "SREP ISF Logical Error Logging Enable CSR" must be set.

If a parity error is detected when the error information and status registers are locked, but the L_I2R_PERR bit in the "SREP ISF Logical Error Detect CSR" is not set, then parity error information is latched in the "SREP I2R BAR and LUT Parity Error Status Register". Additional parity error information is not latched in this case, regardless of the setting of the L_I2R_PERR bit in the "SREP ISF Logical Error Logging Enable CSR".

10.12.8.2 Testing I2R BAR Parity

It is possible to insert and verify BAR entry parity operation under software control. To insert a BAR entry parity error, perform the following:

1. Set the BAR_PAR_INV bit in the "SREP I2R LUT and BAR Parity Control Register" to 1. This causes parity to be inverted when a BAR entry register is written.
2. Write a value to at least one of the "SREP I2R Base Address Register x LUT Entry CSR", "SREP I2R Base Address Register x Upper" or "SREP I2R Base Address Register x Lower".
3. Repeat step 2 until all the BAR registers required for the test are corrupted.
4. Set the BAR_PAR_INV bit in the "SREP I2R LUT and BAR Parity Control Register" to 0 to avoid corrupting BAR parity in further tests.

To cause a BAR parity error, ensure that the BAR_PAR_DIS bit is 0 in the "SREP I2R LUT and BAR Parity Control Register" and then either send a Bridge ISF transaction that uses a BAR with bad parity, or read a BAR register.

To clear the parity errors inserted, rewrite the BAR registers while the BAR_PAR_INV bit is set to 0.

10.12.8.3 Testing I2R LUT Entry Parity

It is possible to insert and verify LUT entry parity operation under software control. To insert a LUT entry parity error, perform the following:

1. Set the LUT_PAR_INV bit in the "SREP I2R LUT and BAR Parity Control Register" to 1. This causes parity to be inverted when a LUT entry register is written.
2. Set the LUT_IDX field to the index of the LUT entry for which parity is to be corrupted.
3. Write a value to either the "SREP I2R Upper LUT Entry Translation Register", "SREP I2R Lower LUT Entry Translation Address Register" or "SREP I2R LUT Translation Parameters Register".
4. Repeat steps 2 and 3 until the LUT entries required for the test are corrupted.
5. Set the PT_PAR_INV bit in the "SREP I2R LUT and BAR Parity Control Register" to 0 to avoid corrupting LUT entry parity in further tests.

To cause a LUT entry parity error, ensure that the LUT_PAR_DIS bit is 0 in the "SREP I2R LUT and BAR Parity Control Register" and then either send a Bridge ISF transaction that uses a LUT entry with bad parity, or read a I2R LUT entry register.

To clear the parity errors inserted, rewrite the LUT entries while the LUT_PAR_INV bit is set to 0.

10.12.9 ECC Error in Data Path

An Error Correcting Code (ECC) is tracked and checked for each 8 byte datum sent or received on Bridge ISF. ECC coverage includes the data of RapidIO packets, and the data and header for Bridge ISF transactions. This allows detection of data corruption within the SREP. ECC is checked at three points:

- When data is read from the R2I Data Buffers, in preparation for transmission to the Bridge ISF (R2I).
- When requests and data are received from the Bridge ISF and placed into the I2R Data Buffers (ISF I2R)
- When data is read from the I2R Data Buffers, in preparation for transfer to the Retry Buffers. (RIO I2R)

ECC allows single bit errors to be corrected, and detects but cannot correct double bit errors.

Detection of three or more incorrect bits in an 8 byte datum is not guaranteed.

The following transactions do not have ECC coverage, even though small amounts of data are associated with them:

- Transactions in the R2R Queue
- Register access Write requests
- Register access Read responses
- Port Write transactions
- Doorbell Requests

The following transactions do not have ECC coverage, since no data is associated with them:

- NREAD Requests
- NWRITE_R responses
- Maintenance Read requests
- Maintenance Write responses
- Doorbell Responses

If an ECC error is detected in data read from the R2I or I2R Data Buffers, or in a transaction received from the Bridge ISF, then the location of the ECC check, along with the ECC syndrome for the error, is locked into the "SREP ISF ECC Error Status Register" by setting one of the ECC_INB, ECC_OUTB, or ECC_ISF bits. For data read from the R2I and I2R data buffers, the data buffer index is locked in the BUFF_IDX field. For transactions received from the Bridge ISF, both the upper and lower syndromes and data are latched. The UNCORR bit is set if the ECC error was uncorrectable.

The BUFF_IDX field allows software to determine whether or not a buffer is failing consistently.

Correctable ECC errors do not change the handling of packets or transactions.

In the RapidIO-to-Bridge ISF direction, an uncorrectable ECC error detected when reading data from the R2I data buffers does not change the completion of the transaction. The 'bad' ECC is forwarded with the data, to allow the receiving block to detect and handle the ECC error.

For transactions received from the Bridge ISF, uncorrectable ECC errors in the first datum of the transaction must be handled differently from those that occur in later datums. If an uncorrectable ECC error is detected in the upper bits of a Bridge ISF transaction, or in the lower bits of a Bridge ISF request that has an address in those bits, then the ISF_REQ bit is set in the "SREP ISF ECC Error Status Register". Uncorrectable errors in the upper bits of a Bridge ISF transaction, or in the lower bits of a Bridge ISF request that has an address, result in the transaction being discarded.

In the Bridge ISF-to-RapidIO direction, ECC errors may also be detected in data received from the Bridge ISF to be stored in the I2R Data Buffers, or when data is read from the I2R Data Buffers to be transferred to the Retry Buffers. If an uncorrectable ECC error is detected in data received from the Bridge ISF, or when data is read from the I2R Data buffers, the following steps are taken to handle the event in hardware. No Bridge ISF error response is sent since the transaction is either a Bridge ISF read response or a Bridge ISF write transaction, neither of which require Bridge ISF responses.

- If the uncorrectable ECC error was detected in data received from the Bridge ISF, the transaction is processed normally to the point of transferring packets to the MAC. Note that ECC is checked for both the upper and lower bits of the data phase.
- Each packet transfer to the MAC is STOMPed. The MAC discards the packet from its buffers.
- Once the transaction is completely processed, the Bridge ISF transaction is discarded from the I2R Data Buffers.

In the special case that an uncorrectable ECC error is detected in the lower bits of a Bridge ISF response without data, the following occurs:

- The uncorrectable ECC error is latched
- The RapidIO response packet is still sent and not STOMPed

ECC errors may be detected at multiple locations on the same clock cycle. When this occurs, only one ECC error is latched. The order of precedence of the status bits in "SREP ISF ECC Error Status Register", from highest to lowest, is: ECC_INB, ECC_OUTB, ECC_ISF.

Information about ECC errors is latched in the "SREP ISF ECC Error Status Register" regardless of whether or not ECC errors are enabled in the "SREP ISF Logical Error Logging Enable CSR".

ECC error information is not locked in the registers until the ECC_ERR_EN bit is set in the "SREP ISF Logical Error Logging Enable CSR".

10.12.9.1 Testing ECC

The "SREP ISF ECC Control Register" controls the ECC codes generated for data inserted into the R2I Data Buffers. This allows software to insert correctable and uncorrectable ECC errors:

- ECC_INV_EN: Enable inversion of ECC for one datum. This bit is 1 if an ECC inversion request is being processed. It clears to 0 when the ECC inversion request is complete.
- ECC_INV_TYPE: Either invert the entire ECC code resulting in an uncorrectable ECC error, or invert one bit of the ECC code which is a correctable error.
- ECC_INV_DLY: The number of Bridge ISF datums to count before inverting the ECC.

It is not possible to insert an ECC error in the first datum of an R2I Bridge ISF Response transaction. ECC_INV_DLY must be at least 1 to insert an ECC error into R2I Bridge ISF Response transactions.

Note that this facility only allows insertion of ECC errors in the RapidIO-to-Bridge ISF direction. ECC errors inserted in this way may be detected in the R2I direction when data is being transferred from the R2I Data Buffers to the Bridge ISF.

ECC testing must be performed with Bridge ISF transactions that do not require decomposition; that is, one RapidIO packet for the Bridge ISF request. Using decomposed Bridge ISF transactions causes an undeterministic corruption of packets.

To test detection of ECC errors in the I2R direction, route the corrupted Bridge ISF request generated by the SREP back to the SREP. Corruption of data in the I2R direction can be detected at the time the data is received from the Bridge ISF. If ECC error detection is disabled at this point, then the corruption can be detected at the time the data is being transferred out of the I2R buffers.

In "SREP ISF ECC Control Register", the mode of operation where OUTB_ECC_CHK_EN is 1 while ISF_ECC_CHK_EN is 1 is not supported.

The OUTB_ECC_CHK_EN and ISF_ECC_CHK_EN bits should be enabled or disabled together during normal operation.

For ECC error insertion/testing purposes, OUTB_ECC_CHK_EN may be 0 while ISF_ECC_CHK_EN is 1.

10.12.10 Read Denied Event

A Read Denied event is detected when a Bridge ISF read transaction attempts to access memory for which read privileges have not been granted.

The SREP supports denial of read access to Bridge ISF address space at the granularity of a Bridge ISF LUT entry. The RD_EN bit in the "SREP I2R Lower LUT Entry Translation Address Register" controls whether or not an NREAD transaction is allowed to be processed. If an NREAD transaction attempts to access memory controlled by a LUT entry without read privileges enabled, a Read Denied Event is detected. For more information about the operation of the I2R LUTs, see "Bridging ISF Requests to RapidIO".

Detection of a Read Denied Event causes the I_NO_RD bit in the "SREP ISF Logical Error Detect CSR" to be set to 1. When the NO_RD_EN bit is set to 1 in the "SREP ISF Logical Error Logging Enable CSR", assertion of I_NO_RD causes the packet information to be latched in the Bridge ISF Logical Error Information registers, starting at "SREP ISF Logical Error Upper Attributes Capture CSR".

For more information about the operation of the Bridge ISF Logical Error Information registers, see "SREP Bridge ISF Event Information Registers".

A Read Denied event causes the Bridge ISF request packet to be dropped from the buffers of the SREP. a Bridge ISF error response is sent to notify the requestor of the error.

To create a Bridge ISF Read Denied event, send a Bridge ISF transaction to a LUT entry that has read privileges disabled.

10.12.11 Write Denied Event

A Write Denied event is detected when a Bridge ISF write transaction attempts to access memory for which write privileges have not been granted.

The SREP supports denial of write access to Bridge ISF address space at the granularity of a LUT entry. The WR_EN bit in the "SREP I2R Upper LUT Entry Translation Register" controls whether or not Bridge ISF write transactions are allowed to be processed. If a write transaction attempts to access memory controlled by a LUT entry without write privileges enabled, a Write Denied Event is detected. For more information about the operation of the I2R LUTs, see "Bridging ISF Requests to RapidIO".

Detection of a Write Denied Event causes the I_NO_WR bit in the "SREP ISF Logical Error Detect CSR" to be set to 1. When the NO_WR_EN bit is set to 1 in the "SREP ISF Logical Error Logging Enable CSR", assertion of I_NO_WR causes the packet information to be latched in the Bridge ISF Logical Error Information registers, starting at "SREP ISF Logical Error Upper Attributes Capture CSR".

For more information about the operation of the Bridge ISF Logical Error Information registers, see "SREP Bridge ISF Event Information Registers".

A Write Denied event causes the Bridge ISF request packet to be dropped from the buffers of the SREP. No error response needs to be sent.

To create a Bridge ISF Write Denied event, send a Bridge ISF transaction to a LUT entry that has write privileges disabled.

10.12.12 Bridge ISF LUT Entry Boundary Crossing Event

A LUT Entry Boundary Crossing event is detected when any Bridge ISF request crosses a LUT entry boundary.

Bridge ISF transactions may begin on any byte address, and to be up to 128 64-bit data phases. Bridge ISF-to-RapidIO translation occurs based on the address ranges controlled by BARs and LUT entries. It is therefore possible for a Bridge ISF transaction to begin in one LUT entry (the ‘first’ entry) and finish in another LUT entry (the ‘second’ entry).

Since the parameters in adjacent LUT entries are usually not consistent with each other, Bridge ISF transactions that span the address spaces controlled by different LUT entries can cause unintended behavior, such as reading or writing RapidIO memory space that does not exist or should not be accessed.

However, some applications make use of contiguous address spaces supported by adjacent LUT entries. Therefore, the SREP supports detection of Bridge ISF transactions that span LUT entry boundaries, but does not handle them as errors..

Transactions that cross LUT boundaries are translated using the LUT entry for the transactions’ starting address. The parameters of the LUT entry that the transaction finishes in, if any, are ignored.

Detection of Bridge ISF LUT Entry Boundary Crossing Events is exact, since the size of each Bridge ISF request is communicated in the Bridge ISF header.

Detection of a Bridge ISF LUT Entry Boundary Crossing Event causes the I_LUT_BND bit in the “SREP ISF Logical Error Detect CSR” to be set to 1. When the LUT_BND_EN bit is set to 1 in the “SREP ISF Logical Error Logging Enable CSR”, assertion of I_LUT_BND causes the packet information to be latched in the Bridge ISF Logical Error Information registers, starting at “SREP ISF Logical Error Upper Attributes Capture CSR”.

For more information about the operation of the Bridge ISF Logical Error Information registers, see “SREP Bridge ISF Event Information Registers”.

a Bridge ISF LUT Entry Boundary Crossing event is used for status purposes only. No error handling is triggered by a Bridge ISF LUT Entry Boundary Crossing event.

To create a Bridge ISF LUT Entry Boundary Crossing event, send a Bridge ISF transaction to the SREP with a size and alignment that extends beyond a LUT entry boundary.

10.12.13 Bridge ISF Byte Enables Discontiguous Event

a Bridge ISF Byte Enables Discontiguous event is detected when a Bridge ISF Write Burst transaction is being SARed as if it were a Bridge ISF Write Block transaction (BST_2_BLK), and the Bridge ISF Write Burst transaction has discontiguous byte enables in the head or body of the transaction. For more information on Bridge ISF Write Burst and Write Block SARing, see "Bridge ISF Memory Write Burst Segmentation" and "Bridge ISF Memory Write Block Segmentation".

a Bridge ISF Write Burst transaction can be SARed as if it were a Bridge ISF Write Block transaction (see "Bridge ISF Memory Write Burst Segmentation"). The Bridge ISF Write Burst transaction has byte enables associated with each 8 bytes of the transaction. When SARed as a Bridge ISF Write Block transaction, these byte enables are assumed to be contiguous within the head and body of the transaction. If the byte enables are not contiguous, bytes will be written which were not expected.

Detection of a Bridge ISF Byte Enables Discontiguous Event causes the I_BE_DISCONT bit in the "SREP ISF Logical Error Detect CSR" to be set to 1. When the BE_DISCONT_EN bit is set to 1 in the "SREP ISF Logical Error Logging Enable CSR", assertion of I_BE_DISCONT causes the packet information to be latched in the Bridge ISF Logical Error Information registers, starting at "SREP ISF Logical Error Upper Attributes Capture CSR".

For more information about the operation of the Bridge ISF Logical Error Information registers, see "SREP Bridge ISF Event Information Registers".

a Bridge ISF Byte Enables Discontiguous event is used for status purposes only. No error handling is triggered by a Bridge ISF Byte Enables Discontiguous event.

To create a Bridge ISF Byte Enables Discontiguous event, send a Bridge ISF Write Burst transaction to the SREP which hits in a LUT entry with BST_2_BLK set to 1, and has discontiguous byte enables. Alternatively, set the I_BE_DISCONT in the "SREP ISF Logical Error Generate CSR".

10.12.14 R2R Time-to-Live Expired

An R2R Time-to-Live Expired Event is detected when the response packet is buffered in the R2R Queue for a period longer than that programmed in the "SREP I2R Transaction Time-To-Live Register".

Checking for R2R Time-to-Live Expired Events is enabled when the "SREP I2R Transaction Time-To-Live Register".TVAL field is non-zero.

Each request and response packet in the R2R Queue has a time-to-live value associated with it. If the time-to-live value is 0 at the time a packet is selected for transmission to the physical layer, a R2R Time-to-Live Expired Event is detected.

Detection of a R2R Time-to-Live Expired Event causes the I_R2R_TTL bit in the "SREP ISF Logical Error Detect CSR" to be set to 1. When the R2R_TTL_EN bit is set to 1 in the "SREP ISF Logical Error Logging Enable CSR", assertion of I_R2R_TTL causes information to be latched in the Bridge ISF Logical Error Information registers.

For more information about the operation of the Bridge ISF Logical Error Information registers, see "SREP Bridge ISF Event Information Registers".

The packet for which a time-to-live expired event is detected is discarded. No error response is sent to the RapidIO requestor.

To create an R2R Time-to-Live Expired Event, perform the following steps:

1. Configure the MAC to stop accepting packets from the SREP.
2. Set the "SREP I2R Transaction Time-To-Live Register".TVAL field to a non-zero value.
3. Send a valid RapidIO NWRITE_R transaction to the SREP.
4. Wait for a period of time that will allow the time-to-live period to expire.
5. Configure the MAC to resume accepting packets from the SREP.

An R2R Time-to-Live Expired Event should be detected at this point.

10.12.15 Register Access Response Time-to-Live Expired

A Register Access Response Time-to-Live Expired Event is detected when a register access response packet is buffered in the Register Access Data Buffer/Response Queue for a period longer than that programmed in the "SREP I2R Transaction Time-To-Live Register".

Checking for Register Access Response Time-to-Live Expired Events is enabled when the "SREP I2R Transaction Time-To-Live Register".TVAL field is non-zero.

Each response in the Register Access Response Queue has a time-to-live value associated with it. If the time-to-live value is 0 at the time a response packet is selected for transmission to the physical layer, a Register Response Time-to-Live Expired Event is detected.

Detection of a Register Access Response Time-to-Live Expired Event causes the I_REG_TTL bit in the "SREP ISF Logical Error Detect CSR" to be set to 1. When the REG_TTL_EN bit is set to 1 in the "SREP ISF Logical Error Logging Enable CSR", assertion of I_REG_TTL causes the packet information to be latched in the Bridge ISF Logical Error Information registers, starting at "SREP ISF Logical Error Upper Attributes Capture CSR".

For more information about the operation of the Bridge ISF Logical Error Information registers, see "SREP Bridge ISF Event Information Registers".

The response packet for which a time-to-live expired event is detected is discarded. No error response is sent to the RapidIO requestor.

To create a Register Access Response Time-to-Live Expired Event, perform the following steps:

1. Configure the MAC to stop accepting packets from the SREP.
2. Set the "SREP I2R Transaction Time-To-Live Register".TVAL field to a non-zero value.
3. Send a valid RapidIO register access transaction that requires a response (NREAD, NWRITE_R, Maintenance Read, Maintenance Write) to the SREP.
4. Wait for a period of time that will allow the time-to-live period to expire.
5. Configure the MAC to resume accepting packets from the SREP.

A Register Access Response Time-to-Live Expired Event should be detected at this point.

10.12.16 I2R Transaction Time-to-Live Expired

An I2R Transaction Time-to-Live Expired Event is detected when a I2R transaction is buffered in the I2R Data Buffer/Header Queue for a period longer than that programmed in the "SREP I2R Transaction Time-To-Live Register".

Checking for I2R Transaction Time-to-Live Expired Events is enabled when the "SREP I2R Transaction Time-To-Live Register".TVAL field is non-zero.

Each transaction in the I2R Queue has a time-to-live value associated with it. If the time-to-live value is 0 at the time a request or response transaction is selected for transmission to the physical layer, a I2R Transaction Time-to-Live Expired Event is detected.

Detection of a I2R Transaction Time-to-Live Expired Event causes the I_I2R_TTL bit in the "SREP ISF Logical Error Detect CSR" to be set to 1. When the I2R_TTL_EN bit is set to 1 in the "SREP ISF Logical Error Logging Enable CSR", assertion of I_I2R_TTL causes the packet information to be latched in the Bridge ISF Logical Error Information registers, starting at "SREP ISF Logical Error Upper Attributes Capture CSR".

For more information about the operation of the Bridge ISF Logical Error Information registers, see "SREP Bridge ISF Event Information Registers".

The handling of the Bridge ISF transaction whose time-to-live period has expired depends upon the type of transaction that timed out. a Bridge ISF response transaction for which a time-to-live expired event is detected is discarded. As a result, the RapidIO requestor does not see a response for its request.

The handling of a Bridge ISF request transaction time-to-live expiry depends upon whether the request transaction has begun to be decomposed. If the request transaction has not begun decomposition, then the request is dropped. No Bridge ISF error response is sent to complete the Bridge ISF transaction.

If the Bridge ISF request has sent at least one packet, then decomposition is halted and the request is discarded. To clean up the decomposed response, a RapidIO response timeout is tracked for a non-existent request packet associated with the Bridge ISF request. Once the requests associated with the Bridge ISF request have finished, either through receiving responses or timing out, the response is discarded.

When a decomposed transaction fails, some of the RapidIO transactions may have completed successfully. No Bridge ISF error response is sent.

To create an I2R Time-to-Live Expired Event, perform the following steps:

1. Configure the MAC to stop accepting packets from the SREP.
2. Set the "SREP I2R Transaction Time-To-Live Register".TVAL field to a non-zero value.
3. Send a valid RapidIO NREAD transaction to the SREP.
4. Wait for a period of time that will allow the time-to-live period to expire.
5. Configure the MAC to resume accepting packets from the SREP.

An NREAD Time-to-Live Expired Event should be detected at this point.

10.12.17 Transfer Error Acknowledge Event (TEA)

A transfer error acknowledge (TEA) event is detected when a RapidIO-to-Bridge ISF request cannot be transferred. For more information about TEA events, see "Transaction Timeout".

TEA events cause the I_TEA bit in the "SREP ISF Logical Error Detect CSR" to be set to 1.

Information about the Bridge ISF request that could not be transmitted can be latched in the Bridge ISF Error Information Registers. To enable latching of information about the TEAed Bridge ISF request, set the I_TEA_EN bit in the "SREP ISF Logical Error Logging Enable CSR" (see "SREP Bridge ISF Event Information Registers").

Detection of a TEA results in the Bridge ISF transaction being dropped. The function for dropping the packet is to send the packet on Bridge ISF with an EOP. If the transaction was a request that required a response, a RapidIO error response may be sent. A RapidIO error response is sent if the ERESP_DIS bit is 0 in the "SREP R2I RapidIO Miscellaneous Control CSR". If the ERESP_DIS bit is 1, then a RapidIO error response is not sent.

A TEA event can be created by routing a packet to a Bridge ISF port that does not exist.

10.12.18 Physical Link Transmission Failure

The MAC may detect an unrecoverable error on the link. At this point, logical layer requests and responses are not able to make forward progress without software intervention.

No specific error handling for physical link errors is used in the Logical Layer. Depending on the control values in the physical layer, the physical layer may begin to discard packets. Packet discard results in I/O Logical response timeout events. If the physical layer is configured to retain packets, I/O Logical response timeout events may be seen if the timeout periods are shorter than the period required for software to restore the physical link to normal operation. For more information on physical layer error conditions and handling, see "Physical Layer Events".

11. SREP Event Management Support

This chapter describes the system of error and event notification in the SREP. Topics discussed include the following:

11.1 Overview

The SREP has the following event detection/management features:

- Compliant to the RapidIO specification for detection of Transport and Logical layer errors.
- Compliant to Part 8: Error Management Extensions of the *RapidIO Interconnect Specification (Revision 1.3)*
- Separate events in the "Block Event Status Register" for the following:
  - SREP Error related events
  - SREP Reception of a Multicast Event Control Symbol
  - SREP Reception of a valid Reset request
  - SREP Reception of a RapidIO port-write
  - SREP Reception of a RapidIO Doorbell
- Ability to receive 2 port-write messages
- Interface to trigger the transmission of a RapidIO port-write from outside the Tsi620
- Interface to control the implementation-specific values in a RapidIO Error Management port-write
- Ability to select which SREP events cause transmission of a port-write
- Ability to select which SREP events cause an error interrupt to be asserted
- With some exceptions, it is possible to configure the SREP to generate none, one, or both types of notification for each event

This section is organized as an event summary, which defines the short phrases that identify events and links these short phrases to specific status bits. This is followed by a discussion of events including:

- Error rate thresholds

- Illegal and unsupported packets
- Port-write packet transmission and reception
- Multicast-event control symbols
- Parity errors

After discussing these events, there is a general discussion of the standard registers that latch information about events. Lastly, there is description of the events, event status hierarchy, and a description of how to enable events for port-write and interrupt notification. The last section is organized based on registers and bit fields to facilitate design of interrupt handlers.

11.2 Event Summary

When describing events, it is useful to use a short phrase rather than refer to a register bit. Table 47, Table 48, and Table 49 identify the ‘short phrases’ associated all the events that can be raised within the SREP, and which status bit(s) these events are associated with. These short phrases are associated with register bits in subsequent tables.

Table 45: Event Summary Field Descriptions

Column Name	Description
Event Name	The ‘short phrase’ which identifies the event and its associated status registers.
Type	This column has two values: • Err - This event identifies an error condition that is detected • Stat - This event identifies an interesting condition that is not an error
Status Bit(s) and Description Cross Reference	This column contains cross references to the register bit(s) that have the status of the event, as well as cross references to text descriptions of the event. Text descriptions of error events describe how the event is handled by the SREP.
INT	This column has two values: • Yes - The event can be reported using an interrupt • No - The event cannot be reported using an interrupt
PW	This column has two values: • Yes - The event can be reported using a port-write • No - The event cannot be reported using a port-write

Table 45: Event Summary Field Descriptions (Continued)

Column Name	Description
Dir	This column has the following values. Each value identifies a request or response source that is relevant to understanding the implications of the Err Resp and Info column: • I2R Req - The handling of this event is described for Bridge ISF-to-RapidIO Request transactions. Note that I2R Req transactions may require R2I Responses... • I2R Resp - The handling of this event is described for Bridge ISF-to-RapidIO Response transactions. Note that these Response transactions must be a result of a R2I Req. • R2I Req - The handling of this event is described for RapidIO-to-Bridge ISF Request transactions. Note that R2I Req transactions may require I2R Responses... • R2I Resp - The handling of this event is described for RapidIO-to-Bridge ISF Response transactions. Note that these Response transactions must be the result of an I2R Req. • R2Reg Req - The handling of this event is described for RapidIO register requests • R2Reg Resp - The handling of this event is described for RapidIO register responses • N/A - The handling of this event does not directly affect the logical layer
Err Resp	This column has the following values: • Yes - The event results in a Logical Layer error response being sent to the originator of a transaction. The originator of the transaction depends upon the value of the 'Dir' field. R2Reg Req and R2I Req error responses are sent on RapidIO. I2R Req error responses are sent on the Bridge ISF. • May - An event may result in a Logical Layer error response being sent, depending upon a control value. These are usually timeout events. • No - The event does not result in a Logical Layer error response being sent.
Info	This column has the following values. Note that the information registers referenced here are described more fully in "Event Capture". • No - Information is not logged for the event • RLog - Additional information about the event may be logged in the RapidIO Logical Layer Error Information Registers (see "RapidIO Logical/Transport Error Information Registers"). • ILog - Additional information about the event may be logged in the Bridge ISF Logical Layer Error Information Registers (see "SREP Bridge ISF Event Information Registers"). • Oth - Additional information about the event may be logged in event-specific registers. Refer to the tables for each event that describe the additional information latched for the event.

The remainder of this chapter uses the “short phrase” form to describe events.

Table 46: SREP Physical Layer Events

Event Name	Type	Status Bits and Description Cross Reference	INT	PW	Dir	Err Resp	Info
Link Init	Stat	“SREP Interrupt Status Register”.LINK_INIT_NOTIFICATION (see “Link Initialization Events”)	Yes	Yes	N/A	No	No

Table 47: SREP Transport Layer Events

Event Name	Type	Status Bits and Description Cross Reference	INT	PW	Dir	Err Resp	Info
Illegal Target	Err	“SREP Logical and Transport Layer Error Detect CSR”.L_ILL_TARG (see “Illegal Target Event”)	Yes	Yes	R2I Req R2I Resp	No No	RLog RLog
R2I Packet CRC Error Detected	Stat	“SREP R2I Event Status Register”.L_PKT_CRC (see “R2I CRC Error Event”)	Yes	No	N/A	No	Phy
Packet Stomped	Stat	“SREP R2I Event Status Register”.L_PKT_STOMP (see “R2I Stomped Packet Event”)	Yes	No	N/A	No	No

Table 48: SREP Maintenance and Logical Layer Events

Event Name	Type	Status Bits and Description Cross Reference	INT	PW	Dir	Err Resp	Info
Doorbell Received	Stat	“SREP Interrupt Status Register”.DB_RX (see “RapidIO Doorbell Request Handling”)	Yes	No	R2I Req	N/A	Oth
Register Read R2Reg BAR Parity Error	Err	“SREP Logical and Transport Layer Error Detect CSR”.L_R2I_PERR (see “RapidIO-to-Bridge ISF Parity Error”)	Yes	Yes	R2Reg Req	No	RLog
Register Read R2Reg LUT Entry Parity Error	Err	“SREP Logical and Transport Layer Error Detect CSR”.L_R2I_PERR (see “RapidIO-to-Bridge ISF Parity Error”)	Yes	Yes	R2Reg Req	No	RLog
Port-Write Received	Stat	“SREP Port-Write Receive Status Register”.PWn_RXD (see “Maintenance Port-Write Support”)	Yes	No	N/A	N/A	Oth

Table 48: SREP Maintenance and Logical Layer Events (Continued)

Event Name	Type	Status Bits and Description Cross Reference	INT	PW	Dir	Err Resp	Info
R2I BAR Parity Error	Err	"SREP Logical and Transport Layer Error Detect CSR".L_R2I_PERR (see "RapidIO-to-Bridge ISF Parity Error")	Yes	Yes	R2I Req R2Reg Req	Yes Yes	RLog RLog
R2I LUT Entry Parity Error	Err	"SREP Logical and Transport Layer Error Detect CSR".L_R2I_PERR (see "RapidIO-to-Bridge ISF Parity Error")	Yes	Yes	R2I Req	Yes	RLog
Error Response	Err	"SREP Logical and Transport Layer Error Detect CSR".L_ERR_RESP (see "RapidIO Error Response Events")	Yes	Yes	R2I Resp I2R Req	No Yes	RLog No
Doorbell Error Response	Err	"SREP Logical and Transport Layer Error Detect CSR".L_DB_ERR_RESP (see "RapidIO Doorbell Error Response Events")	Yes	Yes	R2I Resp I2R Req	No No	RLog No
Illegal Transaction	Err	"SREP Logical and Transport Layer Error Detect CSR".L_ILL_TRANS (see "RapidIO Illegal Transaction Events")	Yes	Yes	R2I Req R2I Resp	Yes No	RLog RLog
Response Timeout	Err	"SREP Logical and Transport Layer Error Detect CSR".L_RESP_TO (see "Logical I/O Packet Response Timeout Events")	Yes	Yes	I2R Req	May	RLog
Unexpected Response	Err	"SREP Logical and Transport Layer Error Detect CSR".L_UNEXP_RESP (see "RapidIO Unexpected Response")	Yes	Yes	R2I Resp	No	RLog
Unsupported Transaction	Err	"SREP Logical and Transport Layer Error Detect CSR".L_UNSUP_TRANS (see "RapidIO Unsupported Transaction Events")	Yes	Yes	N/A	No	RLog
R2I Packet Time-to-Live Expired	Err	"SREP Logical and Transport Layer Error Detect CSR".L_R2I_TTL (see "RapidIO Logical Layer Time-to-Live Expired")	Yes	Yes	R2I Req I2R Resp	No No	RLog RLog
Spoof Response	Err	"SREP Logical and Transport Layer Error Detect CSR".L_SPOOF (see "RapidIO Spoof Response Event")	Yes	Yes	R2I Resp I2R Req	No Yes	RLog No
Out Of Bounds (OOB) RIO Request	Err	"SREP Logical and Transport Layer Error Detect CSR".L_OOB (see "RapidIO OOB Request Event")	Yes	Yes	R2I Req	Yes	RLog
RIO Write Denied	Err	"SREP Logical and Transport Layer Error Detect CSR".L_NO_WR (see "RapidIO Write Denied Event")	Yes	Yes	R2I Req	Yes	RLog

Table 48: SREP Maintenance and Logical Layer Events (Continued)

Event Name	Type	Status Bits and Description Cross Reference	INT	PW	Dir	Err Resp	Info
RIO Read Denied	Err	"SREP Logical and Transport Layer Error Detect CSR".L_NO_RD (see "RapidIO Read Denied Event")	Yes	Yes	R2I Req	Yes	RLog
Illegal Register Access	Err	"SREP R2I Event Status Register".L_BAD_REG_ACC (see "RapidIO Illegal Register Access")	Yes	Yes	R2Reg Req	Yes	RLog
LUT Entry Boundary Crossed	Stat	"SREP R2I Event Status Register".L_LUT_BND (see "RapidIO LUT Entry Boundary Crossing Event")	Yes	Yes	N/A	N/A	RLog

Table 49: SREP Bridge ISF Logical Layer Events

Event Name	Type	Status Bits and Description Cross Reference	INT	PW	Dir	Err Resp	Info
Register Read I2R BAR Parity Error	Err	"SREP ISF Logical Error Detect CSR".I_I2R_PERR (see "Bridge ISF-to-RapidIO Parity Error")	Yes	Yes	R2Reg Req	No	ILog
BAR Entry Parity Error	Err	"SREP ISF Logical Error Detect CSR".I_I2R_PERR (see "Bridge ISF-to-RapidIO Parity Error")	Yes	Yes	I2R Req R2Reg Req	Yes No	ILog ILog
Register Read I2R LUT Entry Parity Error	Err	"SREP ISF Logical Error Detect CSR".I_I2R_PERR (see "Bridge ISF-to-RapidIO Parity Error")	Yes	Yes	R2Reg Req	No	ILog
LUT Entry Parity Error	Err	"SREP ISF Logical Error Detect CSR".I_I2R_PERR (see "Bridge ISF-to-RapidIO Parity Error")	Yes	Yes	R2Reg Req I2R Req	No Yes	ILog ILog
ECC Error	Err	"SREP ISF Logical Error Detect CSR".I_ECC_ERR (see "ECC Error in Data Path")	Yes	Yes	N/A	No	Oth
Bridge ISF Error Response	Err	"SREP ISF Logical Error Detect CSR".I_ERR_RESP (see "Bridge ISF Error Response Events")	Yes	Yes	R2I Req	Yes	ILog
Bridge ISF Response Timeout	Err	"SREP ISF Logical Error Detect CSR".I_RESP_TO (see "Bridge ISF Response Timeout Events")	Yes	Yes	R2I Req	May	ILog
Bridge ISF Unexpected Response	Err	"SREP ISF Logical Error Detect CSR".I_UNEXP_RESP (see "Bridge ISF Unexpected Response")	Yes	Yes	N/A	No	ILog

Table 49: SREP Bridge ISF Logical Layer Events (Continued)

Event Name	Type	Status Bits and Description Cross Reference	INT	PW	Dir	Err Resp	Info
Bridge ISF Unsupported Transaction	Err	"SREP ISF Logical Error Detect CSR".I_UNSUP_TRANS (see "Bridge ISF Unsupported Transaction Events")	Yes	Yes	I2R Req	No	ILog
R2R Queue Time-to-Live Expired	Err	"SREP ISF Logical Error Detect CSR".I_R2R_TTL (see "R2R Time-to-Live Expired")	Yes	Yes	R2I Resp	No	ILog
Register Access Response Time-to-Live Expired	Err	"SREP ISF Logical Error Detect CSR".I_REG_TTL (see "Register Access Response Time-to-Live Expired")	Yes	Yes	R2I Resp	Yes	ILog
I2R Transaction Time-to-Live Expired	Err	"SREP ISF Logical Error Detect CSR".I_I2R_TTL (see "I2R Transaction Time-to-Live Expired")	Yes	Yes	I2R Req R2I Resp	Yes No	ILog ILog
TEA	Err	"SREP ISF Logical Error Detect CSR".I_TEA (see "Transfer Error Acknowledge Event (TEA)")	Yes	Yes	R2I Req R2I Resp	May No	ILog ILog
Bridge ISF Unsegmentable Request	Err	"SREP ISF Logical Error Detect CSR".I_UNSAR_REQ (see "Bridge ISF Unsegmentable Request")	Yes	Yes	I2R Req	Yes	ILog
OOB Bridge ISF Request	Err	"SREP ISF Logical Error Detect CSR".I_OOB (see "Bridge ISF OOB Request Event")	Yes	Yes	I2R Req	Yes	ILog
Bridge ISF Write Denied	Err	"SREP ISF Logical Error Detect CSR".I_NO_WR (see "Write Denied Event")	Yes	Yes	I2R Req	No	ILog
Bridge ISF Read Denied	Err	"SREP ISF Logical Error Detect CSR".I_NO_RD (see "Read Denied Event")	Yes	Yes	I2R Req	Yes	ILog
Bridge ISF LUT Entry Boundary Crossed	Stat	"SREP ISF Logical Error Detect CSR".I_LUT_BND (see "Bridge ISF LUT Entry Boundary Crossing Event")	Yes	Yes	I2R Req	N/A	ILog
Bridge ISF Byte Enables Discontiguous	Stat	"SREP ISF Logical Error Detect CSR".I_BE_DISCONT (see "Bridge ISF Byte Enables Discontiguous Event")	Yes	Yes	I2R Req	N/A	ILog

11.3 Port-Writes

For information on the transmission and reception of port-write packets, see "Maintenance Port-Write Support".

Notification of port-write reception is performed through the top level SREP_PW_RX event in the "Block Event Status Register".

11.3.1 Servicing Port-Writes from "SREP Port-Write Transmit Trigger Register"

If a system host receives a port-write that was transmitted due to the "SREP Port-Write Transmit Trigger Register", the software must ensure that the data in the port-write is sufficient to handle the port-write, and to retransmit the port-write if necessary.

Note that port-writes triggered from the "SREP Port-Write Transmit Trigger Register" are transmitted once. The retransmit timeout period controlled by the "SREP Port-Write Parameters Register" is not invoked.

11.4 Multicast-Event Control Symbol (MECS)

For information on the transmission and reception of Multicast-Event Control Symbols, see "Multicast-event Control Symbols".

Notification of MECS is performed through the top level SREP_MECS_RX event in the "Block Event Status Register".. It is enabled when the MECS_EN bit is set to 1 in the "SREP Interrupt Event Enable Register".

11.5 Reset Request Reception

For information on the transmission and reception of Reset Requests, see "Reset Control Symbol Processing".

Notification of Reset Request Reception is performed through the top level SREP_RESET_RX event in the "Block Event Status Register".. It is enabled when the RCS_EN bit is set to 1 in the "SREP Interrupt Event Enable Register".

11.6 Event Capture

There are two groups of registers that capture information for SREP errors: RapidIO Logical/Transport Error Information registers, and Bridge ISF Logical Error Information registers.

11.6.1 RapidIO Logical/Transport Error Information Registers

The occurrence of RapidIO packet related errors is captured in the Logical/Transport Layer Error Status registers:

- "SREP Logical and Transport Layer Error Detect CSR"
- "SREP R2I Event Status Register"

Information about RapidIO transactions that cause logical and transport layer errors is captured in the Logical/Transport Layer Error Information registers, consisting of the following:

- "SREP Logical and Transport Layer Address Capture CSR"
- "SREP Logical and Transport Layer Device ID Capture CSR"
- "SREP Logical and Transport Layer Control Capture CSR"
- "SREP R2I Error ISF Command Attributes Capture CSR"
- "SREP R2I Error ISF Logical Error Decomposition Attributes Capture CSR"
- "SREP R2I BAR and LUT Parity Error Status Register"

Multiple events may be detected in a packet or in different packets at the same time (see "Precedence of Logical I/O Packet Errors"). The contents of the Logical/Transport Layer Error Information Events may or may not be valid, depending on which event is captured. Therefore, to correctly interpret the contents of the Logical/Transport Layer Error Information registers, it must be possible to determine which event was last captured.

Logical/Transport Layer events can be enabled through the "SREP Logical and Transport Layer Error Logging Enable CSR" and the "SREP R2I Event Status Logging Enable Register", respectively. Only enabled events may cause information to be latched in the Logical/Transport Layer Error Status Information registers.

Notification through an interrupt or port-write only occurs if an event is enabled. Disabled events may set bits in the status registers, but they do not cause an interrupt or port-write, nor do they cause information to be latched for the errors.

Event reporting is delayed until a packet is completely received and checked, in order to allow detection of CRC Error and Packet Stomp events. All disabled events that are detected in a packet cause the associated status bits in the Logical/Transport Layer Error Status registers to be set.

If a packet has multiple errors, then at least one error is detected and reported. Detection of all error conditions is not guaranteed.

If a packet has a CRC Error and/or Packet Stomp error, all other error conditions in the packet are not be reported.

The highest priority enabled event(s) causes the following (see also "Precedence of Logical I/O Packet Errors"):

- The associated status bit to be set in the Logical/Transport Layer Error Status registers.
- The Logical/Transport Layer Error Status registers are locked.
- The Logical/Transport Layer Error Information registers are locked.

Lower priority enabled event(s) detected in the same packet do not cause status bits to be set in the Logical/Transport Layer Error Status registers. When registers are locked, their state does not change for subsequent errors. The Logical/Transport Layer Error Status and Information registers are unlocked when there are no longer any enabled events with their status bit set.

For example, suppose that packet X attempts a read that spans a LUT boundary in an area of memory for which read access is denied. At the same time as packet X completes reception, Packet Y reaches a response timeout. The LUT boundary event is disabled, and the Read Denied and Response Timeout events are enabled. The "SREP R2I Event Status Register". L_LUT_BND bit is set, because this event is disabled. Since a response-timeout event is higher priority than a Read Denied event, the "SREP R2I Event Status Register".L_RESP_TO bit is set and information about packet Y is captured in the Logical/Transport Layer Error Information registers. The "SREP Logical and Transport Layer Error Detect CSR".L_NO_RD bit remains clear.

Table 51 captures what status information is captured for which events. An 'X' indicates that data is latched, a '-' means that the fields are not relevant for the error. An explanation of the column headings of Table 51 is captured in Table 50.

For a specific error, not all fields within a register may be relevant, and therefore may not capture data for the error. In this case, those fields that are not relevant to the error are updated to be 0.

For a specific error, not all registers may be relevant. Registers that are not relevant for an error are not changed.

Within Table 51, the information captured for an event sometimes depends on whether the packet received is a supported request, a supported response, or an unsupported packet type. Information for supported requests are identified in a row starting with 'Req'. Information for supported responses is identified in a row starting with 'Resp'.

Information for unknown packet types is identified in a row starting with 'Unkwn'. When the errors L_UNSUP_TRANS, L_ILL_TRANS, L_PKT_CRC and L_PKT_STOMP occur it is possible that the values that determine where information within the packet lies are corrupted. There are three values that determine the format of the packet: FTYPE, TTYPE, and TT code. If an FTYPE is unsupported or illegal, it should be interpreted as being an NREAD (FTYPE 2) packet. When the TTYPE is unsupported or illegal, it should be interpreted as being a request packet. When the TT code is reserved, it should be interpreted as being '0b01', indicating a 16-bit destination ID is used.

If the packet ends before the location of a field, the error information registers should have a value of 0 for that field. These choices are made to maximize the amount of information latched for the faulty packet.

Table 50: Registers/Bits Associated with Table 51 Columns

Table 51 Heading	Register and Bit
ADDR	"SREP Logical and Transport Layer Address Capture CSR"
SRC/DST	"SREP Logical and Transport Layer Device ID Capture CSR" and "SREP Logical and Transport Layer Control Capture CSR".TT
PKT TYPE	"SREP Logical and Transport Layer Control Capture CSR".FTYPE "SREP Logical and Transport Layer Control Capture CSR".TTYPE

Table 50: Registers/Bits Associated with Table 51 Columns (Continued)

Table 51 Heading	Register and Bit
PKT PRIO	"SREP Logical and Transport Layer Control Capture CSR".PRIO "SREP Logical and Transport Layer Control Capture CSR".CRF
PKT STAT	"SREP Logical and Transport Layer Control Capture CSR".STAT_SIZE "SREP Logical and Transport Layer Control Capture CSR".WDPTR "SREP Logical and Transport Layer Control Capture CSR".RESP_SIZE Note: Described as • SZ (request packet size and WDPTR) • RESP_SIZE (Request packet size, WDPTR, and RESP_SIZE) • STAT (response packet status, bit from WDPTR location in response packet)
Bridge ISF INFO	"SREP R2I Error ISF Command Attributes Capture CSR" and "SREP R2I Error ISF Logical Error Decomposition Attributes Capture CSR"
PERR INFO	"SREP R2I BAR and LUT Parity Error Status Register"

Table 51: Information Captured for RapidIO Logical/Transport Layer Errors

Event	ADDR	SRC/ DST	PKT TYPE	PKT PRIO	PKT STAT	Bridge ISF INFO	PERR INFO
R2I Packet CRC Error Detected[a]	X	X	X	X	SZ / STAT	-	-
Packet Stomped[a]	X	X	X	X	SZ / STAT	-	-
Port-Write Received	-	-	-	-	-	-	-
Illegal Target Note: The fields captured represent offsets within the packet. Software must interpret the fields captured based on the PKT TYPE values.	X[b]	X	X	X	SZ / STAT	-	-
Register Read R2Reg BAR Parity Error	-	-	-	-	-	-	X
Register Read R2Reg LUT Entry Parity Error	-	-	-	-	-	-	X
R2I BAR Parity Error	X[c]	X	X	X	-	-	X
R2I LUT Entry Parity Error	X[c]	X	X	X	-	-	X
Error Response (All information latched is for the RapidIO Request)	X	X	X	X	-	-	-

Table 51: Information Captured for RapidIO Logical/Transport Layer Errors (Continued)

Event		ADDR	SRC/ DST	PKT TYPE	PKT PRIO	PKT STAT	Bridge ISF INFO	PERR INFO
Doorbell Error Response (All information latched is for the RapidIO Request)		-	X	X	-	-	-	-
Illegal Transaction	Req	X	X	X	X	SZ	-	-
	Resp	-	X	X	X	STAT	-	-
	Unkwn	X	X	X	X	SZ		
Response Timeout Note: Information is latched for the Bridge ISF request and RapidIO Request packet.		X	X	X	X	SZ	X	-
Unexpected Response		-	X	X	X	STAT	-	-
Unsupported Transaction		X[d]	X	X	X	-	-	-
R2I Packet Time-to-Live Expired	Req	X	X	X	X	-	-	-
	Resp	-	X	X	X	-	-	-
	I2I	-	X[e]	-	-	-	-	-
Spoof Response		EXP[f]	X	X	X	RESP_ SZ	-	-
Out Of Bounds (OOB) RIO Request		X	X	X	X	SZ	-	-
RIO Write Denied		X	X	X	X	SZ	-	-
RIO Read Denied		X	X	X	X	SZ	-	-
Illegal Register Access		X	X	X	X	-	-	-
LUT Entry Boundary Crossed		X	X	X	X	SZ	-	-

a. Information is latched based on the location of fields within the packet. The contents of the registers may or may not make sense, as the packet may be corrupted or incomplete. If the packet length is too short for capturing a specific field, the field value is 0. For example, SZ_STAT information for SWRITE packets, which do not have a size or status field, is latched from the data position corresponding to where the SZ_STAT field would be for an NWRITE packet.

b. The address field is latched based on the location of the address field within the packet, and so is only valid for request packets with address fields.

c. For more information, see "R2I LUT and BAR Parity Error Information Latching".

d. The contents of the address registers is the bytes where the address would be in an I/O Logical Layer packet. Depending on the type of unsupported transaction, the fields captured in the packet will vary.

e. The contents of the source/destination ID fields indicate the Bridge ISF destination port (destination) of a response packet.

f. The contents of the "SREP Logical and Transport Layer Address Capture CSR" are the expected TT code and SourceID for the response. For information, see the description of this register.

After the SREP is reset, all Logical/Transport Layer Event Enable registers are unchanged. If information for an event was latched before the reset, the Logical/Transport Layer Error Status registers remain locked after the reset.

Logical/Transport Layer events that occur after the reset can be included in the Logical/Transport Layer Error Status registers, if the Logical/Transport Layer Error Status registers are not locked.

The contents of the Logical/Transport Layer Error Status and Information registers are kept over all but power-up resets to enable debugging of catastrophic system events.

11.6.2 SREP Bridge ISF Event Information Registers

The occurrence of SREP Bridge ISF Events is latched in the "SREP ISF Logical Error Detect CSR". Information about SREP Bridge ISF Events is latched in the following Bridge ISF Event Information registers:

- "SREP ISF Logical Error Upper Attributes Capture CSR".
- "SREP ISF Logical Error Middle Attributes Capture CSR"
- "SREP ISF Logical Error Lower Attributes Capture CSR"
- "SREP ISF Logical Error RapidIO Routing Attributes Capture CSR"
- "SREP ISF Logical Error RapidIO Physical Attributes Capture CSR"
- "SREP ISF Logical Error RapidIO Lower Address Capture CSR"
- "SREP I2R BAR and LUT Parity Error Status Register"

Multiple events may be detected in an Bridge ISF transaction or in different Bridge ISF transactions at the same time (see "Precedence of Bridge ISF Errors"). The contents of the Bridge ISF Event Information registers may or may not be valid, depending on which event is captured. Therefore, to correctly interpret the contents of the Bridge ISF Event Information registers, it must be possible to determine which event was last captured.

Bridge ISF Events can be enabled through the "SREP ISF Logical Error Logging Enable CSR". Only enabled events can cause information to be latched in the Bridge ISF Event Information registers.

Notification through an interrupt or port-write only occurs if an event is enabled. Disabled events may set bits in the status registers, but they od not cause an interrupt or port-write.

Event reporting is delayed until a transaction is completely received and checked. All disabled events that are detected in a transaction cause the associated status bits in the "SREP ISF Logical Error Detect CSR" to be set.

If a transaction has multiple errors, then at least one error is detected and reported. Detection of all error conditions is not guaranteed.

The highest priority enabled event(s) causes the following (see also "Precedence of Bridge ISF Errors"):

- The associated status bit to be set in the "SREP ISF Logical Error Detect CSR".
- The "SREP ISF Logical Error Detect CSR" is locked.
- The Bridge ISF Event Information registers are locked.

Lower priority enabled event(s) detected in the same packet do not cause status bits to be set in the "SREP ISF Logical Error Detect CSR". When registers are locked, their state does not change for subsequent errors. The "SREP ISF Logical Error Detect CSR" and Bridge ISF Event Information registers are unlocked when there are no longer any enabled events with their status bit is set.

For example, suppose that Transaction X attempts a read that spans a LUT boundary in an area of memory for which read access is denied. At the same time as Transaction X completes reception, Transaction Y reaches a response timeout. The LUT boundary event is disabled, and the Read Denied and Bridge ISF Response Timeout events are enabled. The "SREP ISF Logical Error Detect CSR".I_LUT_BND bit is set, because this event is disabled. Since a response timeout event is higher priority than a Read Denied event, the "SREP ISF Logical Error Detect CSR".I_RESP_TO bit is set and information about packet Y is captured in the Bridge ISF Event Information registers. The "SREP ISF Logical Error Detect CSR".I_NO_RD bit remains clear.

Table 53 captures what status information is captured for which Bridge ISF events. An 'X' indicates that data is latched, a '-' means that the fields are not relevant for this error. An explanation of the column headings of Table 53 is captured in Table 52.

For a specific error, not all fields within a register may be relevant, and therefore may not capture data for the error. In this case, those fields that are not relevant to the error are updated to be 0.

For a specific error, not all registers may be relevant. Registers that are not relevant for an error are not changed.

Table 52: Registers/Bits Associated with Table 53 Columns

Table 53 Heading	Register and Bit
Bridge ISF INFO	"SREP ISF Logical Error Upper Attributes Capture CSR" and "SREP ISF Logical Error Middle Attributes Capture CSR" and "SREP ISF Logical Error Lower Attributes Capture CSR"
RIO ADDR	"SREP ISF Logical Error RapidIO Lower Address Capture CSR" Note: If the RIO TYPE indicates that this is a Doorbell packet, then the most significant 16 bits of the "SREP ISF Logical Error RapidIO Lower Address Capture CSR" contain the Doorbell payload. Note: Information is for the RapidIO Request

Table 52: Registers/Bits Associated with Table 53 Columns (Continued)

Table 53 Heading	Register and Bit
RIO SRC/DST	"SREP ISF Logical Error RapidIO Routing Attributes Capture CSR" and "SREP ISF Logical Error RapidIO Physical Attributes Capture CSR".TT Note: Information is for the RapidIO Request
RIO TYPE	"SREP ISF Logical Error RapidIO Physical Attributes Capture CSR".FTYPE "SREP ISF Logical Error RapidIO Physical Attributes Capture CSR".TTYPE Note: Information is for the RapidIO Request
RIO PRIO	"SREP ISF Logical Error RapidIO Physical Attributes Capture CSR".PRIO "SREP ISF Logical Error RapidIO Physical Attributes Capture CSR".CRF Note: Information is for the RapidIO Request
PERR INFO	"SREP I2R BAR and LUT Parity Error Status Register"
ECC INFO	"SREP ISF ECC Error Status Register" Note that the validity of the different fields in this register changes depending on where an ECC error is detected (see "ECC Error in Data Path"). If correctable and uncorrectable ECC errors occur in consecutive datums, an indeterminate mixture of information from both errors will be latched. Information on ECC errors is latched in the "SREP ISF ECC Error Status Register", but only locked when the ECC_ERR_EN bit is set in the "SREP ISF Logical Error Logging Enable CSR".

Table 53: Information Captured for Bridge ISF Logical/Transport Layer Errors

Event	Bridge ISF INFO	RIO ADDR	RIO SRC/DST	RIO TYPE	RIO PRIO	PERR INFO	ECC INFO
Register Read I2R BAR Parity Error	-	-	-	-	-	X	-
BAR Entry Parity Error	X[a]	-	-	-	-	X	-
Register Read I2R LUT Entry Parity Error	-	-	-	-	-	X	-
LUT Entry Parity Error	X[a]	-	-	-	-	X	-
ECC Error for Transaction Received from Bridge ISF	X	-	-	-	-	-	X
ECC Error for Data read from R2I or I2R buffers	-	-	-	-	-	-	X
Bridge ISF Error Response (All information is latched for the Bridge ISF Request)	X[b]	X	X	-	X	-	-
Bridge ISF Response Timeout	-	X	X	-	X	-	-

Table 53: Information Captured for Bridge ISF Logical/Transport Layer Errors (Continued)

Event	Bridge ISF INFO	RIO ADDR	RIO SRC/DST	RIO TYPE	RIO PRIO	PERR INFO	ECC INFO
Bridge ISF Unexpected Response	X	-	-	-	-	-	-
Bridge ISF Unsupported Transaction	X	-	-	-	-	-	-
R2R Time-to-Live Expired	-	-	X	X	X	-	-
Register Access Response Time-to-Live Expired	-	-	X	X	X	-	-
I2R Transaction Time-to-Live Expired	X[c]	X	X	X	X	-	-
TEA	X	X[d]	X[d]	-	X[d]	-	-
Bridge ISF Unsegmentable Request	X	-	-	-	-	-	-
OOB Bridge ISF Request	X	-	-	-	-	-	-
Bridge ISF Write Denied	X	-	-	-	-	-	-
Bridge ISF Read Denied	X	-	-	-	-	-	-
ISF LUT Entry Boundary Crossed	X	-	-	-	-	-	-
ISF Byte Enables Discontiguous	X	-	-	-	-	-	-

a. See "I2R LUT and BAR Parity Error Information Latching".

b. Information is captured for the Bridge ISF response

c. Only the Bridge ISF Command and Bridge ISF Byte Count/byte lanes are valid for Bridge ISF Requests. For Bridge ISF responses, no Bridge ISF INFO is latched.

d. RapidIO information is captured if the transaction is the result of the reception of a RapidIO request. For a normal or error response to an Bridge ISF transaction, Response transaction destination (DESTID) are captured in the "SREP ISF Logical Error RapidIO Routing Attributes Capture CSR".

After the SREP is reset, all Bridge ISF events are disabled. This unlocks the "SREP ISF Logical Error Detect CSR". Bridge ISF events that occur after the reset can therefore be included in the "SREP ISF Logical Error Detect CSR".

The contents of the "SREP ISF Logical Error Detect CSR" and Bridge ISF Event Information registers are kept over all but power-up resets to enable debugging of catastrophic system events.

11.6.3 Debug Support

The RapidIO Logical Layer Error registers, and the Bridge ISF Logical Error Information Registers, can all be written to allow software to create an event and then assert an interrupt for test purposes. The RapidIO Logical Layer Error registers are compliant to the *RapidIO Interconnect Specification (Revision 1.3)*. The Bridge ISF Logical Error Information Registers use the "SREP ISF Logical Error Generate CSR" to generate Bridge ISF events.

Many RapidIO and Bridge ISF Logical errors can be created without resorting to writing to the registers. This allows an end-to-end test of software's ability to handle a real error. Creation of errors is described with each event.

11.7 Event Notification and Register Hierarchy

The SREP event structure is hierarchical, which allows software to determine the cause of an interrupt with minimum register access.

For the locations of additional information on how the events in the "SREP Interrupt Status Register" can be enabled, reported, and cleared, see Table 54.

Table 54: SREP Event Status Register Fanout

Interrupt Bit	Further Information
MECS	"Multicast-event Control Symbols"
RCS	"Reset Control Symbol Processing"
PW_RX	"Port-Writes"
DB_RX	"RapidIO Doorbell Request Handling"
RIO_LOG	"Logical/Transport Layer Error Detect Register Event Information"
ISF_LOG	"Bridge ISF Logical Error Detect Register Event Information"
IMPL_PHY_ERR	"Implementation-Specific Physical Layer Events and Event Handling"

"Event Management Register Hierarchy for Interrupts" describes how the groups of events, which are summarized in the status bits of the "SREP Interrupt Status Register", can be mapped to the SREP Error Event. Similarly, "Event Management Register Hierarchy for Port Writes" describes how the groups of events, which are summarized in the "SREP Port-Write Status Register", can trigger the transmission of port-writes. These diagrams capture the relationship between various status and interrupt/port-write enable registers. Additional diagrams showing the relationship between various control registers and the status registers are located with each group of events.

When an event is raised through an interrupt, the "SREP Interrupt Status Register" must be read first to determine why an interrupt was raised. Status bits in the "SREP Interrupt Status Register" allow the event service routine to decide which kind of event raised the interrupt. Port-Write Reception, Reset Control Symbol reception, Multicast Event Control Symbol reception and Doorbell reception have separate indicator bits in the "SREP Interrupt Status Register" to allow for faster handling.

After the software has read the "SREP Interrupt Status Register" and determined which type of event caused an event to be communicated, the event status registers associated with the event type, if any, must be accessed to determine the cause. "SREP Interrupt Status Register". IMPL_PHY_ERR, STD_PHY_ERR, RIO_LOG and ISF_LOG have additional registers associated with their error status.

The algorithm for handling an event communicated by a port-write is identical to that for handling an event communicated by an interrupt, except that the "SREP Port-Write Status Register" is read instead of the "SREP Interrupt Status Register".

The PORT_W_PEND bit in the "SREP Error and Status CSR" is set when a port-write must be sent for hardware reasons. This bit must be cleared after all causes of the port-write have been cleared to stop the SREP from generating port-writes.

Figure 26: Event Management Register Hierarchy for Interrupts

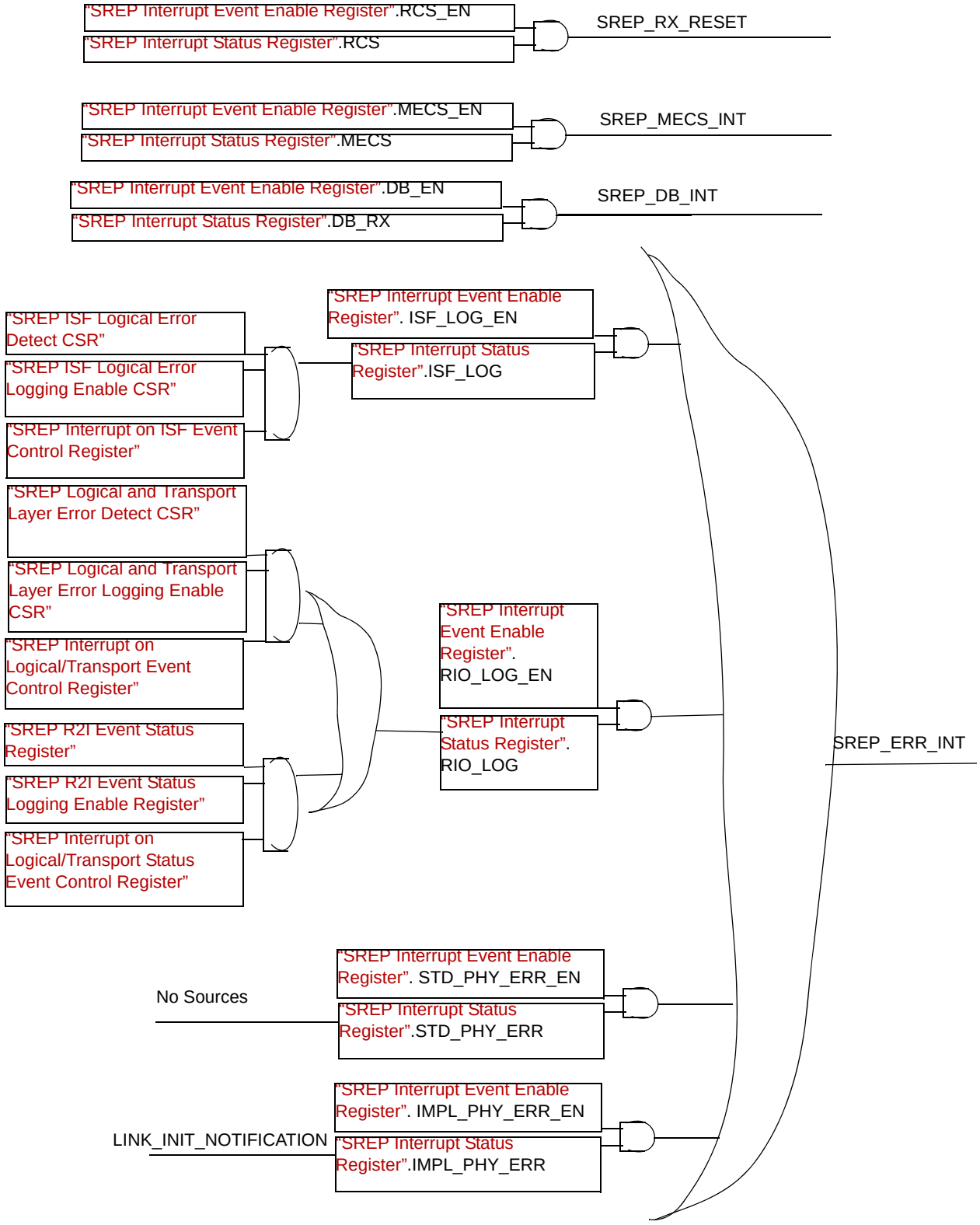
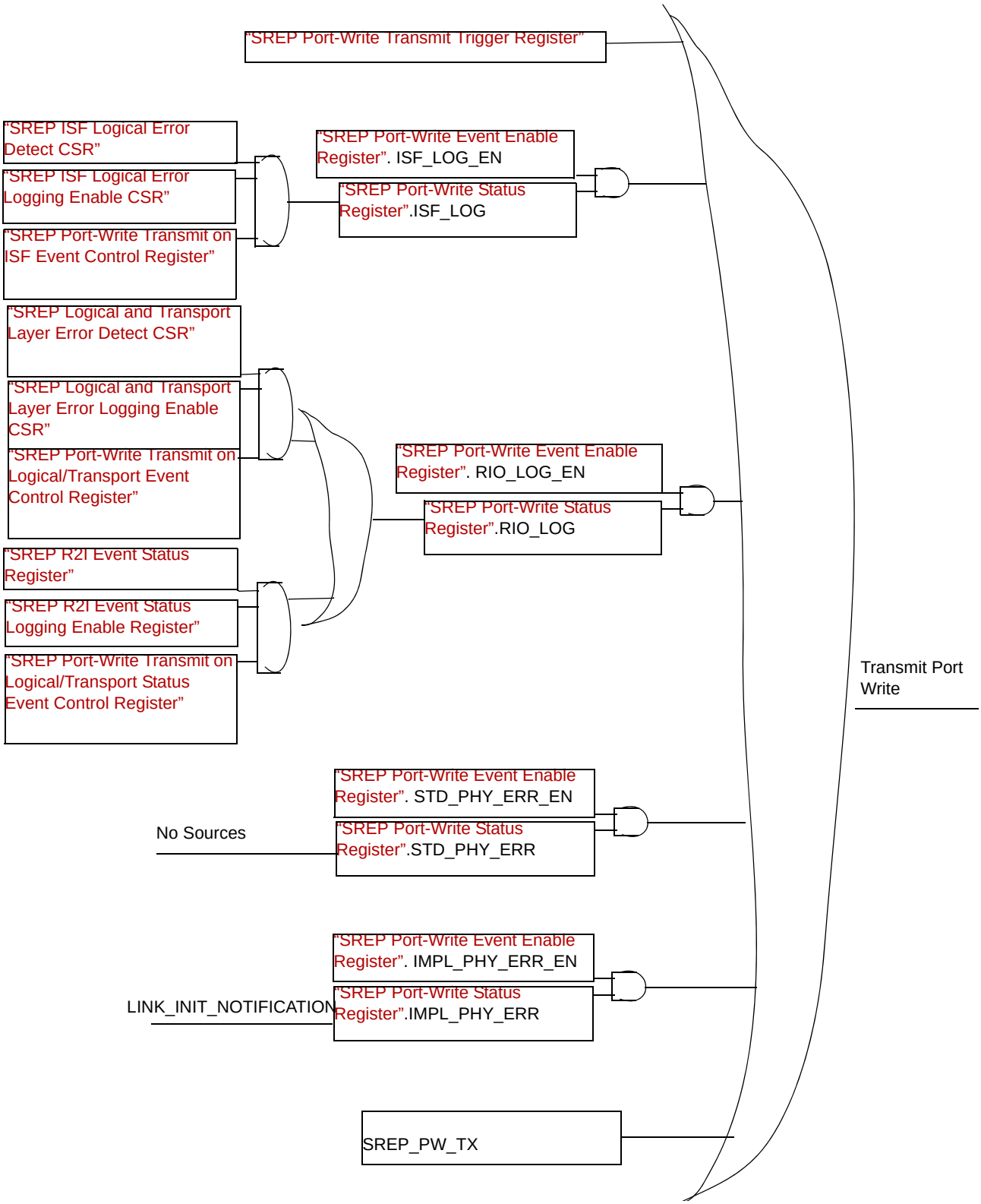


Figure 27: Event Management Register Hierarchy for Port Writes



11.7.1 Physical Layer Error Detect Register Event Information

There is only one Physical Layer event, the LINK_INIT_NOTIFICATION event.

Table 55: Physical Layer Event and Notification Control

Event Bit	Event Enable	Notification and Logging Enable	Interrupt Enable	Port-Write Enable
LINK_INIT_NOTIFICATION	Always detected when "SREP Error and Status CSR".PORT_OK is 1 and "SREP Port Control CSR".PORT_LOCKOUT is set to 1	"SREP Control Independent Register" • Set LINK_INIT_NOTIFICATION_EN bit to 1 No information is latched for this event.	"SREP Interrupt Event Enable Register" • Set IMPL_PHY_ERR_EN bit to 1 Note: The reset default value of IMP_PHY_ERR_EN is 0.	"SREP Port-Write Event Enable Register" • Set IMPL_PHY_ERR_EN bit to 1

Table 56: Physical Layer Event Handling

Event Bit	Event Clearing
LINK_INIT_NOTIFICATION	"SREP Port Control CSR" • Set PORT_LOCKOUT to 0 and then "SREP Interrupt Status Register" • Clear the LINK_INIT_NOTIFICATION bit by writing 1

Table 57: Logical/Transport Layer Status Event and Notification Control

Event Bit	Event Enable	Notification and Logging Enable	Interrupt Enable	Port-Write Enable
L_BAD_REG_ACC	Checking for incorrect size is always enabled. To enable checking for the source of register transactions, program the following registers: • "SREP Register Access Source ID Checking Control Register" • "SREP Register Access Small Source ID Checking Control Register" • "SREP Large Register Access Source ID Checking Register"	"SREP R2I Event Status Logging Enable Register" Set BAD_REG_ACC_EN bit to 1	"SREP Interrupt on Logical/Transport Status Event Control Register" • Set INT_BAD_REG_ACC_EN bit to 1	"SREP Port-Write Transmit on Logical/Transport Status Event Control Register" • Set PW_BAD_REG_ACC_EN bit to 1
L_LUT_BND	Always detected.	"SREP R2I Event Status Logging Enable Register" • Set LUT_BND_EN bit to 1 For more information, see "RapidIO LUT Entry Boundary Crossing Event"	"SREP Interrupt on Logical/Transport Status Event Control Register" • Set INT_LUT_BND_EN bit to 1 Note: The reset default value of INT_LUT_BND_EN is 1.	"SREP Port-Write Transmit on Logical/Transport Status Event Control Register" Set PW_LUT_BND_EN bit to 1

Table 57: Logical/Transport Layer Status Event and Notification Control (Continued)

Event Bit	Event Enable	Notification and Logging Enable	Interrupt Enable	Port-Write Enable
L_PKT_CRC	For information on the control of CRC error checking, see the MAC documentation.	No error information is latched by the Logical Layer for Packet CRC Events.	"SREP Interrupt on Logical/Transport Status Event Control Register" • Set INT_PKT_CRC_EN bit to 1 Note: The reset default value of INT_PKT_CRC_EN is 1.	It is not possible to send a Port-Write for a packet CRC error.
L_PKT_STOMP	Always detected.	No error information is latched by the Logical Layer for Packet Stomp Events.	"SREP Interrupt on Logical/Transport Status Event Control Register" • Set INT_PKT_STOMP_EN bit to 1 Note: The reset default value of INT_PKT_STOMP_EN is 1.	It is not possible to send a Port-Write for a STOMPed packet error.

11.7.2 Logical/Transport Layer Error Detect Register Event Information

Logical and Transport Layer events are in the "SREP Logical and Transport Layer Error Detect CSR" and the "SREP R2I Event Status Register". The following tables describe how to enable detection of events, how to control information logging for events, how to control event notification, and how to clear events in the "SREP Logical and Transport Layer Error Detect CSR" and the "SREP R2I Event Status Register".

For any event detected in the "SREP Logical and Transport Layer Error Detect CSR" or "SREP R2I Event Status Register" to cause an interrupt, the RIO_LOG_EN bit in the "SREP Interrupt Event Enable Register" must be set to 1.

The "SREP Logical and Transport Layer Error Clear CSR" is a 'Write 1 to Clear' version of the RapidIO standard register "SREP Logical and Transport Layer Error Detect CSR". Anything that can be completed with the "SREP Logical and Transport Layer Error Detect CSR" can also be done with the "SREP Logical and Transport Layer Error Clear CSR", without the chance of losing events when clearing the "SREP Logical and Transport Layer Error Detect CSR".

For any event detected in the "SREP Logical and Transport Layer Error Detect CSR" or "SREP R2I Event Status Register" to cause a port-write, the RIO_LOG_EN bit in the "SREP Port-Write Event Enable Register" must be set to 1.

It is only necessary to check the "SREP R2I Event Status Register" if the "SREP Logical and Transport Layer Error Detect CSR".STAT_EVENT bit is set to 1.

If the L_ISF_ERR bit is set in the "SREP R2I Event Status Register", then at least one Bridge ISF event is detected (see "Bridge ISF Logical Error Detect Register Event Information").

The 'Event Clearing' column of Tables 59 and 61 describes how to clear the event. An alternative to clearing the event is to disable further event notification for the event (see Tables 58 and 60).

The relationship among the event enable control registers for each Logical/Transport Layer event is shown in Figure 28.

Table 58: Logical/Transport Layer Event and Notification Control

Event Bit	Event Enable	Notification and Logging Enable	Interrupt Enable	Port-Write Enable
L_ERR_RESP	Always detected.	"SREP Logical and Transport Layer Error Logging Enable CSR" • Set ERR_RESP_EN bit to 1	"SREP Interrupt on Logical/Transport Event Control Register" • Set INT_ERR_RESP_EN bit to 1 Note: The reset default value of INT_ERR_RESP_EN is 1.	"SREP Port-Write Transmit on Logical/Transport Event Control Register" • Set PW_ERR_RESP_EN bit to 1
L_DB_ERR_RESP	Always detected.	"SREP Logical and Transport Layer Error Logging Enable CSR" • Set DB_ERR_RESP_EN bit to 1	"SREP Interrupt on Logical/Transport Event Control Register" • Set INT_DB_ERR_RESP_EN bit to 1 Note: The reset default value of INT_ERR_RESP_EN is 1.	"SREP Port-Write Transmit on Logical/Transport Event Control Register" • Set PW_DB_ERR_RESP_EN bit to 1
L_ILL_TRANS	Always detected.	"SREP Logical and Transport Layer Error Logging Enable CSR" • Set ILL_TRANS_EN bit to 1 (see "RapidIO Illegal Transaction Events").	"SREP Interrupt on Logical/Transport Event Control Register" • Set INT_ILL_TRANS_EN bit to 1 Note: The reset default value of INT_ILL_TRANS_EN is 1	"SREP Port-Write Transmit on Logical/Transport Event Control Register" • Set PW_ILL_TRANS_EN bit to 1

Table 58: Logical/Transport Layer Event and Notification Control (Continued)

Event Bit	Event Enable	Notification and Logging Enable	Interrupt Enable	Port-Write Enable
L_ILL_TARG	• "SREP Base Device ID CSR" • "SREP Destination ID Checking Control Register" • "SREP Large Secondary Destination ID Checking Control Register" For more information, see "8/16-bit Destination ID Support".	"SREP Logical and Transport Layer Error Logging Enable CSR" • Set ILL_TARG_EN bit to 1.	"SREP Interrupt on Logical/Transport Event Control Register" • Set INT_ILL_TARG_EN bit to 1 Note: The reset default value of INT_ILL_TARG_EN is 1.	"SREP Port-Write Transmit on Logical/Transport Event Control Register" • Set PW_ILL_TARG_EN bit to 1
L_RESP_TO	"SREP Response Timeout Control CSR" • Set TVAL to a non-zero value	"SREP Logical and Transport Layer Error Logging Enable CSR" • Set RESP_TO_EN bit to 1	"SREP Interrupt on Logical/Transport Event Control Register" • Set INT_RESP_TO_EN bit to 1 Note: The reset default value of INT_RESP_TO_EN is 1.	"SREP Port-Write Transmit on Logical/Transport Event Control Register" • Set PW_RESP_TO_EN bit to 1
L_UNEXP_RESP	Always detected.	"SREP Logical and Transport Layer Error Logging Enable CSR" • Set UNEXP_RESP_EN bit to 1	"SREP Interrupt on Logical/Transport Event Control Register" • Set INT_UNEXP_RESP_EN bit to 1 Note: The reset default value of INT_UNEXP_RESP_EN is 1.	"SREP Port-Write Transmit on Logical/Transport Event Control Register" • Set PW_UNEXP_RESP_EN bit to 1

Table 58: Logical/Transport Layer Event and Notification Control (Continued)

Event Bit	Event Enable	Notification and Logging Enable	Interrupt Enable	Port-Write Enable
L_UNSUP_TRANS	Always detected.	"SREP Logical and Transport Layer Error Logging Enable CSR" • Set UNSUP_TRANS_EN bit to 1 For more information, see "RapidIO Unsupported Transaction Events"	"SREP Interrupt on Logical/Transport Event Control Register" • Set INT_UNSUP_TRANS_EN bit to 1 Note: The reset default value of INT_UNSUP_TRANS_EN is 1.	"SREP Port-Write Transmit on Logical/Transport Event Control Register" • Set PW_UNSUP_TRANS_EN bit to 1
L_R2I_TTL	"SREP R2I Transaction Time-To-Live Register" • Set TVAL to a non-zero value	"SREP Logical and Transport Layer Error Logging Enable CSR" • Set R2I_TTL_EN bit to 1	"SREP Interrupt on Logical/Transport Event Control Register" • Set INT_R2I_TTL_EN bit to 1 Note: The reset default value of INT_R2I_TTL_EN is 1.	"SREP Port-Write Transmit on Logical/Transport Event Control Register" • Set PW_R2I_TTL_EN bit to 1
L_SPOOF	Always detected.	"SREP Logical and Transport Layer Error Logging Enable CSR" • Set SPOOF_EN bit to 1 For more information, see "RapidIO Unsupported Transaction Events"	"SREP Interrupt on Logical/Transport Event Control Register" • Set INT_SPOOF_EN bit to 1 Note: The reset default value of INT_SPOOF_EN is 1.	"SREP Port-Write Transmit on Logical/Transport Event Control Register" • Set PW_SPOOF_EN bit to 1
L_R2I_PERR	"SREP R2I LUT and Parity Control Register" • Set at least one of LUT_PAR_DIS or BAR_PAR_DIS to 0	"SREP Logical and Transport Layer Error Logging Enable CSR" • Set R2I_PERR_EN bit to 1 For more information, see "RapidIO-to-Bridge ISF Parity Error"	"SREP Interrupt on Logical/Transport Event Control Register" • Set INT_R2I_PERR_EN bit to 1 Note: The reset default value of INT_R2I_PERR_EN is 1.	"SREP Port-Write Transmit on Logical/Transport Event Control Register" Set PW_R2I_PERR_EN bit to 1

Table 58: Logical/Transport Layer Event and Notification Control (Continued)

Event Bit	Event Enable	Notification and Logging Enable	Interrupt Enable	Port-Write Enable
L_OOB	Always detected.	"SREP Logical and Transport Layer Error Logging Enable CSR" • Set OOB_EN bit to 1 For more information, see "Bridging Logical I/O Requests to the Bridge ISF".	"SREP Interrupt on Logical/Transport Event Control Register" • Set INT_OOB_EN bit to 1 Note: The reset default value of INT_OOB_EN is 1.	"SREP Port-Write Transmit on Logical/Transport Event Control Register" • Set PW_OOB_EN bit to 1
L_NO_WR	Always detected.	"SREP Logical and Transport Layer Error Logging Enable CSR" • Set NO_WR_EN bit to 1 For more information, see "Bridging Logical I/O Requests to the Bridge ISF".	"SREP Interrupt on Logical/Transport Event Control Register" • Set INT_NO_WR_EN bit to 1 Note: The reset default value of INT_NO_WR_EN is 1.	"SREP Port-Write Transmit on Logical/Transport Event Control Register" • Set PW_NO_WR_EN bit to 1
L_NO_RD	Always detected.	"SREP Logical and Transport Layer Error Logging Enable CSR" • Set NO_RD_EN bit to 1 For more information, see "Bridging Logical I/O Requests to the Bridge ISF".	"SREP Interrupt on Logical/Transport Event Control Register" • Set INT_NO_RD_EN bit to 1 Note: The reset default value of INT_NO_RD_EN is 1.	"SREP Port-Write Transmit on Logical/Transport Event Control Register" • Set PW_NO_RD_EN bit to 1

Table 59: Logical/Transport Layer Event Handling

Event Bit	Event Clearing
L_ERR_RESP	"SREP Logical and Transport Layer Error Detect CSR" • Set L_ERR_RESP to 0 OR "SREP Logical and Transport Layer Error Clear CSR" • Set CL_ERR_RESP to 1
L_DB_ERR_RESP	"SREP Logical and Transport Layer Error Detect CSR" • Set L_DB_ERR_RESP to 0 OR "SREP Logical and Transport Layer Error Clear CSR" • Set CL_DB_ERR_RESP to 1
L_ILL_TRANS	"SREP Logical and Transport Layer Error Detect CSR" • Set L_ILL_TRANS to 0 OR "SREP Logical and Transport Layer Error Clear CSR" • Set CL_ILL_TRANS to 1
L_ILL_TARG	"SREP Logical and Transport Layer Error Detect CSR" • Set L_ILL_TARG to 0 OR "SREP Logical and Transport Layer Error Clear CSR" • Set CL_ILL_TARG to 1
L_RESP_TO	"SREP Logical and Transport Layer Error Detect CSR" • Set L_RESP_TO to 0 OR "SREP Logical and Transport Layer Error Clear CSR" • Set CL_RESP_TO to 1
L_UNEXP_RESP	"SREP Logical and Transport Layer Error Detect CSR" • Set L_UNEXP_RESP to 0 OR "SREP Logical and Transport Layer Error Clear CSR" • Set CL_UNEXP_RESP to 1
L_UNSUP_TRANS	"SREP Logical and Transport Layer Error Detect CSR" • Set L_UNSUP_TRANS to 0 OR "SREP Logical and Transport Layer Error Clear CSR" • Set CL_UNSUP_TRANS to 1

Table 59: Logical/Transport Layer Event Handling (Continued)

Event Bit	Event Clearing
L_R2I_TTL	"SREP Logical and Transport Layer Error Detect CSR" • Set L_R2I_TTL to 0 OR "SREP Logical and Transport Layer Error Clear CSR" • Set CL_R2I_TTL to 1
L_R2I_PERR	"SREP Logical and Transport Layer Error Detect CSR" • Set L_R2I_PERR to 0 OR "SREP Logical and Transport Layer Error Clear CSR" • Set CL_R2I_PERR to 1
L_SPOOF	"SREP Logical and Transport Layer Error Detect CSR" • Set L_SPOOF to 0 OR "SREP Logical and Transport Layer Error Clear CSR" • Set CL_SPOOF to 1
L_OOB	"SREP Logical and Transport Layer Error Detect CSR" • Set L_OOB to 0 OR "SREP Logical and Transport Layer Error Clear CSR" • Set CL_OOB to 1
L_NO_WR	"SREP Logical and Transport Layer Error Detect CSR" • Set L_NO_WR to 0 OR "SREP Logical and Transport Layer Error Clear CSR" • Set CL_NO_WR to 1
L_NO_RD	"SREP Logical and Transport Layer Error Detect CSR" • Set L_NO_RD to 0 OR "SREP Logical and Transport Layer Error Clear CSR" • Set CL_NO_RD to 1

Table 60: Logical/Transport Layer Status Event and Notification Control

Event Bit	Event Enable	Notification and Logging Enable	Interrupt Enable	Port-Write Enable
L_BAD_REG_ACC	Checking for incorrect size is always enabled. To enable checking for the source of register transactions, program the following registers: • "SREP Register Access Source ID Checking Control Register" • "SREP Register Access Small Source ID Checking Control Register" • "SREP Large Register Access Source ID Checking Register"	"SREP R2I Event Status Logging Enable Register" Set BAD_REG_ACC_EN bit to 1	"SREP Interrupt on Logical/Transport Status Event Control Register" • Set INT_BAD_REG_ACC_EN bit to 1	"SREP Port-Write Transmit on Logical/Transport Status Event Control Register" • Set PW_BAD_REG_ACC_EN bit to 1
L_LUT_BND	Always detected.	"SREP R2I Event Status Logging Enable Register" • Set LUT_BND_EN bit to 1 For more information, see "RapidIO LUT Entry Boundary Crossing Event"	"SREP Interrupt on Logical/Transport Status Event Control Register" • Set INT_LUT_BND_EN bit to 1 Note: The reset default value of INT_LUT_BND_EN is 1.	"SREP Port-Write Transmit on Logical/Transport Status Event Control Register" Set PW_LUT_BND_EN bit to 1

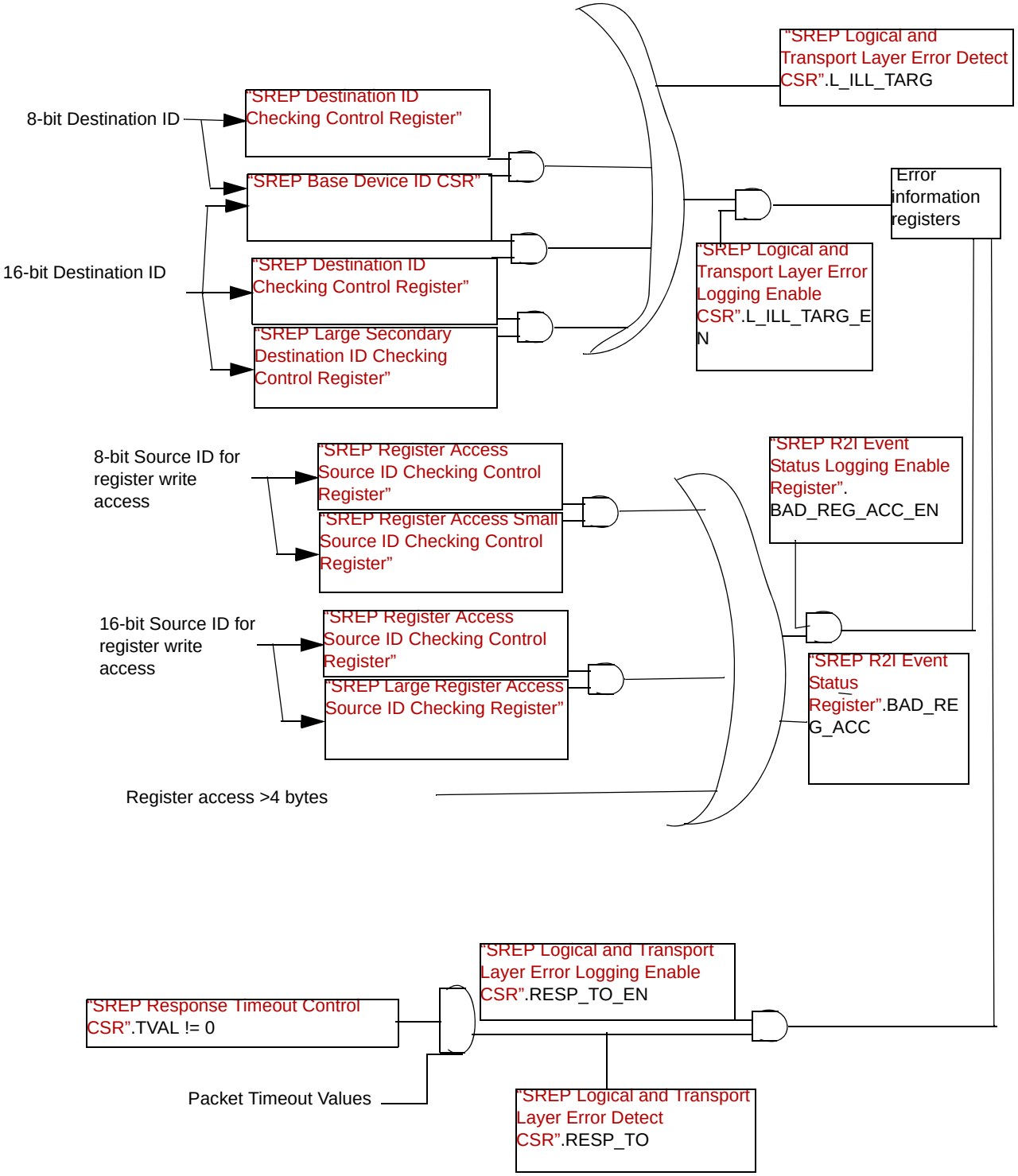
Table 60: Logical/Transport Layer Status Event and Notification Control (Continued)

Event Bit	Event Enable	Notification and Logging Enable	Interrupt Enable	Port-Write Enable
L_PKT_CRC	For information on the control of CRC error checking, see the MAC documentation.	No error information is latched by the Logical Layer for Packet CRC Events.	"SREP Interrupt on Logical/Transport Status Event Control Register" • Set INT_PKT_CRC_EN bit to 1 Note: The reset default value of INT_PKT_CRC_EN is 1.	It is not possible to send a Port-Write for a packet CRC error.
L_PKT_STOMP	Always detected.	No error information is latched by the Logical Layer for Packet Stomp Events.	"SREP Interrupt on Logical/Transport Status Event Control Register" • Set INT_PKT_STOMP_EN bit to 1 Note: The reset default value of INT_PKT_STOMP_EN is 1.	It is not possible to send a Port-Write for a STOMPed packet error.

Table 61: Logical/Transport Layer Status Event Handling

Event Bit	Event Clearing
L_BAD_REG_ACC	"SREP R2I Event Status Register" • Set L_BAD_REG_ACC to 1
L_LUT_BND	"SREP R2I Event Status Register" • Set L_LUT_BND to 1
L_PKT_CRC	"SREP R2I Event Status Register" • Set L_PKT_CRC to 1
L_PKT_STOMP	"SREP R2I Event Status Register" • Set L_PKT_STOMP to 1

Figure 28: Logical/Transport Layer Event Enable/Reporting Diagram



11.7.3 Bridge ISF Logical Error Detect Register Event Information

Bridge ISF Logical events are all in the "SREP ISF Logical Error Detect CSR". The following tables describe how to enable detection of events, how to control information logging for events, how to control event notification, and how to clear events detected in the "SREP ISF Logical Error Detect CSR"..

For any event detected in the "SREP ISF Logical Error Detect CSR" to cause an interrupt, the ISF_LOG_EN bit in the "SREP Interrupt Event Enable Register" must be set to 1.

For any event detected in the "SREP ISF Logical Error Detect CSR" to cause a port-write, the ISF_LOG_EN bit in the "SREP Port-Write Event Enable Register" must be set to 1.

The 'Event Clearing' column of Table 63 describes how to clear the event. An alternative to clearing the event is to disable further event notification for the event (see Table 62).

The relationship among the event enable control registers for each Bridge ISF Logical event is shown in Figure 28. Most Bridge ISF Logical events are checked for by the SREP. Response Timeout events have additional registers that must be programmed in order to be detected.

Table 62: Bridge ISF Logical/Transport Layer Event and Notification Control

Event Bit	Event Enable	Notification and Logging Enable	Interrupt Enable	Port-Write Enable
I_ERR_RESP	Always detected.	"SREP ISF Logical Error Logging Enable CSR" • Set ERR_RESP_EN bit to 1	"SREP Interrupt on ISF Event Control Register" • Set INT_ERR_RESP_EN bit to 1 Note: The reset default value of INT_ERR_RESP_EN is 1.	"SREP Port-Write Transmit on ISF Event Control Register" • Set PW_ERR_RESP_EN bit to 1
I_RESP_TO	"SREP ISF Response Timeout Register" • Set TVAL to appropriate non-zero interval	"SREP ISF Logical Error Logging Enable CSR" • Set RESP_TO_EN bit to 1	"SREP Interrupt on ISF Event Control Register" • Set INT_RESP_TO_EN bit to 1 Note: The reset default value of INT_RESP_TO_EN is 1.	"SREP Port-Write Transmit on ISF Event Control Register" • Set PW_RESP_TO_EN bit to 1

Table 62: Bridge ISF Logical/Transport Layer Event and Notification Control (Continued)

Event Bit	Event Enable	Notification and Logging Enable	Interrupt Enable	Port-Write Enable
I_UNEXP_RESP	Always detected.	"SREP ISF Logical Error Logging Enable CSR" • Set UNEXP_RESP_EN bit to 1	"SREP Interrupt on ISF Event Control Register" • Set INT_UNEXP_RESP_EN bit to 1 Note: The reset default value of INT_UNEXP_RESP_EN is 1.	"SREP Port-Write Transmit on ISF Event Control Register" • Set PW_UNEXP_RESP_EN bit to 1
I_UNSUP_TRANS	Always detected.	"SREP ISF Logical Error Logging Enable CSR" • Set UNSUP_TRANS_EN bit to 1 For more information, see "Bridge ISF Unsupported Transaction Events"	"SREP Interrupt on ISF Event Control Register" • Set INT_UNSUP_TRANS_EN bit to 1 Note: The reset default value of INT_UNSUP_TRANS_EN is 1.	"SREP Port-Write Transmit on ISF Event Control Register" • Set PW_UNSUP_TRANS_EN bit to 1
I_R2R_TTL	"SREP I2R Transaction Time-To-Live Register" • Set TVAL to appropriate non-zero timeout value.	"SREP ISF Logical Error Logging Enable CSR" • Set R2R_TTL_EN bit to 1 For more information, see "R2R Time-to-Live Expired"	"SREP Interrupt on ISF Event Control Register" • Set INT_R2R_TTL_EN bit to 1	"SREP Port-Write Transmit on ISF Event Control Register" • Set INT_R2R_TTL_EN bit to 1
I_REG_TTL	"SREP I2R Transaction Time-To-Live Register" • Set TVAL to appropriate non-zero timeout value.	"SREP ISF Logical Error Logging Enable CSR" • Set REG_TTL_EN bit to 1 For more information, see "Register Access Response Time-to-Live Expired"	"SREP Interrupt on ISF Event Control Register" • Set INT_REG_TTL_EN bit to 1	"SREP Port-Write Transmit on ISF Event Control Register" • Set INT_REG_TTL_EN bit to 1

Table 62: Bridge ISF Logical/Transport Layer Event and Notification Control (Continued)

Event Bit	Event Enable	Notification and Logging Enable	Interrupt Enable	Port-Write Enable
I_I2R_TTL	"SREP I2R Transaction Time-To-Live Register" • Set TVAL to appropriate non-zero timeout value.	"SREP ISF Logical Error Logging Enable CSR" • Set I2R_TTL_EN bit to 1 For more information, see "I2R Transaction Time-to-Live Expired"	"SREP Interrupt on ISF Event Control Register" • Set INT_I2R_TTL_EN bit to 1	"SREP Port-Write Transmit on ISF Event Control Register" • Set PW_I2R_TTL_EN bit to 1
I_TEA	Enable transmission timeout checking in the Bridge ISF.	Enable transmission timeout checking in the Bridge ISF.	"SREP Interrupt on ISF Event Control Register" • Set INT_TEA_EN bit to 1	"SREP Port-Write Transmit on ISF Event Control Register" • Set PW_TEA_EN bit to 1
I_LUT_BND	Always detected.	"SREP ISF Logical Error Logging Enable CSR" • Set LUT_BND_EN bit to 1 For more information, see "Bridge ISF LUT Entry Boundary Crossing Event"	"SREP Interrupt on ISF Event Control Register" • Set INT_LUT_BND_EN bit to 1 Note: The reset default value of INT_LUT_BND_EN is 1.	"SREP Port-Write Transmit on ISF Event Control Register" Set PW_LUT_BND_EN bit to 1
I_BE_DISCONT	Always detected.	"SREP ISF Logical Error Logging Enable CSR" • Set BE_DISCONT_EN bit to 1 For more information, see "Bridge ISF Byte Enables Discontiguous Event"	"SREP Interrupt on ISF Event Control Register" • Set INT_BE_DISCONT_EN bit to 1 Note: The reset default value of INT_BE_DISCONT_EN is 1.	"SREP Port-Write Transmit on ISF Event Control Register" Set PW_BE_DISCONT_EN bit to 1

Table 62: Bridge ISF Logical/Transport Layer Event and Notification Control (Continued)

Event Bit	Event Enable	Notification and Logging Enable	Interrupt Enable	Port-Write Enable
I_I2R_PERR	"SREP I2R LUT and BAR Parity Control Register" • Set at least one of LUT_PAR_DIS or BAR_PAR_DIS to 0	"SREP ISF Logical Error Logging Enable CSR" • Set I2R_PERR_EN bit to 1	"SREP Interrupt on ISF Event Control Register" • Set INT_I2R_PERR_EN bit to 1	"SREP Port-Write Transmit on ISF Event Control Register" • Set PW_I2R_PERR_EN bit to 1
I_ECC_ERR	"SREP ISF ECC Control Register" • Set at least one of INB_ECC_CHK_EN, OUTB_ECC_CHK_EN, or ISF_ECC_CHK_EN to 1.	"SREP ISF Logical Error Logging Enable CSR" • Set ECC_ERR_EN bit to 1	"SREP Interrupt on ISF Event Control Register" • Set INT_ECC_ERR_EN bit to 1	"SREP Port-Write Transmit on ISF Event Control Register" • Set PW_ECC_ERR_EN bit to 1
I_UNSAR_REQ	Always detected.	"SREP ISF Logical Error Logging Enable CSR" • Set UNSAR_REQ_EN bit to 1	"SREP Interrupt on ISF Event Control Register" • Set INT_UNSAR_REQ_EN bit to 1	"SREP Port-Write Transmit on ISF Event Control Register" • Set PW_UNSAR_REQ_EN bit to 1
I_OOB	Always detected. Note: To transmit RapidIO requests, the MAST_EN bit in the "SREP General Control CSR" must be set to 1.	"SREP ISF Logical Error Logging Enable CSR" • Set OOB_EN bit to 1 For more information, see "RapidIO OOB Request Event".	"SREP Interrupt on ISF Event Control Register" • Set INT_OOB_EN bit to 1 Note: The reset default value of INT_OOB_EN is 1.	"SREP Port-Write Transmit on ISF Event Control Register" • Set PW_OOB_EN bit to 1

Table 62: Bridge ISF Logical/Transport Layer Event and Notification Control (Continued)

Event Bit	Event Enable	Notification and Logging Enable	Interrupt Enable	Port-Write Enable
I_NO_WR	Always detected.	"SREP ISF Logical Error Logging Enable CSR" • Set NO_WR_EN bit to 1 For more information, see "RapidIO Write Denied Event" .	"SREP Interrupt on Logical/Transport Event Control Register" • Set INT_NO_WR_EN bit to 1 Note: The reset default value of INT_NO_WR_EN is 1.	"SREP Port-Write Transmit on Logical/Transport Event Control Register" • Set PW_NO_WR_EN bit to 1
I_NO_RD	Always detected.	"SREP ISF Logical Error Logging Enable CSR" • Set NO_RD_EN bit to 1 For more information, see "RapidIO Read Denied Event".	"SREP Interrupt on ISF Event Control Register" • Set INT_NO_RD_EN bit to 1 Note: The reset default value of INT_NO_RD_EN is 1.	"SREP Port-Write Transmit on ISF Event Control Register" • Set PW_NO_RD_EN bit to 1

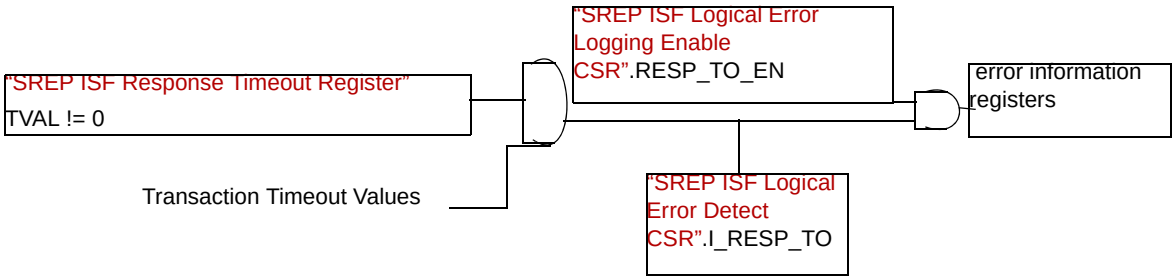
Table 63: Bridge ISF Logical/Transport Layer Event Handling

Event Bit	Event Clearing
I_ERR_RESP	"SREP ISF Logical Error Detect CSR" • Set I_ERR_RESP to 1
I_RESP_TO	"SREP ISF Logical Error Detect CSR" • Set I_RESP_TO to 1
I_UNEXP_RESP	"SREP ISF Logical Error Detect CSR" • Set I_UNEXP_RESP to 1
I_UNSUP_TRANS	"SREP ISF Logical Error Detect CSR" • Set I_UNSUP_TRANS to 1
I_R2R_TTL	"SREP ISF Logical Error Detect CSR" • Set I_R2R_TTL to 1
I_REG_TTL	"SREP ISF Logical Error Detect CSR" • Set I_REG_TTL to 1

Table 63: Bridge ISF Logical/Transport Layer Event Handling (Continued)

Event Bit	Event Clearing
I_I2R_TTL	"SREP ISF Logical Error Detect CSR" • Set I_I2R_TTL to 1
I_TEA	"SREP ISF Logical Error Detect CSR" • Set I_TEA to 1
I_LUT_BND	"SREP ISF Logical Error Detect CSR" • Set I_LUT_BND to 1
I_BE_DISCONT	"SREP ISF Logical Error Detect CSR" • Set I_BE_DISCONT to 1
I_I2R_PERR	"SREP ISF Logical Error Detect CSR" • Set I_I2R_PERR to 1
I_ECC_ERR	"SREP ISF Logical Error Detect CSR" • Set I_ECC_ERR to 1
I_UNSAR_REQ	"SREP ISF Logical Error Detect CSR" • Set I_UNSAR_REQ to 1
I_OOB	"SREP ISF Logical Error Detect CSR" • Set I_OOB to 1
I_NO_WR	"SREP ISF Logical Error Detect CSR" • Set I_NO_WR to 1
I_NO_RD	"SREP ISF Logical Error Detect CSR" • Set I_NO_RD to 1

Figure 29: Bridge ISF Logical RTO Event Enable Diagram



11.7.4 Implementation-Specific Physical Layer Events and Event Handling

"SREP Interrupt Status Register".IMPL_PHY_ERR bit summarizes the status of the implementation-specific physical layer error events detected by the physical layer. For more information on these errors, see "Physical Layer Events".

For any Implementation-Specific Physical Layer Event to cause an interrupt, the IMP_PHY_ERR_EN bit in the "SREP Interrupt Event Enable Register" must be set to 1.

For any Implementation-Specific Physical Layer Event to cause a port-write, the IMPL_PHY_ERR_EN bit in the "SREP Port-Write Event Enable Register" must be set to 1.

12. Bridge ISF

Topics discussed include the following:

12.1 Overview

The Bridge ISF interconnects the Tsi620's PCI Interface and SREP by providing a low latency transaction flow. The Bridge ISF has the following features:

- Contains two 64-bit ports
- Operates at either 125 or 156.25 MHz
- Supports simultaneous block-to-block and non-blocking transactions
- Provides ECC generation and error correction/detection on transfers
- Provides a peak throughput of 10 Gbps from port to port
- Does not require configuration by the end user

12.2 Port Numbering

The Bridge ISF ports are numbered as in the following table.

Table 64: Bridge ISF Port Numbering

Bridge ISF Port Number	Block
Port 0	SREP
Port 1	PCI Interface

12.3 ECC Protection

The Bridge ISF offers ECC protection in its data path. The transmitter generates ECC words on the ingress side of the Bridge ISF, while ECC is checked by the receiver on the egress side of the Bridge ISF. Single bit errors are corrected and reported. Double-bit errors cannot be corrected, and are reported only.

Errors that affect more than two bits may not be detected, or may be interpreted as single-bit errors and erroneously corrected.

12.3.1 Enabling ECC

ECC protection is enabled by setting the ECC_EN bit in the "Bridge ISF Control Register".

There is a latency impact when ECC protection is enabled. Two extra clock cycles are required: one cycle to generate the ECC check words, and a second to check ECC. Disabling ECC removes this latency impact.

12.3.2 ECC Generation

Error protection is used as two 8-bit ECC check words. One ECC word for the upper 64-bits of the Bridge ISF's data path, and the second for the lower 64-bits.

12.3.3 Correctable Error Detection

Correctable errors are single bit errors in either the data word or the ECC check word. Any single bit error can be detected and corrected. These errors are reported in the "Bridge ISF ECC CE Interrupt Status Register". An interrupt is raised if the corresponding EN bit is set in the "Bridge ISF ECC CE Interrupt Enable Register". For test purposes, the status bit can be set by software by writing a 1 to the corresponding bit in the "Bridge ISF ECC CE Interrupt Set Register".

12.3.4 Uncorrectable Error Detection

Uncorrectable errors are double-bit errors in either the data word or the ECC check word. Double-bit errors can be detected but not corrected. These errors are reported in the "Bridge ISF ECC UE Interrupt Status Register". An interrupt is raised if the corresponding EN bit is set in the "Bridge ISF ECC UE Interrupt Enable Register". For test purposes, the status bit can be set by software by writing a 1 to the corresponding bit in the "Bridge ISF ECC UE Interrupt Set Register".

12.3.5 Error Logging

When an ECC error is detected — correctable or uncorrectable — the first data phase of the transaction with the error is divided and latched in three registers: "SREP ISF Logical Error Upper Attributes Capture CSR", "SREP ISF Logical Error Middle Attributes Capture CSR", and "SREP ISF Logical Error Lower Attributes Capture CSR". The error logging must first be enabled by clearing the LOCKED bit in the "Bridge ISF Port Error Status Register". The contents of these registers are preserved for recovery after a reset, which may be required to clear the problem.

12.3.6 Global Status Reporting

Both correctable and uncorrectable errors are reported to a global status reporting function. This function exists in the "Block Event Status Register", as well as the Tsi620 I^2C Interface through the "Externally Visible I2C Status Register". The global status reporting signals can be exercised using the corresponding SET bit in the "Bridge ISF ECC CE Interrupt Set Register" and the "Bridge ISF ECC UE Interrupt Enable Register".

12.3.7 ECC Configuration for Tsi620

The PCI Interface does not support ECC generation or checking. Therefore, the Bridge ISF's ECC checking and generation function must be used for the PCI Interface.

The SREP supports ECC end-to-end. Therefore, the Bridge ISF's ECC checking and generation function should be used for the SREP port in order to extend ECC coverage to the RapidIO portion of the Tsi620. Also note that once a Bridge ISF transaction is translated to a RapidIO packet, the data is protected by the RapidIO CRC code.

12.3.8 Transaction Timeout

Two types of timeouts can occur in the Bridge ISF. Transactions may spend too long in the request queue waiting for service from the destination port; or a request may be made to a destination that requires a response and the response is delayed too long. These are briefly explained.

12.3.8.1 Timeout on Transaction Transfer

The Bridge ISF provides a function for timing out a transfer if it spend too long at the head of the request queue. When a transaction is placed at the head of the queue, a timer is initialized. If the timer expires, the Bridge ISF indicates to the source that the transaction transfer was unable to complete. The source must then take appropriate action.

Each time the queue is reordered and a new transaction is placed at the head of the queue, the timer is re-initialized. Only the time a transaction spends at the head of the queue is timed. Because there is only a single timer, low priority transactions may spend much longer in the queue than the value of the timer.

If the timeout expires, the request is removed from the queue and (optionally) an interrupt is raised. The transaction transfer timeout on the Tsi620 is 65535 Bridge ISF clock periods. The timeout is set using "Bridge ISF Port TEA Timeout Register". Bridge ISF TEA status is located in each block, as well as in the "Bridge ISF TEA Interrupt Status Register". The "Bridge ISF TEA Interrupt Enable Register" controls event notification for Bridge ISF TEA timeouts, while the "Bridge ISF TEA Interrupt Set Register" creates TEA errors for software testing of Bridge ISF TEA events.

12.3.8.2 Timeout on Response

The Bridge ISF does not track the time between when a request is made to a destination and the response that is returned. The source determines if the response is too long in returning, and whether or not to take appropriate action.

13. PCI Interface

Topics discussed include the following:

13.1 Overview

The PCI Interface transfers PCI data between the bus and the Bridge ISF, and vice versa. This design allows other interfaces that are connected to the Bridge ISF to initiate transactions on the PCI bus. This also allows PCI devices to initiate transactions that access Tsi620 internal registers as well as other devices connected to the Bridge ISF.

The PCI Interface has the following features:

- Compliant with the following specifications:
  - *PCI Local Bus Specification (Revision 2.3)*
  - *PCI Bus Power Management Interface Specification (Revision 1.1)*
- 32- or 64-bit addressing
- 32-bit data bus
- PCI operation from 25 to 66 MHz
- Provides bus arbitration for four external PCI devices
- Supports vital product data (VPD)
- Supports message signaled interrupts (MSIs)
  - Generates outgoing MSIs
  - Handles incoming MSIs as posted write operations
- PCI master capability
  - Generates all *PCI Local Bus Specification (Revision 2.3)* commands except IACK and Special cycle
  - Multi-threading for PCI delayed read cycles
- PCI target capability
  - Accepts *PCI Local Bus Specification (Revision 2.3)* commands except IACK, Special cycle, I/O read, and I/O write
  - No target wait states
  - Supports up to four concurrent reads
  - Supports posted writes, delayed reads, and config type 0 writes
- Supports direct address translation (that is, the PCI address is passed unmodified directly to the Bridge ISF) and indexed address translation with the Bridge ISF
- 3.3V compliant I/Os; 5V is not supported
- Register support
  - Accessed through multi-master internal register bus
  - Full runtime access of PCI configuration space through PCI or internal register bus

The PCI Interface does not support the following:

- PCI lock command. In a switch fabric architecture, this type of command can cause transaction congestion and deadlock scenarios.
- 64-bit data bus
- Compact PCI Hot Swap
- Less than 25-MHz minimum speed of operation, as defined in the *PCI Local Bus Specification (Revision 2.3)*

13.1.1 Terms

The following terms are used in this chapter.

Table 65: PCI Interface Terminology

Term	Definition	Use relating to
PW	Abbreviation for Posted Write Transaction	O2P or P2O Queues
RQ	Abbreviation for Request Transaction	O2P or P2O Queues
RC	Abbreviation for Completion Transaction	O2P or P2O Queues

13.2 Design Considerations

The following are some descriptions of PCI Interface-specific behaviors. These are described from a PCI centric viewpoint, unless otherwise noted. This means that the term target applies to a PCI Interface that is selected by another PCI master device, attached to its local PCI segment. These are not comprehensive transaction descriptions, but illustrations of how crossing between PCI and the Bridge ISF constrains (somewhat) the PCI protocol.

13.2.1 Functional Constraints

The PCI Interface is compatible with the *PCI Local Bus Specification (Revision 2.3)*. The following list, however, discusses several constraints when using the PCI Interface:

- Generates all *PCI Local Bus Specification (Revision 2.3)* commands except IACK and Special cycle
- Lock command not supported due to impact on Bridge ISF
- Multiple secondary Expansion ROM access by POST code (behind a single address window) is not supported
- The PCI Interface does not claim Type 1 Configuration cycles

13.2.2 Allowable Disconnect Boundaries

An Allowable Disconnect Boundary (ADB) is a PCI-X term that represents a naturally aligned 128-byte boundary. In PCI-X, bus masters and targets are permitted to disconnect burst transactions only on ADBs. The PCI Interface uses elements of this concept in PCI for queue management.

As a target device, the PCI Interface disconnects a transfer on a 2xADB (256-byte address boundary). The disconnect occurs on memory write and PFM read transfers. This allows the PCI Interface to use simpler packetizing algorithms when transferring data from PCI to the Bridge ISF. As a mastering device, the PCI Interface does not force disconnects on ADBs unless the transfer size and alignment originating from the Bridge ISF fall on an ADB.

13.2.3 Prefetchable Memory Reads

The *PCI Local Bus Specification (Revision 2.3)* has three commands for memory reads (see the following list). The PCI Interface complies with PCI recommendations for the amount of data fetched for each command:

- Memory read – Fetch 1 Dword of data
- Memory read line – Fetch 1 cacheline of data
- Memory read multiple – Fetch 2 cachelines of data

13.2.4 I/O Reads/Writes

The Tsi620 PCI Target does not support I/O Reads or I/O Writes.

13.2.5 Configuration Operations

All configuration operations (memory-mapped or Type 0 configuration) to the PCI Interface must be 32 bits in width. The PCI Interface retries and forwards all transactions. It then uses snooped information from the transactions to initiate register bus transactions. These transactions require a response from the register bus target port. When the response is received, the PCI Interface completes the transaction on the PCI side as a delayed operation.

Because the most common method for generating PCI_IDSELn is to resistively connect PCI_IDSELn to one of PCI_AD, external devices that need to execute configuration cycles must pre-drive the PCI_AD bus for four clock cycles before asserting PCI_FRAMEn. The Tsi620 pre-drives PCI_AD before asserting PCI_FRAMEn when it issues configuration read or write cycles.

The *PCI Local Bus Specification (Revision 2.3)* does not specify the number of cycles PCI_AD must be driven before PCI_FRAMEn is asserted for a configuration cycle. The Tsi620 pre-drives PCI_AD for four clock cycles when enabled.

The selection of the resistor between a bit of PCI_AD and PCI_IDSELn must consider the capactive load of the PCI_IDSELn trace and device load, and the time allowed to charge and discharged PCI_IDSELn to valid PCI logic levels. The resistor chosen for PCI 2.3 systems, must be such that PCI_IDSELn can be charged and discharged in the time allotted by PRE_DRIVE in the “PCI Miscellaneous Control and Status Register”.

13.2.6 Parity Across PCI

PCI parity is checked at the PCI Interface. Because parity is not carried through the Bridge ISF, parity information is not passed from the PCI Interface through the Bridge ISF to another Tsi620 interface.

If the PCI Interface detects a parity error on incoming data, the appropriate status bits are set in the "PCI Registers". If the data is a read response, the PCI Interface discards the data and returns an error response packet to the initiating Tsi620 port. If the data is a part of a write transaction, the data is discarded and is not forwarded to the destination Tsi620 port.

13.3 Transaction Mapping

The PCI Interface handles three classes of transactions: Posted Writes, Completions, and Requests. Because the Bridge ISF is PCI-centric, the conversion from PCI commands to Bridge ISF commands is minimal.

13.3.1 PCI Posted Writes

PCI operations are mapped directly from the source transaction. The only conversion that is made is PCI memory write and invalidates; these are converted to memory writes.

13.3.2 PCI Requests

PCI read transactions are mapped directly from the source transaction. Table 66 describes the conversion from a PCI transaction to a Bridge ISF destination block. PCI write requests (configuration writes and I/O writes) are mapped directly between PCI and the Bridge ISF.

Table 66: PCI to Bridge ISF Read Command Conversion

PCI Source Command	Bridge ISF Destination Command
Configuration read	Not Supported
I/O read	Not supported
Memory read	Memory read dword
Memory read line	Memory read block
Memory read multiple	Memory read block

13.4 Transaction Ordering Rules

The PCI Interface adheres to the following ordering rules, which are compliant with the PCI specifications:

- Posted Memory Writes (PMW) must be able to pass Requests (Read Request - RR, Write Request - WR) and Completions (Read Completion - RC, Write Completion - WC)
- No transactions can pass Posted Memory Writes (PMW), including other PMWs

- Completions (Read Completion - RC, Write Completion - WC) must be able to pass Requests (Read Request - RR, Write Request - WR), and can pass other Completions with different sequence identifications
- Requests (RR, WR) can only pass other Requests

Table 67: PCI Ordering Rules

Can Row pass column?	PMW	RR	WR	RC	WC
Posted Memory Write (PMW)	No	Yes	Yes	Yes	Yes
Split Read Request (RR)	No	Yes	Yes	No	No
Split Write Request (WR)	No	Yes	Yes	No	No
Split Read Completion (RC)	No	Yes	Yes	Yes	Yes
Split Write Completion (WC)	No	Yes	Yes	Yes	Yes

Posted memory writes have the highest priority, followed by responses at the second highest priority, and by requests at the lowest priority.

13.5 Addressing

The Tsi620 uses base address registers (BARs) and translation methods to decode and translate PCI transactions. The type of BAR and translation method used depends on two things:

- Transaction direction – Whether the transaction originated from the PCI Interface or from the Bridge ISF.
- Transaction address – Where the transaction address falls in the memory map (for example, Configuration space versus I/O space).

The following table summarizes the different transaction flows and modes that are supported by the PCI Interface.

Table 68: PCI Addressing Overview

Transaction Direction (Source to Destination)	PCI Address Decode	Bridge ISF Address Translation	Destination Port Lookup	Bridge ISF Address Decode	PCI Address Translation
Bridge ISF to PCI	n/a	n/a	n/a	PFAB_BARs	Remap registers
PCI to Bridge ISF	P2O_BARs	Indexed	Indexed	n/a	n/a

Tsi620's PCI addressing is also highly configurable. The addressing modes, registers, and lookup tables can be loaded using either of the following methods:

- At reset from a serial EEPROM
- At power-on, self-test (*POST*) time
- At run-time by the operating system

13.5.1 Bridge ISF to PCI Address Decoding

The PCI Interface uses five base address registers (BARs), or windows, to decode Bridge ISF transactions that are for the PCI bus (see Table 69). If a transaction falls within the address window of a Bridge ISF BAR, it decodes the transaction and uses direct remapping tables (if applicable) to translate the Bridge ISF address to a PCI address. The PCI Interface then completes the transaction on the PCI bus.

Table 69: Bridge ISF-to-PCI Address Decoding (and Decode Order)

Address Window/Decode Order[a]	BAR[b]	Address Translation
1. Configuration Memory (64-bit)	PFAB_BAR0	None
	PFAB_BAR0_UPPER	
2. I/O Memory (64-bit)	PFAB_IO	None
	PFAB_IO_UPPER	
3. Non-prefetchable Memory (32-bit)	PFAB_MEM32	Direct Remap
4. Prefetchable Memory (64-bit)	PFAB_PFM3	Direct Remap
5. Prefetchable Memory (64-bit)	PFAB_PFM4	Direct Remap

a. If an address decodes into more than one BAR window, the transaction is interpreted according to the decode order, with the lowest number taking precedence. This allows part of a large memory window to be efficiently used for I/O or Configuration space.

b. The PFAB_BAR address windows are optional and are not required to access the PCI bus. When a Bridge ISF packet is destined for the PCI Interface it is executed on the PCI bus as a memory transaction unless it falls within the I/O or Configuration PFAB_BAR.

The Configuration and I/O BARs enable Bridge ISF initiators to generate configuration and I/O PCI transactions if the Bridge ISF initiator's interface does not support configuration and I/O cycles (for example, the SREP can only perform Bridge ISF memory reads and writes). By directing a Bridge ISF read packet at the configuration BAR, the PCI Interface can generate a configuration transaction on the PCI (similarly for I/O transactions).

The MEM32 BAR provides a translation of a Bridge ISF address in which the upper 32 bits (63:32) == 0. The PFM BARs provide a translation of a full 64-bit address and may generate a dual address cycle (if 63:32 is not equal to 0), or a single address cycle (if 63:32 == 0).

The Bridge ISF translates addresses that are decoded to the MEM32 and PFM BARs using direct remap (configuration and I/O transaction addresses are not translated). This method of translation requires that bits in the address defined by a 1 at that location in the BAR's mask register are replaced by the bit at the same bit location in the remap register.

13.5.1.1 Configuration Memory Window (PFAB_BAR0)

The Bridge ISF uses this address window to generate configuration transactions on the destination PCI bus. The window has the following characteristics:

- Addressing – 64-bit.
- Size – Fixed at 16 MB.
- Enabled – Set BAR0_EN to 1 "PFAB_BAR0 Register".
- Defined – Write to PFAB_BAR0 and PFAB_BAR0_UPPER (see "PFAB_BAR0_UPPER Register").

This BAR generates dword transactions on the PCI bus. All Bridge ISF transactions that hit this BAR must not cross a dword boundary (that is, address[2:0] + size must be less-than or equal to 4). Violating this requirement causes undefined behavior.

The incoming Bridge ISF address is used as displayed in the following table.

Table 70: Bridge ISF and Incoming Configuration Address Bits

Incoming Configuration Address Bits	Used for...
Addr[63:24]	BAR decode.
Addr[23:16]	Outgoing bus number. The configuration type is based on the value in this field. For example, if the value matches our destination bus number then the outgoing configuration type is Type 0; at all other times it will be Configuration Type 1. The bus number is configured in the BUS_NUM field of the "PCI Bus Number Register".
Addr[15:11]	Outgoing device number.
Addr[10:8]	Outgoing function number.
Addr[7:2]	Outgoing register number.
Addr[1:0]	These bits have no effect and are undefined.

For Type 0 configuration cycles, the mapping of Addr[15:11] to PCI_AD[31:16] is displayed in the following figure.

Table 71: Mapping of Addr to PCI_AD[31:16] for Type 0 Configuration Cycles

Addr[15:11]	AD[31:16] for Type 0 Configuration Cycles
0 0000	0000 0000 0000 0001
0 0001	0000 0000 0000 0010
0 0010	0000 0000 0000 0100
0 0011	0000 0000 0000 1000
0 0100	0000 0000 0001 0000
0 0101	0000 0000 0010 0000
0 0110	0000 0000 0100 0000
0 0111	0000 0000 1000 0000
0 1000	0000 0001 0000 0000
0 1001	0000 0010 0000 0000
0 1010	0000 0100 0000 0000
0 1011	0000 1000 0000 0000
0 1100	0001 0000 0000 0000
0 1101	0010 0000 0000 0000
0 1110	0100 0000 0000 0000
0 1111	1000 0000 0000 0000
1 XXXX	0000 0000 0000 0000

13.5.1.2 I/O Memory Window (PFAB_IO)

The Bridge ISF uses this address window to generate I/O transactions on the destination PCI bus. The window has the following characteristics:

- Addressing – 64-bit.
- Size – Fixed at 64 KB.
- Enabled – Set EN to 1 in "PFAB_IO Base Address Register".
- Defined – Write to PFAB_IO and PFAB_IO_UPPER in "PFAB_IO_UPPER Register".

13.5.1.3 MEM32 Memory Window (PFAB_MEM32)

The Bridge ISF uses this address window to generate non-prefetchable transactions on the destination PCI bus. The window has the following characteristics:

- Addressing – 32-bit.

  Note. Address bits 63:32 are compared against 0 while address bits 31:29 are compared against values in the PFAB_MEM32 register.
- Size – 512 MB or 1 GB. The size is based on the value written to SIZE in the "PFAB_MEM32 Base Address Register" (see also Table 72).
- Enabled – Set PFAB_MEM32[EN] to 1.
- Defined – Write to BA in the "PFAB_BAR0 Register".

Table 72: MEM32 (Bridge ISF) Window Size

PFAB_MEM32_SIZE	Bridge ISF Window Size	Active BAR (Bits)
0	512 MB	OCN_AD[63:29]
1	1 GB	OCN_AD[63:30]

13.5.1.4 Prefetchable Memory Window (PFAB_PFM3)

The Bridge ISF uses this address window to generate prefetchable transactions on the destination PCI bus. The window has the following characteristics:

- Addressing – 64-bit.
- Size – 1 or 2 GB. The size is based on the value written to SIZE in the "PFAB_PFM3 Base Address Register" (see also Table 73).
- Enabled – Set PFAB_PFM3[EN] to 1.
- Defined – Write to PFAB_PFM3[BA].

Table 73: PFAB_PFM3 (Bridge ISF) Window Size

PFAB_PFM3[SIZE]	Bridge ISF Window Size	Active BAR (Bits)
0	1 GB	OCN_AD[63:30]
1	2 GB	OCN_AD[63:31]

13.5.1.5 Prefetchable Memory Window (PFAB_PFM4)

The Bridge ISF uses this address window to generate prefetchable transactions on the destination PCI bus. The window has the following characteristics:

- Addressing – 64-bit.
- Size – 1 or 2 GB. The size is based on the value written to SIZE in the "PFAB_PFM4 Base Address Register" (see also Table 74).
- Enabled – Set PFAB_PFM4[EN] to 1.
- Defined – Write to PFAB_PFM4[BA].

Table 74: PFAB_PFM4 (Bridge ISF) Window Size

PFAB_PFM4[SIZE]	Bridge ISF Window Size	Active BAR (Bits)
0	1 GB	OCN_AD[63:30]
1	2 GB	OCN_AD[63:31]

13.5.2 Bridge ISF to PCI Address Translation

If a Bridge ISF address falls in a PFAB_MEM32, PFAB_PFM3, or PFAB_PFM4 memory window, the Tsi620 uses remap and mask registers to translate the address to a PCI address (see Table 75). The remap register defines the address bits that comprise the translated PCI address, while the mask register determines which remap register bits are used in the translated PCI address.

For example, the Tsi620 checks the bit values programmed in the mask register. Bits configured as 0 indicate the original Bridge ISF address bit values must be used in the remapped PCI address. Bits configured as 1, however, must be replaced with the relevant values programmed in the remap register.

Table 75: PCI Address Remap Registers

Remap Register Name	Purpose	Translation
PFAB_MEM32_REMAP PFAB_MEM32_MASK	Mem32 PCI Window Base Mem32 Window Address Remap Mask	Direct Remap
PFAB_PFM3_REMAP_UPPER PFAB_PFM3_REMAP_LOWER PFAB_PFM3_MASK	Prefetchable Memory 3 PCI Window Base PFM 3 PCI Window Address Remap Mask	Direct Remap
PFAB_PFM4_REMAP_UPPER PFAB_PFM4_MASK_LOWER	Prefetchable Memory 4 PCI Window Base PFM 4 PCI Window Address Remap Mask	Direct Remap

The following sections provide examples of how the Tsi620 translates addresses that are decoded to the PFAB_MEM32, PFAB_PFM3, or PFAB_PFM4 memory windows.

13.5.2.1 PFAB_MEM32_REMAP and PFAB_MEM32_MASK Register Operation

The MEM32 remap operation sets the base address of the window through which all the Bridge ISF to PCI MEM32 operations are located. The window is a minimum of 4 KB. Unpredictable results can occur if the mask register does not have consecutive 1s in the high order bits and consecutive 0s in the low order bits (no alternating 1s and 0s).

Table 76: Address Translation using MEM32 Remap and Mask Registers — Example

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bridge ISF Address (Mem32)	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
PFAB_MEM32_REMAP	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R												
PFAB_MEM32_MASK[a]	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0												
PCI MEM32 Address	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A

a. The mask register can set the window smaller than the minimum BAR window size. Unless desired, this situation should be avoided since it can cause some aliasing of address windows.

For more information on the registers that are used for MEM32 address translation, see:

- "PFAB_MEM32_REMAP Register"
- "PFAB_MEM32_MASK Register"

13.5.2.2 PFAB_PFM3, 4_REMAP, and PFAB_PFM3,4_MASK Register Operation

The PFM remap operations set the base addresses of the one window through which all the Bridge ISF to PCI Prefetchable Memory operations are located. The two windows are a minimum of 4 KB. Unpredictable results can occur if either mask register does not have consecutive 1s in the high order bits and consecutive 0s in the low order bits (no alternating 1s and 0s.

Table 77: Address Translation using PFM3/PFM4 Remap and Mask Registers — High Order 32 bits

	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Bridge ISF Address (PFM)	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
PFAB_PFM3,4_ REMAP	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PFAB_PFM3,4_ MASK	Reserved - No Masking																				1	1	1	1	1	1	1	1	1	1	1	1
PCI PFM Address	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 78: Address Translation using PFM3/PFM4 Remap and Mask Registers — Low Order 32 bits

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bridge ISF Address (PFM)	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
PFAB_PFM3,4_ REMAP	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R												
PFAB_PFM3,4_ MASK	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Reserved - No Masking											
PCI PFM Address	R	R	R	R	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A

For more information on the registers that are used for PFM3 and PFM4 address translation, see:

- "PFAB_PFM3_REMAP_UPPER Register"
- "PFAB_PFM3_MASK Register"
- "PFAB_PFM4_REMAP_UPPER Register"
- "PFAB_PFM4_MASK Register"

13.5.3 PCI-to-Bridge ISF Address Decoding

The PCI Interface uses three base address registers (BARs), or windows, to decode PCI transactions that are for other Tsi620 ports (see Table 79). If a transaction falls within the address window of a PCI BAR, the PCI Interface decodes the transaction and uses its lookup tables (if applicable) to translate the PCI address to a Bridge ISF address. The Bridge ISF then transfers the transaction data to the SREP block.

Table 79: PCI-to-Bridge ISF Address Decoding

Address Window	BAR	Address Translation
1. Configuration Memory (32/64-bit)	P2O_BAR0	None
	P2O_BAR0_UPPER	
2. Prefetchable Memory (32/64-bit)	P2O_BAR2	Lookup Table (Address Indexed)
	P2O_BAR2_UPPER	
3. Prefetchable Memory (32/64-bit)	P2O_BAR3	Lookup Table (Address Indexed)
	P2O_BAR3_UPPER	

13.5.3.1 Configuration Memory Window (P2O_BAR0)

The PCI Interface uses this address window to decode PCI memory accesses of the device's internal registers. The window has the following characteristics:

- Addressing – 32- and 64-bit.
- Size – Fixed at 256 KB.
- Enabled – Set BAR0_EN to 1 in the "PCI Miscellaneous Control and Status Register".
- Defined – Write to "PCI Base Address Register 0" and "PCI Base Address Register 0 Upper".

The Tsi620 supports 256 KB of registers. All registers in the Switch and Bridge can be accessed through this register using PCI read and write transactions of 4 bytes or less. For more information on the Tsi620 register map, see "Register Access".

13.5.3.2 Prefetchable Memory Window (P2O_BAR2)

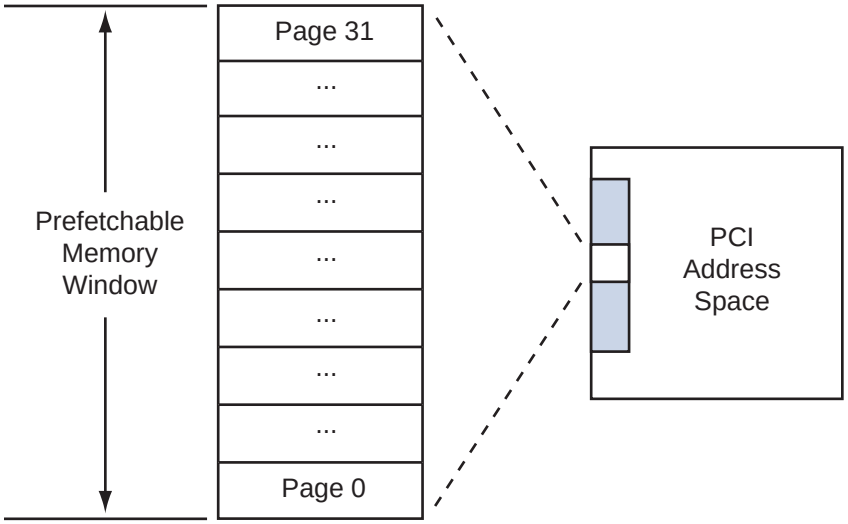
The PCI Interface uses this address window to decode prefetchable memory transactions on the PCI bus. The window has the following characteristics:

- Addressing – 32- and 64-bit.
- Size – 32 KB to 64 TB. The size is based on the value written to BAR2_SIZE in the "PCI Page Size Register".
- Enabled – Set P2O_PAGE_SIZES[BAR2_EN] to 1.
- Defined – Write to "PCI Base Address Register 2" and "PCI Base Address Register 2 Upper".

The Prefetchable memory window also contains 32 equal-sized pages of memory (see Figure 30). These pages have the following characteristics:

- Each page is 1/32 of the prefetchable window size.
- Each page acts as an additional decode window on the PCI bus. This implies that a prefetchable memory transaction is decoded based on where it falls in the 32 pages of memory.
- Each page size can be between 1 KB and 2 TB, and is based on the value written to BAR2_SIZE in the "PCI Page Size Register".

Figure 30: Prefetchable Memory Window



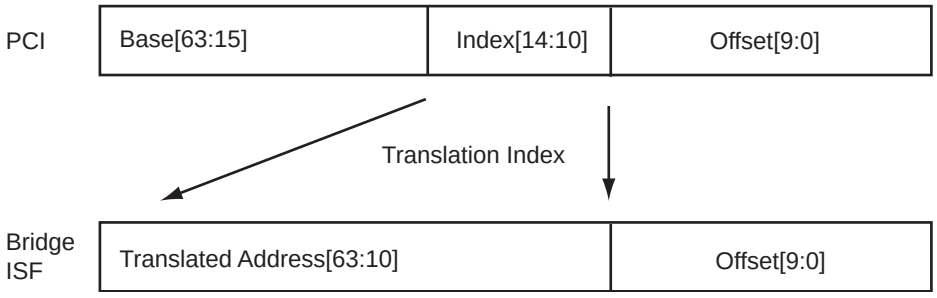
Once the PCI Interface decodes a prefetchable memory address, it is translated to a Bridge ISF address format and is passed to the appropriate Tsi620 port. The PCI Interface does this by dividing the address into three portions:

- Base – This portion is compared with the PCI address to determine if the transaction falls within the prefetchable memory window. This portion can be discarded during address translation provided translation is enabled, or it can be passed directly to the Bridge ISF.
- Index – This portion of the PCI address selects one of the 32 entries (pages) in the Address translation lookup tables. This portion can be discarded during address translation provided translation is enabled, or it can be passed directly to the Bridge ISF.
- Offset – This portion of the PCI address is left unchanged and is passed directly to the Bridge ISF.

Figure 31 shows an example of where the page size is programmed to 1 KB (BAR2_SIZE is defined as 00). The page size sets the *Offset* bits to be bits [9:0] of the BAR for addressing within the selected page.

The next 5 bits, [14:10], form the index that selects one of the 32 index table entries (pages). The remainder of the address is used as the Base for decoding memory address and claiming transactions; however, it is discarded during the translation process provided the P2O_PAGE_SIZES[BAR2_NOTRAN] is not enabled.

Figure 31: Prefetchable Memory Transaction — Address Translation



The translation lookup table provides address bits to replace the “Active BAR” and “Lookup Table Index” bits (see Table 187), as well as the Bridge ISF destination port. For example, for a window size of 32 KB, the lookup table provides address bits [63:10].

13.5.3.3 Prefetchable Memory Window (P2O_BAR3)

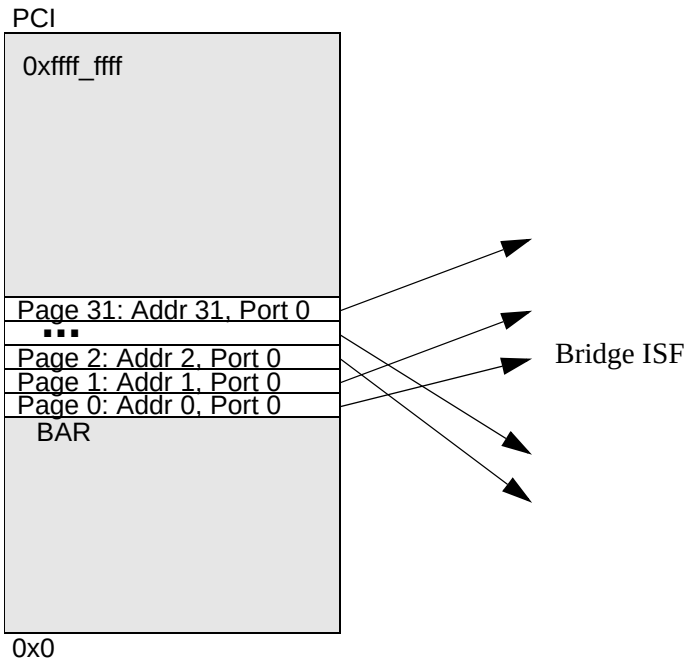
The 64-bit memory BAR, P2O_BAR3, operates identically to P2O_BAR2. Two BARs are provided to allow the user the ability to map one BAR into a prefetchable memory area and the other into a non-prefetchable memory area. The operation of the Tsi620 is unaffected by the area the BAR is mapped to. The external PCI device is responsible not to perform transactions that can cause destructive reads.

13.5.4 Route Mapping

Route mapping uses the same lookup table function as indexed address translation. The same index looks up the Bridge ISF destination port identifier.

Figure 32 shows that of the 32 pages within a PCI address window, each page can be mapped to one of the two Tsi620 Bridge ISF ports, PCI Interface and SREP. By default, all pages should be mapped to port 0 (for more information, “Port Numbering”).

Figure 32: Route Mapping



13.5.5 Indexed Lookup Table

The Address Index Lookup table can be configured by memory accesses from either PCI or the Bridge ISF. The number of bits that are used depends on the page size. For more information on Tsi620's lookup tables, see "P2O_BAR2_LUT{0..31} Register" through "P2O_BAR3_LUT_UPPER{0..31} Register". Figure 33 shows an example of the Address Index Lookup table.

Figure 33: PFM 3 Port, Address Indexed Lookup Table (Page Size 1K)

Page 31	Address [63:10]	Port[3:0]
Page 1	Address [63:10]	Port[3:0]
Page 0	Address [63:10]	Port[3:0]

13.6 Error Handling

The PCI Interface uses status and control registers to detect and report on three types of errors (each error type has two classes of errors):

- Parity
  - Address
  - Data
- Bus error
  - Master abort
  - Target abort
- Internal errors
  - Discard timer
  - Maximum retry counter expiring

Parity errors can occur on address or data phases of a PCI transaction. Address parity errors are fatal errors since devices in the system cannot determine whether or not the address falls within an image. The response to address parity errors is to ignore the cycle and drive PCI_SERRn if enabled. Data parity occurs during the data phase of the transaction and does not cause a cycle to terminate. The system usually detects the incorrect parity and drives PERR.

Bus errors consist of Master Abort and Target Abort. A Master Abort occurs when no target device claims a cycle with DEVSEL. After 5 clocks of no response, the initiating master releases the bus and sets internal status bits. A Target Abort occurs when a target device indicates an error occurred during the transfer.

Internal errors consist of discard timer and maximum retry counters expiring. The discard timer is used for each delayed transfer. If the external originating device never repeats a delayed cycle after a retry, the PCI Interface waits 2^{16} clocks and releases the delayed cycle internally.

When the PCI Interface detects an error, or samples the PCI_PERRn and/or PCI_SERRn lines as being active, it sets the appropriate register bits. The register bits used are the status bits defined in the *PCI Local Bus Specification (Revision 2.3)*.

When the PCI Interface detects asserts PCI_SERRn, it is assumed to be a fatal error. No transactions are accepted from other PCI masters after PCI_SERRn is asserted.

13.6.1 PCI Read Transactions

When a read transaction is claimed by the Tsi620 based upon the address, a Bridge ISF read request can be made. If the destination port receives any kind of error (for example, if the interface is disabled), the response packet indicates an error.

13.6.2 PCI Error Response Tables

Tables 80 to 82 describe PCI error possibilities. The *Error* column indicates the type of error handled by the PCI Interface. The *External Signals* column describes any external pins that the PCI Interface drives when it detects a specific error. The *Registers* column describes the register bits set for each of the error conditions. The *Other* column contains additional information about the error.

Posted Write Error Responses

Table 80: Posted Write Errors

Error	External Signals	Registers	Other
Target			
Address Parity	Drive PCI_SERRn if enabled	IRP_STAT[P_CSR] IRP_STAT[APE] PE_CSR[S_SERR] PE_CSR[D_PE] PE_CSR[S_TA]	Target Abort
Data Parity	Drive PERR if enabled	IRP_STAT[P_CSR] PE_CSR[D_PE] IRP_STAT[PW_DPE]	Allow the write to complete on the PCI bus. The write data is discarded within the PCI block and not forwarded to its Bridge ISF destination (see "Parity Across PCI").
Master			
Address Parity	Receive PCI_SERRn	n/a	n/a
Data Parity	Receive PERR Drive PCI_SERRn if enabled	IRP_STAT[P_CSR] IRP_STAT[PW_DPE] PE_CSR[MDP_D] PE_CSR[S_SERR]	n/a
Master Abort	Drive PCI_SERRn if enabled	IRP_STAT[P_CSR] IRP_STAT[PW_MA] PE_CSR[R_MA] PE_CSR[S_SERR]	Purge remaining write data
Target Abort	Drive PCI_SERRn if enabled	IRP_STAT[P_CSR] IRP_STAT[PW_TA] PE_CSR[R_TA] PE_CSR[S_SERR]	Purge remaining write data
Max Retry	Drive PCI_SERRn if enabled	IRP_STAT[PW_Retry] PE_CSR[S_SERR]	Purge remaining write data

Delayed Write Error Responses

Table 81: Delayed Write Errors

Error	External Signals	Registers	Other
Target			
Address Parity	Drive SERR if enabled	IRP_STAT[P_CSR] IRP_STAT[APE] PE_CSR[S_SERR] PE_CSR[D_PE] PE_CSR[S_TA]	Target Abort
Data Parity	Drive PERR if enabled	IRP_STAT[P_CSR] PE_CSR[D_PE]	Target Abort (see below)
Discard Timer	Drive SERR if enabled	IRP_STAT[DW_ND] PE_CSR[S_SERR]	Discard delayed write completion
Signal Target Abort	Target Abort	IRP_STAT[P_CSR] PE_CSR[S_TA]	Signal target abort in response to a max retry, address or data parity error in the destination.
Master			
Address Parity	Receive SERR	n/a	N/A
Data Parity	Receive PERR	IRP_STAT[P_CSR] PE_CSR[MDP_D]	N/A
Master Abort	Send Master Abort packet if enabled	IRP_STAT[P_CSR] PE_CSR[R_MA]	Return Bridge ISF Error Response to source bus
Target Abort	Send Target Abort packet	IRP_STAT[P_CSR] PE_CSR[R_TA]	Return Bridge ISF Error Response to source bus
Max Retry	Drive SERR if enabled	IRP_STAT[P_RETRY] PE_CSR[S_SERR]	Return Bridge ISF Error Response to source bus

Delayed Read Error Responses

Table 82: Delayed Read Errors

Error	External Signals	Registers	Other
Target			
Address Parity	Drive SERR if enabled	IRP_STAT[P_CSR] IRP_STAT[APE] PE_CSR[S_SERR] PE_CSR[D_PE] PE_CSR[S_TA]	Target Abort
Data Parity	Receive PERR	n/a	n/a
Discard Timer	Drive SERR if enabled	IRP_STAT[DR_ND] PE_CSR[S_SERR]	Discard delayed read completion
Signal Target Abort	Target Abort	IRP_STAT[P_CSR] PE_CSR[S_TA]	Bridge ISF Error Response to a master abort, target abort, max retry or address or data parity error in the destination
Master			
Address Parity	Receive SERR	n/a	N/A
Data Parity	Drive PERR if enabled Send PERR packet if enabled	IRP_STAT[P_CSR] PE_CSR[MDP_D]	N/A
Master Abort	Send Master Abort packet if enabled	IRP_STAT[P_CSR] PE_CSR[R_MA]	Return Bridge ISF Error Response
Target Abort	Send Target Abort packet	IRP_STAT[P_CSR] PE_CSR[R_TA]	Return Bridge ISF Error Response to source bus
Max Retry	Drive SERR if enabled	IRP_STAT[P_RETRY] PE_CSR[S_SERR]	Return Bridge ISF Error Response to source bus

13.7 Interrupt Handling

The PCI Interface has an independent interrupt router that is separate from the Tsi620 Interrupt Controller. The PCI interrupt router has the following features:

- Supports 11 types of internal interrupt sources (see Table 83)
- Handles two external sources for interrupts
- Handles two external destinations for interrupts
- Handles up to four external PCI interrupts
- Supports interrupt-to-MSI conversion

Table 83: Internal Sources for PCI Interrupts — IRP_STAT Register

Interrupt Type[a]	Description
DR_ND	Delayed Read No Data. This is caused by a discard timer event on a delayed read transaction.
DW_ND	Delayed Write No Data. This is caused by a discard timer event on a delayed write transaction.
PW_MA	Posted Write Master Abort. This is caused by a master abort on a posted write transaction.
PW_TA	Posted Write Target Abort. This is caused by a target abort on a posted write transaction.
PW_RETRY	Posted Write Retry Error. This is caused by a retry timer event on a posted write transaction
PW_DPE	Posted Write Data Parity Error. This is caused by a data parity error on a posted write transaction.
APE	Address Parity Error. This is caused by an address parity error on a transaction.
P_CSR	P[ET]_CSR Register Event. This is caused by any one of the following events occurring in the CSR register: D_PE - Detected Parity Error S_SERR - Signaled PCI_SERRn R_MA - Received master abort R_TA - Received target abort S_TA - Signaled target abort MDP_D - Master Data Parity Detected
P_INT	P_INTAD Interrupt. This is caused by any of the PCI bus interrupts being asserted if configured as inputs.
HS_CSR	HS_CSR Interrupt. This is caused by either INS or EXT being asserted during an insertion or extraction, respectively.
FAB	Bridge ISF Interrupt. This is caused by an interrupt event in the Bridge ISF interrupt register.

a. The INTA_EN, INTB_EN, INTC_EN, and INTD_EN fields of the IRP_INTAD register must not be used to attempt to mask the four external PCI interrupts. Instead, use IRP_ENABLE[P_INT] to globally enable and disable all four external interrupt sources (PCI_INTAn, PCI_INTBn, PCI_INTCn, and PCI_INTDn).

Each interrupt type can be enabled by setting the corresponding enable bit in the "PCI Interrupt Enable Register". All internal types are mapped to the PCI_INTAn pin of the PCI Interface when the direction bit for the PCI_INTAn pin is set as an output.

The PCI Interface interrupt router accepts and routes interrupts to/from the same two sources and destinations:

- PCI bus interrupts
- Tsi620 Interrupt Controller interrupts

The IRP_CFG_CTL register contains two valid fields to control how interrupts are routed. The first setting INT[A-D]_TYPE[1:0] controls how the PCI interrupts, PCI_INT[A:D], are routed through the Interrupt Controller (see Table 84). The interrupt direction INT[A:D]_DIR is also configurable through the IRP_CFG_CTL register.

Table 84: INTx_TYPE Encodings

INTx_TYPE	Description[a]
00	Unused/Reserved
01	Reserved
10	Reserved
11	This setting causes the PCI_INT[A:D]n interrupts to be routed to the Interrupt Controller if the PCI interrupts are configured as inputs, or to route the Interrupt Controller outputs to PCI_INTAn if the PCI_INTAn interrupt is configured as an output (see "PCI Interrupt Control Register").

a. These settings have the desired effect only if INTx_EN is set in the "PCI Interrupt CSR", and the PCI_INTx direction is set as an input by setting the INTx_DIR bits in the "PCI Interrupt Control Register".

The second setting, LOC_INT_DEST[1:0], controls the PCI internal interrupt routing. The internal interrupt sources can be routed to three destinations (see Table 85).

Table 85: LOC_INT_DEST Register Setting Description

LOC_INT_DEST	Description
00	Routes all internal interrupt sources to the Interrupt Controller.
01	Routes all internal interrupt sources to the PCI_INTAn pin of the PCI Interface if PCI_INTAn is configured as an output.
10	Routes all internal interrupt sources to an MSI event.
11	Reserved

13.7.1 MSI Handling

The PCI Interface supports Message Signaled Interrupts (MSIs). Incoming MSIs are handled as posted write transactions. The external device must ensure that MSIs are directed to the appropriate destination in the Tsi620 by providing the correct address.

Outgoing MSIs can be generated only for internal PCI events.

13.8 Clock Generation

This section describes the operation of PCI Clock Generation when the PLLs are engaged. For a complete description of the operation of the PLLs, refer to “Clocks” on page 423.

The PCI block has five clock outputs associated with it. These clock outputs are controlled by an SSPLL, whose input reference frequency is the RapidIO reference clock. The SSPLL can generate 33.33 and 66.67 MHz, and drives this single frequency on all five clock outputs. One clock output is then connected back to the PCI clock input of the Tsi620 with the remainder being available for use by other devices on the PCI bus.

The Clock Generator PLLs and dividers provide flexible options for generating clock outputs. It uses the 125 or 156.25 MHz RapidIO reference clock input and provides output clock signals for the PCI Interface.

13.8.1 PLLs

The Tsi620’s PLLs have a range of frequencies that can be programmed by power-up pins. Table 86 shows the different PLLs, how they can be bypassed, and which registers capture the value of the PLLs from the power-up pins.

Table 86: Clock PLLs

PLL	Nominal Reference Frequency	Bypass Control	Frequency	Registers
SSPLL1	125 or 156.25 MHz	• Set PCI_RSTDIR == 0, or • Power down the Tsi620 Bridge (for information, see “Bridge Shutdown”). or • Set the PWRDWN bit to 1 in the “Clock Generator PLL0 Control Register 0”.	• SP_CLK_SEL	• PWDN_X4 for Port 8 “RapidIO SMAC x Digital Loopback and Clock Selection Register” and PWDN_X4 for “SREP Digital Loopback and Clock Selection Register”, OR • PWRDWN field of “Clock Generator PLL0 Control Register 0”
PCI PLL	25–66 MHz	PCI_PLL_BYPASS	• PCI_M66EN	N/A

The PLLs provide clock generation and timing functionality. As indicated in Table 86, SSPLL1 generates the PCI clock outputs.

13.8.1.1 PLL Bypassing

The Tsi620 provides PCI PLL bypass mode for testing and debugging. PLL bypassing is selected using the power-up option, PCI_PLL_BYPASS (see "Power-up").

- PCI de-skew PLL – PCI_CLK input is used as the internal PCI clock when PCI PLL bypass mode is selected. At PCI bus speeds of 10 MHz and less, the PCI_PLL_BYPASS can be asserted. At PCI bus speeds above 10 MHz, the PCI_PLL_BYPASS must not be asserted.

13.8.1.2 PCI Clock Outputs

The PCI output clocks are provided as a convenience to system designers. The Clock Generator (CG) uses S_CLK_p/n as the reference clock. The dividers that generate 33 and 66 MHz are selected by the PCI_M66EN and SP_CLK_SEL signals. Neither the host/agent status of the Tsi620, nor the PCI reset initialization pattern, is considered when determining the output clock frequency.

Table 87: PCI_CAP Encoding of Clock Frequency

PCI_M66EN	PCI_CLKO Output Frequency[a]
0	33.3 MHz
1	66.7 MHz

a. For the complete list of output frequencies, see Table 112.

There is no internal path from the Clock Generator outputs to the PCI PLL; one of the five PCI_CLKO PCI output clock pins must be connected to the PCI_CLK input of the Tsi620 if the PCI output clocks are used for system timing.

If the PCI reset is configured as an input, the Tsi620 cannot generate the PCI clocks if the Clock Generator is used because the internal PLLs require a reset. Since the PCI reset resets the Clock Generator PLL, the clocks generated by that PLL are not available when PCI_RSTn is de-asserted.

If the Clock Generator is bypassed, the CG reference clock input can generate 66-MHz or 33-MHz output clocks.

Unused PCI_CLKO outputs can be disabled by software through the "Clock Generator Output Control Register".

13.9 Reset Options

The PCI Interface can drive or sample the active-low PCI_RSTn signal (see "PCI Signals"). The direction of the PCI_RSTn signal is controlled by the state of the PCI_RSTDIR signal. When PCI_RSTn is an output, the PCI block directs the reset from the central reset controller to the PCI bus. When PCI_RSTn is an input, the PCI Interface forwards the reset from the PCI bus to the central reset controller. In both cases, the PCI Interface is reset by the reset it receives from the central controller.

The response to the assertion of reset is asynchronous: it does not require the presence of a clock. The PCI Interface synchronizes the negation of reset. However, since the reset release is synchronous, a clock must be present at the PCI_CLK input.

13.10 Initialization

The PCI Interface operates in 32-bit mode. Depending on the PCI_M66EN pin value, the bus operates in 25-33 MHz or 50-66 MHz clock speeds (see Table 88).

The PCI_RSTDIR pin determines the direction of the initialization:

- 0 = PCI detects initialization – The PCI Interface detects if its PCI Interface is configured as a PCI agent by latching the PCI initialization pattern. The port latches the initialization pattern on the rising edge of PCI_RSTn.
- 1 = PCI drives initialization – The PCI Interface drives its PCI Interface to indicate configuration as a PCI bus. The PCI Interface drives the initialization pattern on the rising edge of PCI_RSTn.

Table 88: PCI Initialization Pattern

PCI_DEVSELn	PCI_STOPn	PCI_TRDYn	Mode	Minimum Frequency (MHz)	Maximum Frequency (MHz)
De-asserted	De-asserted	De-asserted	PCI_M66EN = 0[a]	25	33
De-asserted	De-asserted	De-asserted	PCI_M66EN = 1	33	66

a. PCI_M66EN indicates the operating frequency of the PCI Interface. The value of PCI_M66EN is valid on the rising edge of PCI_RSTn.

13.11 Power Management

The PCI Interface supports features that meet the *PCI Bus Power Management Interface Specification (Revision 1.1)*. This capability enables an operating system to control the hardware that implements power saving features independently of the platform.

The PCI Interface supports the minimum requirements that are mandated by the *PCI Bus Power Management Interface Specification (Revision 1.1)*. These requirements include the following power states: D0, D3hot, and D3cold. Data registers are not supported.

The PCI Interface does not assert PCI_PMEn when changing between different power states. The PCI_PMEn output is not connected directly to the power management states. The PCI_PMEn signal functions as an additional interrupt output. If enabled by setting PCI control bit PMCS[PME_EN] (see "PCI Power Management Control and Status Register"), the PCI_PMEn output is asserted by the INT0 output from the internal Interrupt Controller.

PWR_ST in the "PCI Power Management Control and Status Register" shows the power state of the PCI Interface. The power state can be changed by writing a valid value to PWR_ST: either 00 for D0, or 11 for D3hot. The PCI Interface is in the D3cold power state when power is removed, and only a power reset can change the state from the D3cold to the D0 power state. The following table shows the events that cause a transition between power states.

Table 89: Events Causing Transition between Bus Power Management States

Current State	Event	New Power State
D0	Write 11 to P_PMCS[PWR_ST];	$D3_{hot}$
$D3_{hot}$	Remove power from the Tsi620	$D3_{cold}$
$D3_{hot}$	Write 00 to P_PMCS[PWR_ST]	D0
$D3_{cold}$	Power-on reset	D0

When a power state transition occurs (for example, from D0 to $D3_{hot}$ and from $D3_{hot}$ to D0) in an external PCI device, and it asserts PCI_PMEn, IRP_INTAD[PME] is set to indicate the assertion of PCI_PMEn (see "PCI Interrupt CSR").

13.12 Endian Modes

The PCI Bridge ISF supports byte swapping, word swapping within 32-bit boundaries, and byte swapping within 32-bit boundaries if both byte and word swapping is enabled. For information on enabling swapping, see "PFAB Control and Status Register".

13.12.1 Byte/Word Swap Impact on Internal Register Access

All register accesses within the PCI Interface are completed through an internal multi-master peripheral bus. This bus accesses local configuration space registers and provides access to other configuration registers inside the Tsi620. This bus accesses configuration registers whether the configuration transaction is a PCI configuration transaction, a PCI read or write to memory-mapped registers, or a Bridge ISF configuration transaction.

All configuration registers within the PCI configuration space and extended space are assumed to be in the same byte/word format as the PCI bus. As a result, byte/word swapping is not performed when accessing these registers from the PCI bus. All other registers attached to the internal peripheral bus, including Bridge ISF registers and registers within other Tsi620 ports, are assumed to be in internal address space. To avoid conflicts with assumptions made with other Tsi620 ports, the PCI Interface uses a byte swap bit called CFG_BSWAP in the "PFAB Control and Status Register". When this bit is set, any access to other Tsi620 ports' register space through the PCI bus is byte swapped. If the register byte swap bit is clear all internal register accesses are passed to the PCI bus as received from the peripheral bus.

All PCI and Bridge ISF configuration registers inside the PCI Interface can also be accessed either by another Tsi620 port, which is a master on the peripheral bus, or by a configuration transaction through the Bridge ISF. The same configuration options apply. All accesses to PCI configuration registers from an outside source (Bridge ISF or external peripheral bus master) are byte swapped if the register byte swap bit is set; otherwise, all accesses to all registers are passed out with no modification to the peripheral bus. If the configuration access is performed through the Bridge ISF, the register read data is placed on the 64-bit Bridge ISF word, as described for byte/word mode (address 00 to the left). The Bridge ISF configuration transactions are also passed through the byte/word mode conversion engine and the conversion specified is performed.

13.13 Vital Product Data

Vital Product Data (VPD) is defined in the *PCI Local Bus Specification (Revision 2.3)* as information that uniquely defines the following system elements: hardware, software, and microcode. VPD also provides a function for storing information such as performance data on a device. VPD resides in a local storage device, such as an EEPROM or other memory device, which is connected through RapidIO or PCI to the Tsi620. The location of the storage is specified by the Bridge ISF port and address set by P_VPD_CSR.

The PCI Interface supports VPD through a series of registers:

- VPD_EN in "PCI Miscellaneous Control and Status Register" – This bit enables and disables VPD operations. Unless VPD is enabled, it cannot be read or written.
- "PCI VPD Control and Status Register" – This register specifies the Bridge ISF port and the upper 20 bits of address to be used by that port to access the VPD.
- VPDA in "PCI Vital Product Data Capability Register" – This field provides the lower 8 bits of address to be used by the destination Bridge ISF port. It also specifies the direction of the access: read or write and access completion status. Writes to this register initiate VPD accesses.
- "PCI Vital Product Data Register" – This register contains the returned 4 bytes of data on read accesses and the 4 bytes of data to be written on write accesses.

The address composed for the Bridge ISF is formatted as described in Table 90.

Table 90: VPD Address Composition

Bits[63:28]	Bits[27:8]	Bits[7:2]	Bits[1:0]
000...000	P_VPD_CSR[VPD_OFFSET[19:0]]	P_VPDC[VPDA[7:2]]	00

All accesses to the VPD are managed through the registers in PCI configuration space.

13.13.1 Reading VPD Data

A single VPD read access reads four consecutive bytes starting from the VPD address from the VPD register. The VPD address should be Dword-aligned, as per the *PCI Local Bus Specification (Revision 2.3)*.

On a read access, the VPD Address and the VPD Flag bit are written with the Flag bit set to 0 to indicate a VPD read access (see "PCI Vital Product Data Capability Register"). The PCI Interface sets the Flag bit to 1 after it reads the 4 bytes. The Flag bit should be polled to determine when the read is complete. Byte 0 (bits 7 - 0) of the P_VPDD register contains the data referenced by the VPD Address; bytes 1, 2 and 3 contain the successive bytes (see "PCI Vital Product Data Register"). If the P_VPDD register is written to, prior to the flag bit being set to one, the results of the read operation are unpredictable.

13.13.2 Writing VPD Data

Similar to the read, the write operation writes four consecutive bytes starting from the VPD Address. The VPD address should be Dword-aligned, as per the *PCI Local Bus Specification (Revision 2.3)*.

The P_VPDD data register is written with the 4 bytes of data (see "PCI Vital Product Data Register"). Byte 0 (register bits 7- 0) contains the data to be written to the location referenced by the VPD Address; bytes 1, 2, and 3 contain the data for the successive bytes. The VPD Address and Flag should then be written with the Flag bit set to 1 to indicate a VPD write (see "PCI Vital Product Data Capability Register"). The Flag bit should be polled to determine when the write is complete. The PCI Interface sets the Flag bit is set to 0 when the write is completed.

The *PCI Local Bus Specification (Revision 2.3)* specifies a VPD data structure in which some fields are read only. The Tsi620 does not provide any write protection for these fields.

Writes must be targeted at devices connected to interfaces that are connected to the PCI Interface through the Bridge ISF. The targeted interface must be able to accept simple write commands.

13.14 Slot Numbering

Slot Numbering is a PCI Extended Capability that identifies a device that provides external expansion capabilities. PCI bus numbers throughout the system can change with the insertion or removal of a PCI-based bridging device anywhere in the system. This means that bus and device numbers are not suitable for the user to physically identify the device. The slot number is the constant physical property that identifies an expansion board. For information on how to configure this PCI extended capability, see “PCI Slot Identification Capabilities Register”.

13.15 Bus Arbitration

The Tsi620 has a PCI bus arbiter that offers dedicated support for its PCI Interface, and up to four external PCI masters. If required, the PCI bus arbiter can be disabled in a system that already has a PCI bus arbiter. In this case, the Tsi620 uses its own Request and Grant signals to arbitrate access to the PCI bus (see “PCI Signals”).

If an external bus arbiter is used, the PCI Interface uses PCI_REQn[1] and PCI_GNTn[1] to acquire access to the PCI bus.

The PCI bus arbiter uses a fairness algorithm to prevent deadlock, and supports bus parking on the last master or a specific master. These bus arbitration features are discussed in the following sections.

13.15.1 Fairness Algorithm

The PCI bus arbiter implements a *fairness algorithm* to prevent deadlock on its PCI bus (see Figure 34). PCI devices that are able to master the bus are assigned one of two priority levels: level 0 or level 1 (see Mx_PRI in “PCI Arbiter Control and Status Register”). Devices assigned to level 0 are of a lower priority than devices assigned to level 1. Devices assigned to a same level have equal priority in the arbitration process.

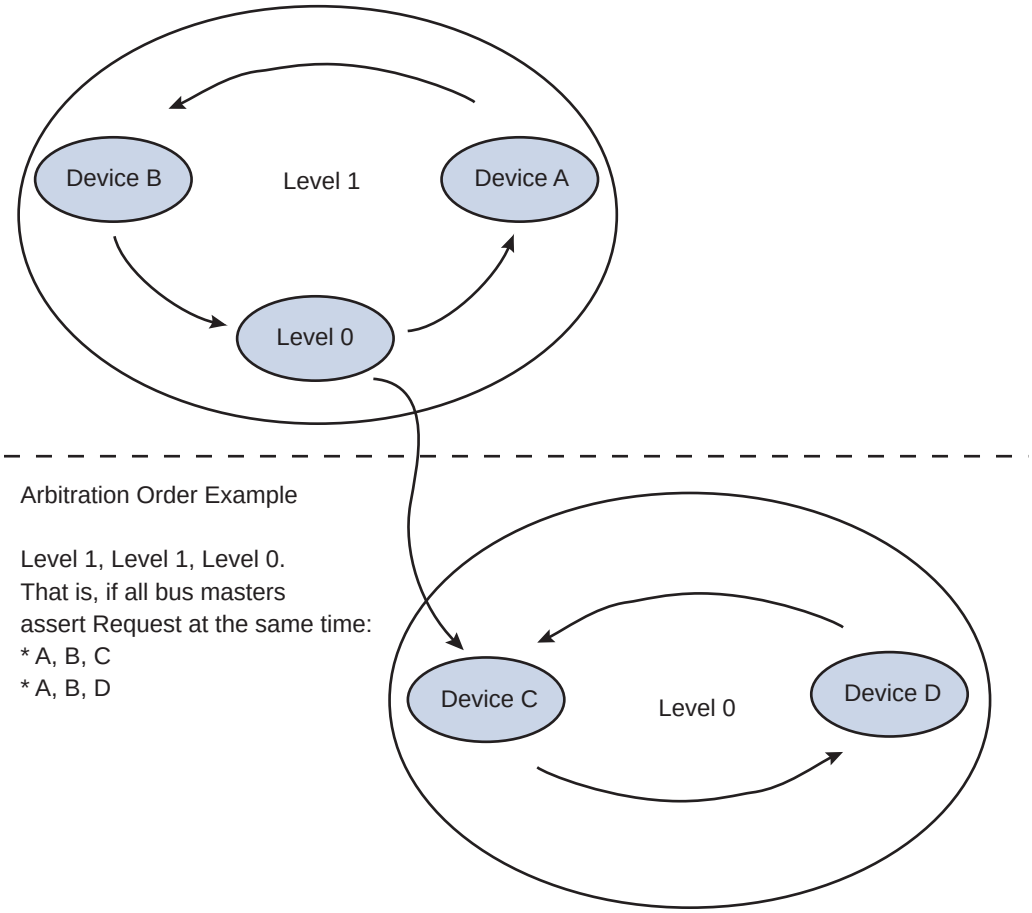
In the figure, it is assumed that all four PCI bus masters are requesting the bus at the same time. Since devices A and B are assigned a level 1 priority they are granted access to the bus first. Once device B is granted access, the fairness algorithm processes one of the bus requests from devices assigned level 0 (in this example, device C). After device C is granted bus master, the algorithm checks the level 1 priority devices *before* granting device D access to the PCI bus. If a level 1 priority device is requesting the bus then it is granted access before device D can become bus master. This process continues as long as the fairness algorithm is used.

Fairness is defined by the *PCI Local Bus Specification (Revision 2.3)* as a algorithm that grants PCI masters access to the PCI bus, independent of other requests.

Arbitration is performed among master devices asserting PCI_REQn to the bus arbiter. PCI master devices that do not drive PCI_FRAMEn on the first PCI positive clock edge after PCI_GNTn is asserted, risk losing their turn to access the PCI bus.

Bus arbitration is hidden, which means it occurs during the previous access so that no PCI cycles are consumed due to arbitration, except when the bus is in an idle state.

Figure 34: PCI Bus Arbiter — Fairness Algorithm



13.15.1.1 PCI Master Requesting the PCI Bus

When the bus is idle, a master that is requesting the bus has a programmable number of clocks from the detection of PCI_GNTn asserted to drive PCI_FRAMEn asserted. By default, the limit is set to infinite. If the programmed clock limit is exceeded, the arbiter assumes the bus master is unable to drive the bus and re-arbitrates the bus to another requesting master. PCI masters unable to assert PCI_FRAMEn within the programmed number of clocks of detecting PCI_GNTn asserted lose their turn to access the PCI bus. These devices are handled as “broken” as indicated in the PCI specification.

The PCI bus is idle when both PCI_FRAMEn and PCI_IRDYn are negated.

13.15.1.2 PCI Master Driving the PCI Bus

A PCI master accessing the PCI bus has extended assertion of PCI_GNTn by the arbiter if no other master is attempting to access the bus. The arbiter keeps PCI_GNTn asserted for the PCI master actively driving the bus. This enables the PCI master to extend its bus access beyond the Latency Timer limit (see "PCI Miscellaneous 0 Register"). The PCI_GNTn to the driving bus master remains asserted for the duration of the transaction, regardless of the state of the master's PCI_REQn signal.

The arbiter does not try to park the bus on another master while the present master is actively driving the bus with PCI_REQn negated. The arbiter negates all PCI_GNTn lines for all masters except the master accessing the PCI bus if another master asserts PCI_REQn to gain the bus. PCI_GNTn is negated for the duration of the active access. The bus arbiter updates the arbitration when it detects PCI_FRAMEn negated and PCI_IRDYn asserted, which occurs in the last data phase of the transaction. The arbitration update minimizes the latency to higher priority PCI masters that may have asserted their PCI_REQn while the present transaction was active.

13.15.2 Bus Parking

The PCI bus arbiter supports a flexible bus parking scheme using the following two methods:

- Last master
- Specific master

The bus parking mode is configured using PARK in the "PCI Arbiter Control and Status Register". Last master is the default mode. If specific master mode is selected, BM_PARK selects the specific bus master for parking. The parked master must enable its drivers for the following PCI signals:

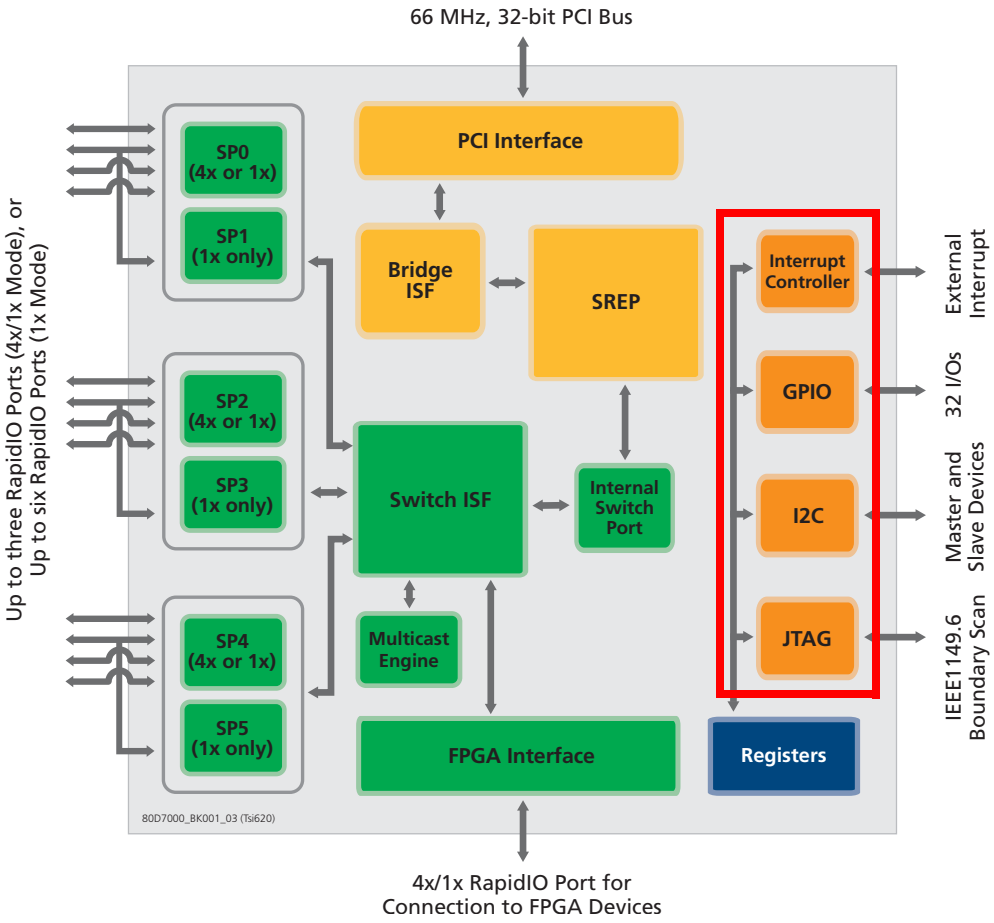
- PCI_AD[31:0]
- PCI_CBE[3:0]
- PCI_PAR

Bus parking does not occur until the PCI bus is idle. If a PCI master accesses the bus while there are no PCI_REQn signals asserted to the bus arbiter, PCI_GNTn should remain asserted to this master until the bus becomes idle. At the same time, the bus arbiter should negate this PCI_GNTn and assert grant to the parked bus master.

PART 4: SECONDARY INTERFACES

The Tsi620 contains numerous interfaces that are secondary to the device, including an Interrupt Controller, GPIO Interface, an I2C Interface, and a JTAG Interface.

The secondary interface topics that are discussed in this part of the document are highlighted in the following figure.



14. Interrupt Controller

This chapter describes the top-level event management scheme of Tsi620. Topics discussed include the following:

14.1 Overview

The Tsi620 supports two types of events:

- Reset events
- Non-reset events

Reset events notify system software to place the hardware in a safe state before a reset occurs (see "Resets"). The three sources of reset events are the RapidIO ports, SREP, and PCI. The only notification method for reset events is the RST_IRQ_b interrupt pin (see "Interrupt Signals").

Non-reset events are all other events that can occur in the Tsi620. Every Tsi620 block can detect non-reset events. Software can be notified of non-reset events using the IRQ_b interrupt pin, the PCI INTA interrupt pin, or through the use of RapidIO port-write transactions.

14.2 Reset Event Sources

As described in "Resets", reset requests from PCI, SREP, or the RapidIO ports can be handled as software interrupts instead of resetting the Tsi620 immediately. Use of an interrupt allows software to place hardware into a safe state before a reset occurs.

The only notification method for reset events is the RST_IRQ_b interrupt pin.

The RST_IRQ_b pin can be routed to a GPIO input configured as an interrupt source. The GPIO interrupt can then be routed to PCI INTA, INT_b, or to a port-write originated by SREP.

The RST_IRQ_b pin is asserted for at least 4 clock cycles before any reset is performed by the Tsi620 (see "Resets").

To handle the assertion of the RST_IRQ_b pin:

1. Read the "Block Event Status Register" to determine the source(s) of the reset event (PCI_RESET_RX, SREP_RESET_RX, SW_RESET_RX).
2. When the SW_RESET_RX bit is set, check the RCS field of the "RapidIO Port x Multicast-Event Control Symbol and Reset Control Symbol Interrupt CSR" to determine which port(s) received a reset request.
3. Perform the actions necessary to handle the reset request.
4. Perform a reset using the "Block Reset Control Register".

14.2.1 PCI Reset Request Event

The PCI reset request occurs when the PCI_RST signal is asserted, and the PCI_RSTDIR signal is 0 [(PCI_RST is an input) see "PCI Signals").

To configure the PCI Interface to assert RST_IRQ_b when PCI_RSTn is asserted, set the PCI_SELF_RST bit to 0 in the "Block Reset Control Register".

The PCI_RESET_RX field of the "Block Event Status Register" is set to 1 when a PCI reset request occurs.

14.2.2 SREP Reset Request Event

The SREP can receive a RapidIO reset request from the Internal Switch Port.

To configure the SREP to assert RST_IRQ_b when it receives a RapidIO reset request, the following register settings are necessary:

- Set SELF_RST to 0 in the "SREP Mode CSR".

The SREP_RESET_RX field of the "Block Event Status Register" is set to 1 when SREP receives a reset request.

14.2.3 Switch Port Reset Request Event

Any RapidIO port can receive a RapidIO reset request from its link partner.

RapidIO ports can be configured individually to perform a reset, or to assert RST_IRQ_b, when it receives a reset request from its link partner. To configure a RapidIO port to assert RST_IRQ_b when it receives a RapidIO reset request, set SELF_RST to 0 and RCS_INT_EN to 1 in the ports (see "RapidIO Port x Mode CSR").

To ignore reset requests on a RapidIO port, set SELF_RST to 0 and set RCS_INT_EN to 0.

The SW_RESET_RX bit of the "Block Event Status Register" is set to 1 when a RapidIO port reset request occurs.

To determine which RapidIO port received the reset request, check the RCS field of the "RapidIO Port x Multicast-Event Control Symbol and Reset Control Symbol Interrupt CSR".

14.3 Non-Reset Event Sources

This section discusses the types of non-reset event sources supported the Tsi620.

14.3.1 PCI Event

The PCI Interface can produce a single interrupt output signal to the Tsi620 Interrupt Controller. The PCI event comes from the PCI Interface's internal events, which include the PCI interrupt input signals (see "Interrupt Handling").

A value of 1 in the PCI_ERR field of the "Block Event Status Register" indicates that an internal PCI event has occurred.

The PCI Interface can route internal PCI events directly to PCI INTA, or to a PCI MSI, using the LOC_INT_DEST field of the "PCI Interrupt Control Register".

When internal PCI events are routed directly to PCI INTA or a PCI MSI, the PCI_ERR field can never become 1.

14.3.2 I^2C Event

The I^2C Interface produces a single interrupt event for the Tsi620 Interrupt Controller (see "Interrupt Handling").

14.3.3 Bridge ISF Event

The Bridge ISF produces a single interrupt event for the Tsi620 Interrupt Controller (see "Global Status Reporting"). This event is asserted if one of the TEA, ECC_CE, or ECC_UE bits is set in the "Bridge ISF Port Error Status Register". For more information on Bridge ISF event sources, see "Transaction Timeout", "Correctable Error Detection", and "Uncorrectable Error Detection".

14.3.4 Switch Event

The Tsi620 Switch produces a single interrupt output, SWITCH_ERR, which is used by the Tsi620 Interrupt Controller. This includes interrupts from the RapidIO MACs, the Multicast Engine, the Switch ISF, and the Internal Switch Port (see "Switch Interrupt Notifications").

Events detected by RapidIO ports 0 through 6 can be routed to become a port-write transaction which is originated by the RapidIO port. Events detected by RapidIO ports 0 through 6 can also be routed to become an interrupt (SWITCH_ERR), which can in turn be routed to INT_b or PCI INTA.

14.3.5 GPIO Events

The Tsi620 supports 32 GPIO signals, each one of which can be used as an interrupt input. These interrupt inputs are divided into two groups, each of which has a top-level interrupt event (see "GPIO Interface").

14.3.6 SREP Event Sources

The SREP has four separate non-reset related interrupt outputs that provide interrupt data to the Tsi620 Interrupt Controller:

- Error Event Interrupt
- Doorbell Received Interrupt
- Port-Write Received Interrupt
- Multicast Event Control Symbol Received Interrupt

For more information on these interrupts, see “SREP Event Management Support”.

14.3.7 Clock Generator Event

The Tsi620 has a clock generator that can generate an event when the Phase Locked Loops encounter an error condition (see “Clock Generator Interrupt Control Register”).

14.4 Non-Reset Event Notification

External entities can be notified of Tsi620 non-reset events using the Tsi620 interrupt pin (INT_b), the PCI Interface (PCI_INTA), or RapidIO port writes. The control of how events are routed to the various notification options is located in two registers:

- “Event Routing Register 1”
- “Event Routing Register 2”

14.4.1 Tsi620 INT_b Signal

The Tsi620 interrupt output pin, INT_b, indicates that a non-reset event has occurred. This pin is normally routed directly to a processor.

14.4.2 PCI Event Notification

Events that are routed for PCI event notification cause the assertion of the PCI INTA signal. PCI internal events can be mapped to the PCI INTA signal, or alternatively can trigger a PCI MSI write transaction (see “Interrupt Handling”).

Note that when PCI internal events are mapped directly to the PCI INTA signal or to a PCI MSI, the status of the PCI event is not available in the “Block Event Status Register”.

14.4.3 SREP Event Notification

The only function for event notification in the SREP is the transmission of a port-write packet (see "SREP Event Management Support"). The format of a port-write that is generated by the SREP for the Tsi620 is displayed in the following table.

Table 91: SREP Event Notification — Format

Data Payload Byte Offset	Bit 7	Bit 6	5Bit	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Component Tag CSR ("SREP Component Tag CSR")							
1								
2								
3								
4	Reserved (0)							
5								
6								
7								
8	Reserved (0)							
9	Reserved (0)							
10	Reserved (0)	CLK_GEN	I2C	GPIO	PCI Error	Reserved (0)	Bridge ISF Error	SREP Switch Port Error
11	Port ID (Always 0)							
12	Logical/Transport Error Detect CSR ("SREP Logical and Transport Layer Error Detect CSR")							
13								
14								
15								

14.4.4 Switch Event Notification

Each RapidIO port can be configured to send port-writes to notify a RapidIO entity that it has detected an event. For more information on Tsi620 switch event notification, see "Switch Port-Write Notifications" and "Switch Interrupt Notifications".

14.5 Event Mapping

Any event input can be mapped to any event output using the "Event Routing Register 1" and the "Event Routing Register 2".

Once an event notification is received, if the event notification could have been triggered by multiple event sources, the first step is to read the "Block Event Status Register" to determine the source of the event notification. Each bit in this register indicates which block(s) caused the event notification.

15. GPIO Interface

Topics discussed include the following:

15.1 Overview

The Tsi620 General Purpose Input/Output (GPIO) Interface is a 32-bit I/O port that supports a variety of functions, such as LED state generation, control of external devices, and application-specific interrupts. The GPIO Interface has the following features:

- Supports 32 3.3V GPIO pins
- Each pin is configurable as an input or an output
- Always able to read the current state of each signal
- Each I/O configured as an input can be used as an event-latched interrupt source for the Tsi620 Interrupt Controller
- Supports active high and active low level sensitive interrupt inputs
- Supports positive-edge and negative-edge triggered interrupts

15.2 GPIO as Inputs/Outputs

The GPIO pins are arranged in two groups of 16 pins. When the GPIO pins are used as input/output signals, they are controlled using the following registers:

- "GPIO 0 Data Register" – This register controls the data value driven by GPIO pins configured as outputs. It also gives the current value of the GPIO signal, whether the GPIO is configured to be an input or an output.
- "GPIO 0 Control Register" – This register controls whether the GPIO pin is an input or an output pin.

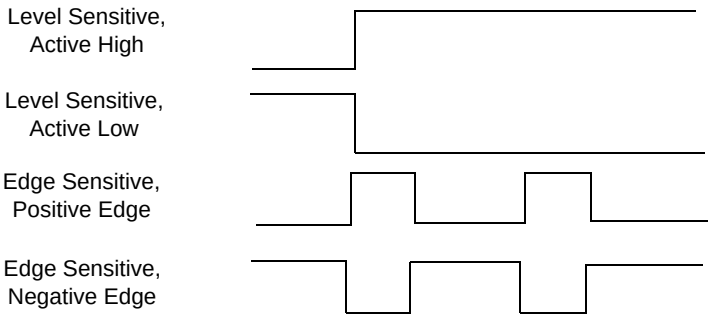
15.3 GPIO as Event Sources

A GPIO pin configured as an input can be used as an event source that can be handled by the Tsi620 Interrupt Controller. The GPIO pins support four different methods of detecting an event (see also Figure 35):

- When configured as level sensitive, active high, an event is signaled when the GPIO value is 1.
- When configured as level sensitive, active low, an event is signaled when the GPIO is 0.

- When configured as edge sensitive, positive edge, an event is signaled when the GPIO signal transitions from 0 to 1.
- When configured as edge sensitive, negative edge, an event is signaled when the GPIO signal transitions from 1 to 0.

Figure 35: GPIO Pin Interrupt Trigger Options



GPIO pins are configured as interrupts using the "GPIO 0 Interrupt Status Register" and the "GPIO 0 Edge Control Register". The Interrupt status register selects the GPIO signals that are used as events. The Edge control register controls whether a GPIO is level or edge sensitive, and what polarity of level or edge sensitivity the signal has.

The events signaled by the GPIO pins are cleared using the "GPIO 0 Interrupt Status Register".

A GPIO is asserted regardless of the state of the GPIO input pin until it is cleared in the "GPIO 0 Interrupt Status Register".

16. I²C Interface

Topics discussed include the following:

16.1 Overview

The I²C Interface provides a master and slave serial interface that can be used for the following purposes:

- Initializing device registers from an EEPROM after reset
- Reading and writing external devices on the I²C bus
- Reading and writing Tsi620's internal registers for management purposes by an external I²C master

The I²C Interface has the following features:

- Operates as a master or slave on the I²C bus
  - — Multi-master support
    - – Arbitrates among multiple masters for ownership of the I²C bus
    - – Automatically retries accesses if arbitration is lost
    - – Provides timeout indication if the Tsi620 is unable to arbitrate for the I²C bus

  - I2C Interface: Master interface
    - Supports 7-bit device addressing
    - Supports 0, 1, or 2-byte peripheral addressing
    - Supports 0, 1, 2, 3, or 4-byte data transfers
    - Reverts to slave mode if arbitration is lost
    - Supports clock stretching by an external slave to limit bus speed to less than 100 kHz
    - Handles timeouts and reports them through interrupts
  - I2C Interface: Slave interface
    - Slave address can be loaded from sources: boot load from EEPROM, or by software configuration
    - Provides read and write accesses that are 32 bits in size to all Tsi620 registers
    - Ignores General-Call accesses
    - Ignores Start-Byte protocol
    - Provides a status register for determination of Tsi620's health
    - Slave operation enabled/disabled through boot load from EEPROM, or by software configuration
- Supports I2C operations up to 100 kHz
- Provides boot-time register initialization
  - Supports 1- and 2-byte addressing of the EEPROM selected by power-up signal
  - Verifies the number of registers to be loaded is legal before loading registers
  - Supports up to 2K byte address space and up to 255 address/data pairs for register configuration in 1-byte addressing mode, or up to Kbyte address space and up to K-1 address/data pairs in 2-byte addressing mode.
  - Supports chaining to a different EEPROM and/or EEPROM address during initialization.

The I2C Interface does not support the following features:

- START Byte protocol
  - Tsi620 does not provide a START Byte in transactions it masters
  - Tsi620 does not respond to START Bytes in transactions initiated by other devices. The Tsi620 will respond to the repeated start following the start byte provided the 7-bit address provided matches the Tsi620 device address.
- CBUS compatibility
  - Tsi620 does not provide the DLEN signal
  - Tsi620 does not respond as a CBUS device when addressed with the CBUS address. The Tsi620 will interpret the CBUS address like any other 7-bit address and compare it to its device address without consideration for any other meaning.
- Fast Mode or High-Speed Mode (HS-MODE)

- Reserved 7-bit addresses should not be used as the Tsi620's 7-bit address. If a reserved address is programmed, the Tsi620 will respond to that address as though it were any other 7-bit address with no consideration of any other meaning.
- 10-bit addressing
  - Tsi620 must not have its device address programmed to the 10-bit address selection (11110XXb) in systems that use 10-bit addressing. The Tsi620 will interpret this address like any other 7-bit address and compare it to its device address without consideration for any other meaning.
- General Call. The general call address will be NACK'd and the remainder of the transaction ignored up to a subsequent Restart or Stop.

16.2 Protocol Overview

The I2C protocol is a two-wire serial interface that consists of a bidirectional, open-drain clock bus (I2C_SCLK), and a bidirectional open-drain data bus (I2C_SD). Multiple master and/or slave devices can be connected to an I2C bus. I2C data is transmitted from one device to another across the I2C_SD bus with timing referenced to the I2C_SCLK bus. With some exceptions, each bus can be driven low (to a logic 0) by any device, but is pulled high (to a logic 1) by an external resistor tied to VDD. This creates a "wired-and" configuration, where any single device can drive a bus to a logic 0, but a bus rises to a logic 1 only if no devices are driving to a logic 0, allowing the pull-up resistor to bring the bus to a logic 1 voltage.

I2C requires one device to assume the role of master during a transfer. The master generates the clock on the I2C_SCLK bus and controls the overall transfer protocol, as defined by the *I2C Specification*. One or more devices assume the role of slaves during the transfer and respond to the master by either accepting data from the I2C_SD bus, or providing data to the I2C_SD bus. The selection of a specific device to act as a slave results from a master transmitting a unique slave address as part of the I2C protocol. Only one device is normally configured with the specific slave address and is the only device to respond to the master. Other parts of the I2C protocol provide for arbitration between multiple master devices, allowing more than one master device to share the bus on a one-at-a-time basis.

This document refers to I2C signals, serial clock and serial data, by names that are defined by the device package as opposed to the *I2C Specification*. For example:

- Serial clock – Package name is I2C_SCLK, specification name is SCL.
- Serial data – Package name is I2C_SD, specification name is SDA.

16.3 Block Diagram

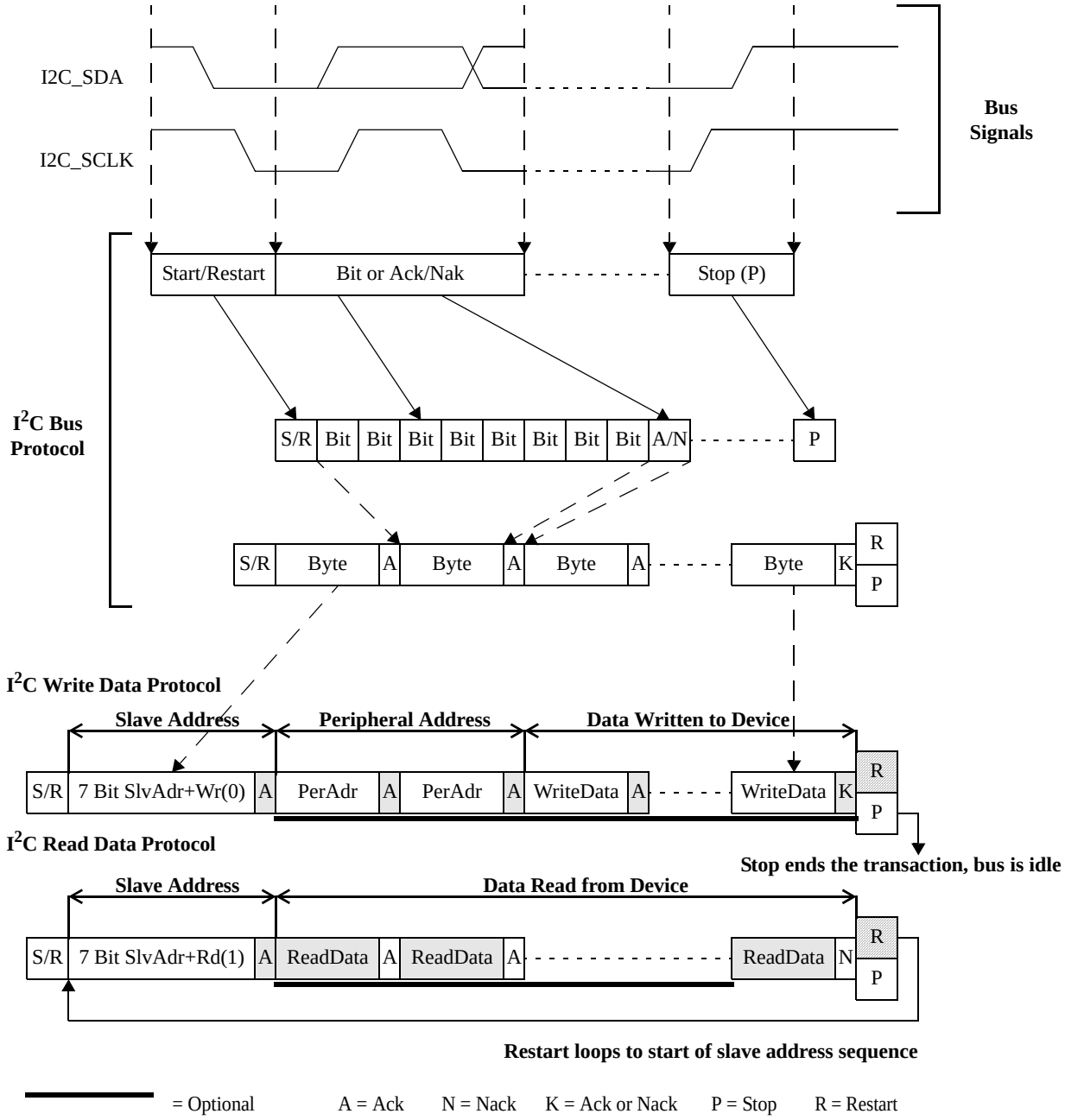
On the core side, the I2C block connects to the internal device register bus as a slave and master:

- As a slave, it enables access to the I2C block registers by a host or processor.
- As a master, it enables access to other device registers (for example, during the I2C load at power-up).

In addition, various signals relate to the boot load sequencer, an interrupt signal connects to the Tsi620 Interrupt Controller, and device-level status connects to the EXI2C_STAT register in the externally visible registers.

The reference diagram in Figure 36 shows the I2C bus and data protocol. Three basic signal relationships are defined by the bus timing: Start/Restart, Bit, and Stop.

Figure 36: I2C Reference Diagram



Note: The I2C read data protocol section of this figure implies that the peripheral addressing phase has already occurred. The I2C Interface will remember where it left off such that new reads to the same device do not require the peripheral addressing phase; however, an initial read of an I2C device will require a peripheral addressing phase.

The *start/restart* and *stop* conditions delineate a transaction – a master issues a *start* to claim ownership of the bus and a *stop* to release ownership. A *restart* is a repeated start condition between the first *start* and the terminating *stop*, and is used by a master to start a new transaction without giving up bus ownership. Between the start and stop, data is transferred one bit at a time, with the basic protocol calling for full bytes of data, this being 8 consecutive bits from master to slave or slave to master, followed by 1 more bit driven by the device receiving the data to acknowledge the receipt of the byte. The first stream of bytes following a start/restart is the *device connection sequence*, where the master places a slave address on the bus to make a connection to the device with which it needs to communicate. It is also during this period that primary arbitration for bus ownership is completed for the bus between multiple masters. If more than one master attempts to address a slave, a well-defined procedure results in all but one master backing off and waiting for a stop condition, when the bus ownership is again released.

Once a connection is made between a master and a slave, data can be transferred to or from the slave in the *data read/write sequence* phase of the transaction. This sequence is device-dependent, but a common protocol used by memory-oriented devices such as EEPROMs involves the master sending one or more bytes of *memory address* to the slave to position the slave's memory address (or peripheral address), then the master writes/reads data to/from the slave. Eventually the master ends the transaction with a *stop condition*, at which point the bus is free for other masters to start transactions.

These I2C master and slave operations are explained in the following sections.

16.4 Tsi620 as I2C Master

When the Tsi620 is an I2C master, it addresses an external slave device, generates the I2C_SCLK clock, and controls the overall transfer protocol. There are two instances where the Tsi620 is master: boot loading (see "Boot Load Sequence"), and transactions initiated by setting the START bit in the "I2C Master Control Register" (see the following section).

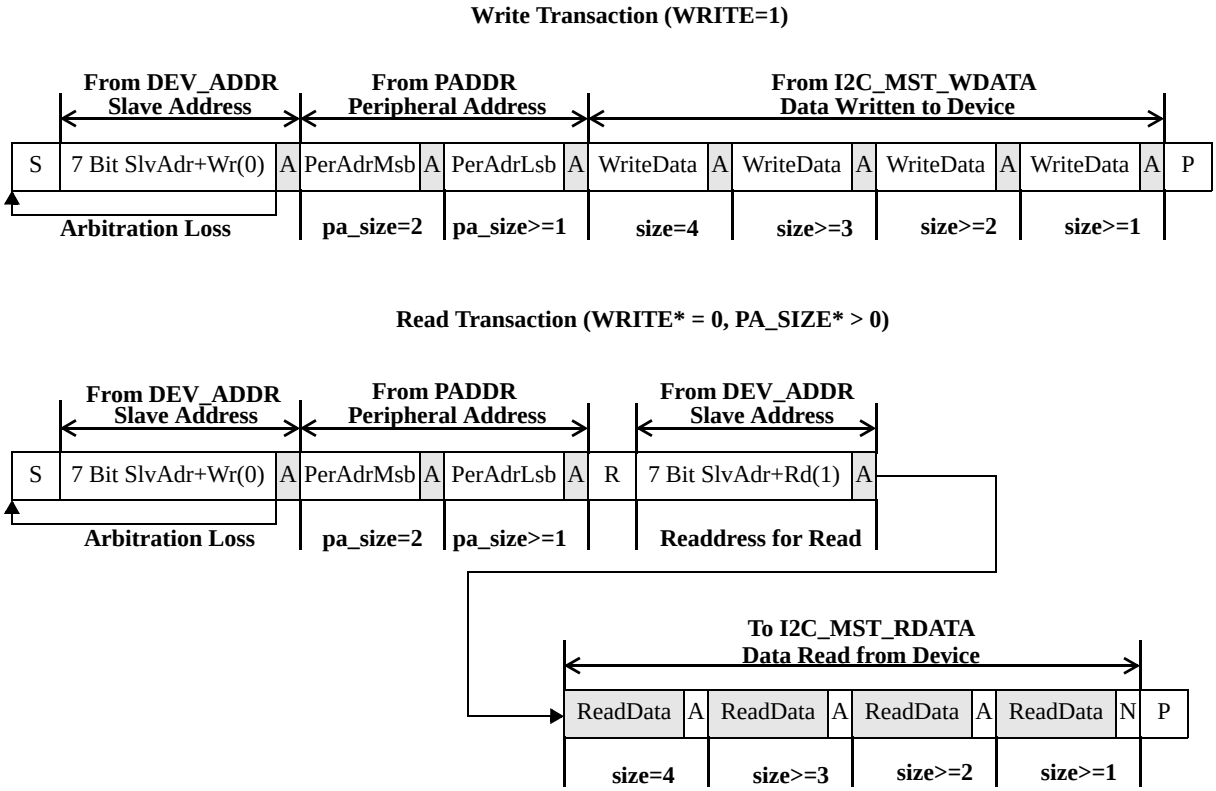
Because EEPROM devices do not have reset pins, if the Tsi620 is reset the EEPROM is unaffected and can continue to drive data at the previous state. For more information on how this issue may affect your application, and possible work around options, see "Masterless bus busy" in the *Tsi620 Design Notes*.

Software can instruct the Tsi620 to read or write to an external slave device using the following registers:

- "I2C Master Configuration Register" to configure external device parameters
- "I2C Master Control Register" to select and start the transaction
- "I2C Master Receive Data Register" for data to be read (received)
- "I2C Master Transmit Data Register" for data to be written (transmitted)
- "I2C Access Status Register" for status

Figure 37 depicts the sequences on the I2C bus when the Tsi620 is mastering a read or write transaction:

Figure 37: Software-initiated Master Transactions



Note: WRITE resides in the "I2C Master Control Register", while PA_SIZE resides in the "I2C Master Configuration Register".

The overall procedure is to configure the device through the "I2C Master Configuration Register", load the data to be written in the "I2C Master Transmit Data Register" (only needed for write operations), then load the "I2C Master Control Register" with the specific transaction information and have the START bit in that register set to 1. This initiates the master transaction. Software usually then waits for a master-related interrupt to know when the transaction has completed, or the status can be polled using the "I2C Access Status Register". If the operation was a read, the data can then be retrieved by reading the "I2C Master Receive Data Register".

Once a master operation is started, it cannot be aborted by software except when the I2C Interface is reset using the "I2C Reset Register". Reset using the "I2C Reset Register" is not recommended, however, since it can leave the I2C bus in a state that makes it difficult to ensure that all devices are back to an idle state.

The internal delay required by an EEPROM device to complete a write operation to its memory array must be managed by software. The I2C Interface only tracks the status of the operation by the bus signals.

16.4.1 Master Clock Generation

The I2C clock (I2C_SCLK) for master operation is generated by dividing down the reference clock, which is the reference clock divided by 2. The reset value for the I2C_SCLK generates a nominal 100-kHz operating speed. The bus speed is affected by external devices stretching the clock.

For master operations, the clock frequency can be changed by modifying the timing parameter registers (see "Bus Timing"). Operation above 100 kHz is possible but the Tsi620 does not implement all the standards requirements for fast mode.

16.4.2 Master Bus Arbitration

Because the Tsi620 can operate in a multi-master I2C system, it arbitrates for the I2C bus as required by the *I2C Specification*. During the Start and Slave Address phase, any unexpected state on the bus causes the Tsi620 to back off, release the bus, and wait for a Stop before retrying the transaction. If the arbitration timer configured in the "I2C_SCLK Low and Arbitration Timeout Register" expires before the device can get through the slave address phase without collision, then the transaction is aborted with the MA_ATMO status in the "I2C Interrupt Status Register". An optional interrupt can also be sent to the Interrupt Controller if enabled in the "I2C Interrupt Enable Register".

If the Tsi620 loses the arbitration for the I2C bus, and the winning master selects the Tsi620 as the target of its access, the Tsi620 responds as the slave.

16.4.3 Master External Device Addressing

A master transaction starts with a Start condition followed by the 7-bit slave address from the DEV_ADDR field of the "I2C Master Configuration Register", followed by the R/W bit. Assuming the 8 bits did not collide with another master, an ACK/NACK response bit is expected. If an ACK is received, the slave device exists and the transaction proceeds. If a NACK is received, the slave device is presumed to not exist and the transaction is aborted with a Stop and an MA_NACK status in the I2C_INT_STAT register, and an optional MA_NACK interrupt to the Interrupt Controller if enabled in MA_NACK of "I2C Interrupt Enable Register". In this case, the transaction is not automatically retried and it is up to software to retry if needed.

If the master transaction addresses the slave address of the Tsi620, the slave logic responds correctly as the target device.

16.4.4 Master Peripheral Addressing

Some devices, such as EEPROMs, require a peripheral address to be specified to set a starting position in their memory or address space for the read or write. The Tsi620 supports transactions with 0, 1, or 2 bytes of peripheral address. Because this is device dependent, the correct setting for the target device must be set in PA_SIZE of the "I2C Master Configuration Register"; otherwise, the transaction protocol is not correct.

The peripheral address is also set in the "I2C Master Configuration Register" when the transaction is started. If the transaction is a read, the Tsi620 must switch the I2C bus protocol to read mode following the peripheral address (sending the peripheral address requires write mode). To do this, a Restart condition is generated, followed by a repeat of the slave address to readdress the device in read mode. Because the bus was not released by the Restart, this phase is not subject to the arbitration timer, therefore any mismatch on the bus aborts the transaction with a MA_COL interrupt. This restart is not necessary if there is no peripheral address status in the "I2C Interrupt Status Register". An optional interrupt can also be sent to the Interrupt Controller if enabled in MA_COL of the "I2C Interrupt Enable Register".

16.4.5 Master Data Transactions

After the peripheral address phase, if any, 0 to 4 bytes of data are read or written, followed by the Stop condition. The number of bytes to be transferred is set in the SIZE field of the "I2C Master Control Register" when the operation is started, the type of transaction (read or write) is set in the WRITE field of the same register, and the order of byte transfer is set in the DORDER field of the "I2C Master Configuration Register".

For a write transaction, bytes are taken from the "I2C Master Transmit Data Register" based on DORDER and sent to the target device. Each byte must be ACKed by the device. If a NACK is received, the transaction is aborted with a Stop, an MA_NACK interrupt status in the "I2C Interrupt Status Register". An optional interrupt can also be sent to the Interrupt Controller if enabled in MA_NACK of the "I2C Interrupt Enable Register".

For a read transaction, bytes are received from the target device and placed in the "I2C Master Receive Data Register" based on DORDER. Each byte is ACKed by the Tsi620, except for the final byte that is NACKed to tell the target device to stop sending data, followed by a Stop condition to idle the bus.

Upon successful completion of a transaction, the MA_OK interrupt status is updated in the "I2C Interrupt Status Register". An optional interrupt can also be sent to the Interrupt Controller if enabled in MA_OK of the "I2C Interrupt Enable Register".

If the Tsi620 experiences a chip reset while it is writing to an EEPROM, the write does not complete and the data at the target EEPROM address may be corrupted.

16.5 Tsi620 as I2C Slave

The Tsi620 can operate as a slave device on the I2C bus. An external master device places a transaction on the bus with a device address that matches that programmed in the SLV_ADDR field of the "I2C Slave Configuration Register", or matches the fixed SMBus Alert Response address. The external master can then read or write to the Tsi620 through a small block of 256 addresses called the *Tsi620 peripheral address space*, and do the following:

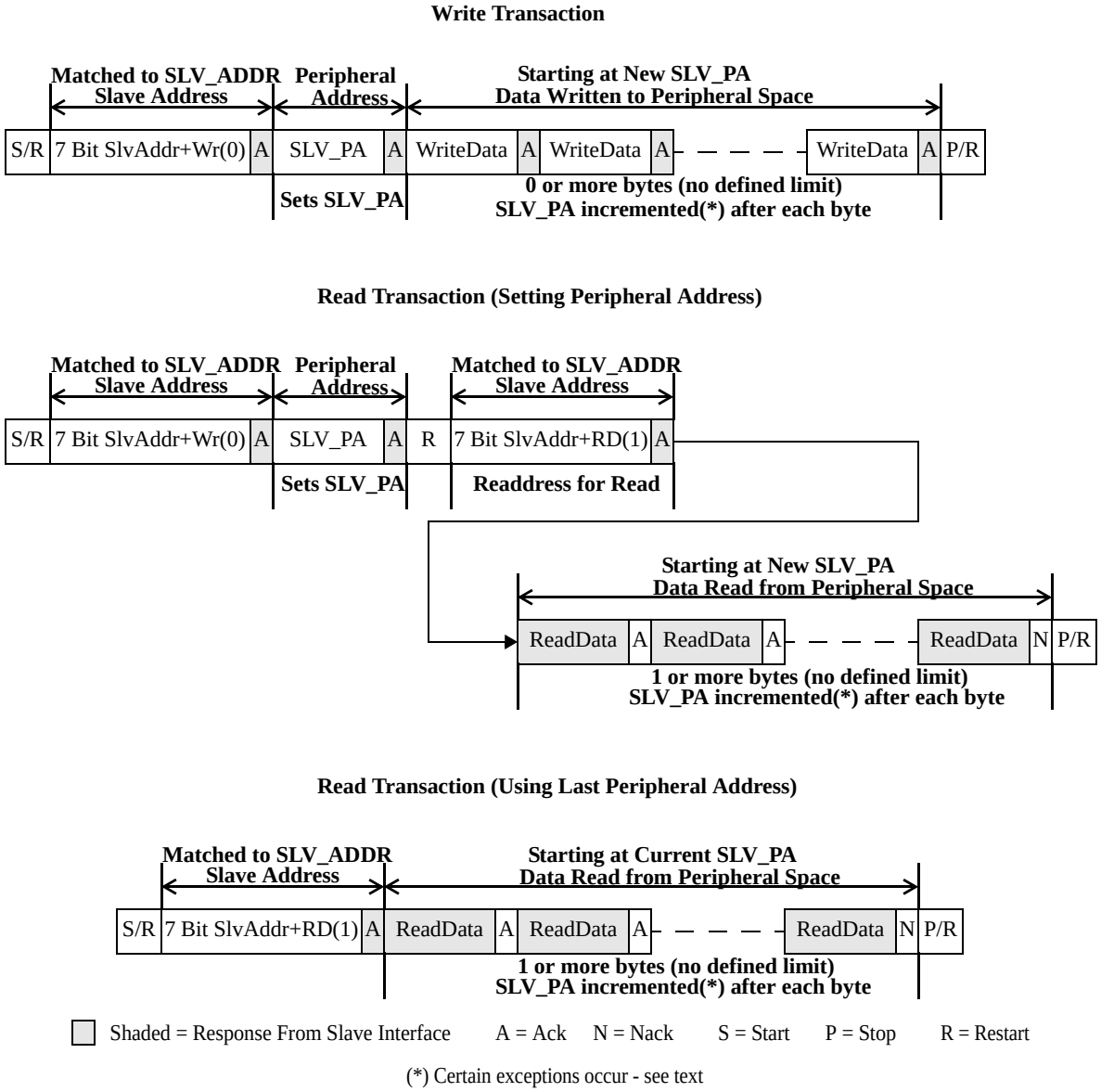
- Directly access limited status, read and write mailboxes
- Configure some options related to the slave access
- Indirectly read or write any other register in the Tsi620 that is accessible through the register bus

Figure 38 shows the bus protocols for accessing the Tsi620 as a slave device. The general procedure requires the external master to address the Tsi620, set the peripheral address to a position within the Tsi620 peripheral address space, then write or read some number of bytes. A write is terminated with a Stop or Restart, and a read is ended when the master responds to a byte with a NACK. There is no limit to the number of bytes that can be read or written in one transaction. The Tsi620 increments the peripheral address pointer after each byte, and wraps within the 256 space (see "Slave Peripheral Addressing"). Read and write transactions can be mixed by the external master issuing a Restart instead of a Stop, then sending a new transaction that addresses the Tsi620 (all writes must include the peripheral address byte).

At the completion of any slave transaction, either the SA_OK or SA_FAIL interrupt status is updated in the "I2C Interrupt Status Register". An optional interrupt can be sent to the Interrupt Controller, if enabled in SA_OK or SA_FAIL of the "I2C Interrupt Enable Register", to alert the processor/host that an external device has accessed the peripheral address space.

In addition, either the SA_WRITE or SA_READ interrupt status is updated in the "I2C Interrupt Status Register" if a read or write to the internal register space was triggered by the access. An optional interrupt can also be sent to the Interrupt Controller if enabled in SA_WRITE or SA_READ of the "I2C Interrupt Enable Register" The SA_FAIL interrupt indicates the slave transaction encountered an error. An SA_FAIL includes the slave logic detecting a collision when it was responding with data or an ACK/NACK, or one of the time-outs triggering and aborting the transaction (see "Timeouts"). If no error condition occurred, then the SA_OK interrupt is triggered.

Figure 38: Transaction Protocols for Tsi620 as Slave



16.5.1 Slave Clock Stretching

When the Tsi620 is a slave, the external master generates the I2C clock (I2C_SCLK). If the Tsi620 must access the internal register bus, I2C_SCLK is held low until data is available on a register read, or until a register write completes.

16.5.2 Slave Device Addressing

The Tsi620 supports 7-bit device addressing. The device address of the Tsi620 is set in the SLV_ADDR field of the "I2C Slave Configuration Register". For the Tsi620 to respond to an external master, the slave address on the bus must match either the address in the SLV_ADDR field, or the SMBus alert response address (see "SMBus Alert Response Protocol Support". However, an address of all zeros (0000000) never triggers a response because this address is used for the START byte and General Call protocol. Neither the START byte or General Call protocol are supported by the Tsi620.

The SLV_ADDR field can be changed at any time, either using the boot load sequence or by the processor/host. At power-up, the 7-bit device address (defined by SLV_ADDR) is loaded from the I2C_SA pin. Changing the SLV_ADDR field does not change the value of the slave address unless those bits are unlocked by first setting the SLV_UNLK field in the "I2C Slave Configuration Register".

If the Tsi620 masters an I2C transaction and the device address matches the 7-bit address programmed by SLV_ADDR, the Tsi620 will respond to its own transaction. This is the only method that allows software to write to any of the externally visible registers that are read-only from the internal register bus.

The Tsi620 only responds as a slave if the SLV_EN bit in the "I2C Slave Configuration Register" is set to 1.

The reset value of the SLV_EN bit in the "I2C Slave Configuration Register" is configured through the I2C_SLAVE pin (see "Power-up Options" in "Signal Descriptions".

16.5.3 Slave Peripheral Addressing

The Tsi620 peripheral space comprises an addressable range of 256 bytes (from 0x00 to 0xFF) that can be directly read and written by an external I2C master device. When an external master sets the peripheral address, this sets a pointer (viewable in the SLV_PA field of the "I2C Access Status Register") maintained in the Tsi620 that determines where bytes read and written by the external master are within the peripheral address space.

This 256-byte space is mapped to the Externally Visible Registers within the I2C Interface; these registers all start with EXI2C_ (see "Externally Visible I2C Internal Write Address Register"). These registers can be accessed either directly by an external master using the addresses in the peripheral address space, or by the processor/host using the internal register bus addresses. Depending on how the registers are accessed also defines their properties: for example, some registers are read-only through the peripheral address space but read/write through the internal register bus, and vice-versa.

The next section discusses the mapping between the peripheral address space and the externally visible registers.

16.5.4 External I2C Register Map

Table 92 lists the register map that is visible to external I2C devices. The lowest peripheral address maps to the LSB of the register, while the highest peripheral address maps to the MSB of the register.

The external master can set the peripheral address to any location in the 256-byte range. If that byte maps to a register, and the byte is read or written, then the specific byte within the register is read or written. These reads and writes are not through the internal address bus but are instead local to the I2C Interface. If the peripheral address does not map to a register, then a read returns 0 and a write has no effect except to increment the peripheral address.

When a byte is read or written, the peripheral address is automatically incremented to the next value, with three exceptions listed in the table (addresses 0x07, 0x17, and 0xFF).

Table 92: Externally Visible I2C Register Map

Tsi620 Peripheral Address Range	Mapped Register	Description
0x00–0x03 R/W	EXI2C_REG_WADDR	Specifies the 4-byte aligned internal register address for the register bus write access performed when EXI2C_REG_WDATA is written.
0x04–0x07 R/W	EXI2C_REG_WDATA	Specifies the data to write to the internal register address held in EXI2C_REG_WADDR. Side effects: When address 0x07 (the MSB) is written, the data in this register is written to the internal register address held in EXI2C_REG_WADDR, and the peripheral address is returned to 0x04 (the LSB). This allows consecutive internal registers to be written in one transaction without resetting the peripheral address. Note: If 0x07 is read, the peripheral address increments to 0x08; if written, the peripheral address does not increment.
0x08–0x0F Read-Only	Reserved	This range does not map to any registers.
0x10–0x13 R/W	EXI2C_REG_RADDR	Specifies the 4-byte aligned internal register address for the register bus read access performed when EXI2C_REG_RDATA is read.
0x14–0x17 Read-Only	EXI2C_REG_RDATA	Returns the data read from the internal register address held in EXI2C_REG_RADDR. Side effects: When address 0x14 (the LSB) is read, the data in this register is loaded from the internal register address held in EXI2C_REG_RADDR, before the data is returned on the I2C bus. When peripheral address 0x17 (the MSB) is read, the peripheral address is returned to 0x14 (the LSB). This allows consecutive internal registers to be read in one transaction without resetting the peripheral address. Note: If 0x17 is written, the peripheral address increments to 0x18; if written, the peripheral address does not increment.
0x18–0x1F Read-Only	Reserved	This range does not map to any registers.

Table 92: Externally Visible I2C Register Map (Continued)

Tsi620 Peripheral Address Range	Mapped Register	Description
0x20–0x23 Read-Only	EXI2C_ACC_STAT	Returns status information on accesses performed by external devices, on the incoming/outgoing mailboxes and on the state of the alert response flag.
0x24–0x27 R/W	EXI2C_ACC_CNTRL	Provides control information on how the Tsi620 handles internal register accesses through the EXI2C_REG_RDATA and EXI2C_REG_WDATA registers.
0x28–0x7F Read-Only	Reserved	This range does not map to any registers.
0x80–0x83 Read-Only	EXI2C_STAT	Returns a summary of the internal status of the Tsi620.
0x84–0x87 R/W	EXI2C_STAT_ ENABLE	Enables the bits in the status summary (EXI2C_STAT) to set the alert flag in the EXI2C_ACC_STAT register.
0x88–0x8B Read-Only	Reserved	This range does not map to any registers.
0x90–0x93 Read-Only	EXI2C_MBOX_OUT	This register allows a processor to transfer a 32-bit message to an external I2C master.
0x94–0x97 R/W	EXI2C_MBOX_IN	This register allows an external I2C master to transfer a 32-bit message to a processor.
0x98–0xFF Read-Only	Reserved	This range does not map to any registers. Side effects: When peripheral address 0xFF is read or written, the peripheral address wraps back to 0x00.

16.5.5 Slave Write Data Transactions

An external master must set the peripheral address as part of a write transaction before transferring any data. The effect of the written data depends on the register in which the peripheral address maps (see Table 92). The peripheral address is usually incremented after each byte such that consecutive bytes are written into increasing addresses within the peripheral address space. Certain exceptions exist, however, as indicated in Table 92. In addition, a write that hits the most significant byte of the "Externally Visible I2C Internal Write Address Register" (peripheral address 0x07) has the side-effect of triggering a write to a register on the Tsi620 internal register bus.

16.5.6 Slave Read Data Transactions

An external master is not required to set the peripheral address as part of a read transaction, but can do so by first writing the peripheral address and then issuing a Restart before writing any data. If not set, the read data starts wherever the peripheral address pointer was left by the previous transaction. Because it is possible that another master changed the pointer, it is recommended that the peripheral address be set at the start of a transaction. Data is returned to the external master from consecutive bytes within the peripheral address space, with certain exceptions indicated in Table 92.

There also can be side effects to reading some bytes (see Table 92 and the EXI2C register descriptions). For example, a read that hits the LSB of the "Externally Visible I2C Internal Read Address Register" (peripheral address 0x14), also triggers a read from a register on the Tsi620 internal register bus.

16.5.7 Slave Internal Register Accesses

The Tsi620 allows external masters to access all of its internal registers through the externally visible I2C registers.

The address in the register definitions refers to an offset only. The offset must be prefixed with the block address.

The "Externally Visible I2C Internal Read Data Register" is a special register in that a read operation to the first (least significant) byte of this register through the peripheral address space (0x14) triggers the Tsi620 to perform an internal register read operation using the address in the "Externally Visible I2C Internal Read Address Register". When the internal register read operation is completed, the data is first loaded into the "Externally Visible I2C Internal Read Data Register", then returned to the external I2C master byte by byte. The "Externally Visible I2C Internal Access Control Register" controls the internal register read operations and allows the user to specify when and how the register read is performed.

Likewise, the "Externally Visible I2C Internal Write Data Register" triggers the Tsi620 to perform an internal register write operation using the address in the "Externally Visible I2C Internal Write Address Register". The external device writes data to the "Externally Visible I2C Internal Write Data Register" through the peripheral space, and when the last (most significant) byte is written (0x07), the register contents is written through the internal register bus to the address in the "Externally Visible I2C Internal Write Address Register". The "Externally Visible I2C Internal Access Control Register" controls the internal register write operation and allows the user to specify when and how the register write is performed.

Internal register accesses can be prohibited by the processor/host through the RD_EN and WR_EN fields in the "I2C Slave Configuration Register".

At the completion of a slave transaction that includes a successful read or write to an internal register, a SA_WRITE or SA_READ interrupt status is updated in the I2C_INT_STAT register. An optional interrupt can also be sent to the Interrupt Controller if enabled in SA_WRITE and SA_READ of "I2C Interrupt Enable Register". This allows the processor/host to be aware that an external device is accessing the internal registers of the Tsi620.

16.5.8 Slave Access Examples

This section shows a slave internal register access by an external master. The following abbreviations are used:

<S> Start condition
<R> Restart condition
<SLVA> The 7-bit Tsi620 slave address (that matches SLV_ADDR)
<PA=#> 8-bit peripheral address (current setting viewable in SLV_PA)
<A> Acknowledge (ACK)
<N> Not acknowledge (NACK)
<P> Stop condition
<W> Write
<$\overline{W}$> Read (not write)
<WD=#> 8-bit write data (from master to Tsi620)
<RD=#> 8-bit read data (from Tsi620 to master)

Also, registers and register fields are referenced by name.

All examples assume that the transactions occur in the order given; only one master is accessing (such that nothing is changed by another master between transactions); and nothing is changed by the processor/host during the transactions.

The following conditions pre-exist: ALERT_FLAG is set in the "Externally Visible I2C Slave Access Status Register".

1. External device reads "Externally Visible I2C Slave Access Status Register" (LSB only). The returned value of 0x01 is the ALERT_FLAG. External device must NACK after the first read byte to stop the transfer.

   I2C Sequence: <S><SLVA><W><PA=0x20><A><R><SLVA><$\overline{W}$><RD=0x01><N><P>

   Following the transaction, SLV_PA is 0x21 and interrupt status SA_OK asserts. An optional interrupt can also be sent to the Interrupt Controller if enabled in SA_OK of "I2C Interrupt Enable Register".

2. External device reads "Externally Visible I2C Status Register" (all 4 bytes). Note that the data from this register is returned LSB to MSB. External device must NACK the fourth byte to stop the transfer.

   I2C Sequence:

   <S><SLVA><W><PA=0x80><A><R><SLVA><$\overline{W}$>

   <RD=0x56><A><RD=0x34><A><RD=0x12><A><RD=0x80><N><P>

   Following the transaction, SLV_PA is 0x84 and interrupt status SA_OK asserts. An optional interrupt can also be sent to the Interrupt Controller if enabled in SA_OK of "I2C Interrupt Enable Register".

3. External device does an Alert Response request. Because ALERT_FLAG is asserted, the alert response address is ACK'd and the Tsi620 slave address is returned.

   I2C Sequence: <S><0001100><$\overline{W}$><A><RD=SLVA+0><N><P>

   Following the transaction, SLV_PA is 0x84 (unchanged from previous transaction) and interrupt status SA_OK asserts. An optional interrupt can also be sent to the Interrupt Controller if enabled in SA_OK of "I2C Interrupt Enable Register". The ALERT_FLAG in "Externally Visible I2C Slave Access Status Register" is cleared because of the successful response.

4. External device writes all 4 bytes of the "Externally Visible I2C Enable Register" to 0, reads "Externally Visible I2C Slave Access Status Register", then does another Alert Response request. The ALERT_FLAG is zero (all enables were cleared), so the alert response address is NACKed.

   I2C Sequence:

   <S><SLVA><W><PA=0x84><A>

   <WD=0x00><A><WD=0x00><A><WD=0x00><A><WD=0x00><A>

   <R><SLVA><W><PA=0x20><A><R><SLVA><$\overline{W}$><RD=0x00><N>

   <R><0001100><$\overline{W}$><N><P>

   Following the transaction, SLV_PA is 0x21, "Externally Visible I2C Enable Register" is 0x00000000 and interrupt status SA_OK asserts. An optional interrupt can also be sent to the Interrupt Controller if enabled in SA_OK of "I2C Interrupt Enable Register".

5. External device writes "Externally Visible I2C Internal Access Control Register" to enable internal register auto-incrementing.

   I2C Sequence: <S><SLVA><W><PA=0x24><A><WD=0xAC><A><P>

   Following the transaction, SLV_PA is 0x25, "Externally Visible I2C Internal Access Control Register" is 0x000000AC and interrupt status SA_OK asserts. An optional interrupt can also be sent to the Interrupt Controller if enabled in SA_OK of "I2C Interrupt Enable Register".

6. External device sets up "I2C_SCLK Low and Arbitration Timeout Register" address (0x354) in EXI2C_REG_WADDR, then writes three registers back-to-back with 0x11223344, 0x55667788, and 0x99AABBCC. Because of the register auto-increment, and because the PA auto-wraps from 0x07 to 0x04, the writes can be completed in a stream. Note that data is written from LSB to MSB.

   I2C Sequence:

   <S><SLVA><W><PA=0x00><A>

   <0x44><A><0x33><A><0x22><A><0x11><A>

   <0x88><A><0x77><A><0x66><A><0x55><A>

   <0xCC><A><0xBB><A><0xAA><A><0x99><A><P>

   Following the transaction, SLV_PA is 0x04, "Externally Visible I2C Internal Write Address Register" is 0x0001_D360, "I2C_SCLK Low and Arbitration Timeout Register" is 0x11223344, "I2C Byte/Transaction Timeout Register" is 0x55667788, and "I2C Boot and Diagnostic Timer" I2C_BOOT_DIAG_TIMER is 0x8000BBCC (reserved fields stay zero) and interrupt status SA_OK and SA_WRITE assert. An optional interrupt can also be sent to the Interrupt Controller if enabled in the "I2C Interrupt Enable Register".

7. External device sets up "I2C_SCLK Low and Arbitration Timeout Register" address (0x354) in "Externally Visible I2C Internal Read Address Register", then reads 3 registers back-to-back. Because of the register auto-increment, and because the PA auto-wraps from 0x17 to 0x14, the reads can be completed in a stream. Note that data is read from LSB to MSB.

   I2C Sequence:

   <S><SLVA><W><PA=0x10><A>

   <S><SLAV><$\overline{W}$><A>

   <RD=0x44><A><RD=0x33><A><RD=0x22><A><RD=0x11><A>

   <RD=0x88><A><RD=0x77><A><RD=0x66><A><RD=0x55><A>

   <RD=0xCC><A><RD=0xBB><A><RD=0x00><A><RD=0x80><N><P>

   Following the transaction, SLV_PA is 0x14, "Externally Visible I2C Internal Read Address Register" is 0x0001_D360, and interrupt status SA_OK and SA_READ assert. An optional interrupt can also be sent to the Interrupt Controller if enabled in SA_OK and SA_READ of the "I2C Interrupt Enable Register".

8. External device writes "Externally Visible I2C Internal Access Control Register" to disable internal register auto-incrementing.

   I2C Sequence: <S><SLVA><W><PA=0x24><A><WD=0xA0><A><P>

   Following the transaction, SLV_PA is 0x25, "Externally Visible I2C Internal Access Control Register" is 0x000000A0, and interrupt status SA_OK asserts. An optional interrupt can also be sent to the Interrupt Controller if enabled in SA_OK of "I2C Interrupt Enable Register".

9. External device sets up "I2C_SCLK Low and Arbitration Timeout Register" address (0x354) in "Externally Visible I2C Internal Read Address Register", then reads same register 3 times. The register address no longer auto-increments, but the PA still auto-wraps from 0x17 to 0x14, so the reads can be completed in a stream. Note that data is read from LSB to MSB.

   I2C Sequence:

   <S><SLVA><W><PA=0x10><A>

   <S><SLAV><$\overline{W}$><A>

   <RD=0x44><A><RD=0x33><A><RD=0x22><A><RD=0x11><A>

   <RD=0x44><A><RD=0x33><A><RD=0x22><A><RD=0x11><A>

   <RD=0x44><A><RD=0x33><A><RD=0x22><A><RD=0x11><N><P>

   Following the transaction, SLV_PA is 0x14, "Externally Visible I2C Internal Read Address Register" is0x001D354, and interrupt status SA_OK and SA_READ assert. An optional interrupt can also be sent to the Interrupt Controller if enabled in SA_OK and SA_READ of "I2C Interrupt Enable Register".

16.5.9 Resetting the I2C Slave Interface

The I2C slave interface is reset by two conditions: chip reset or the detection of a START condition. When a reset is applied, the I2C slave interface immediately returns to the idle state. Any active transfer, to or from the Tsi620 when the reset is asserted, is interrupted. All registers are initialized by a reset.

As required by the *I2C Specification*, the Tsi620 resets its bus interface logic on receipt of a START or repeated START condition such that it anticipates receiving a device address phase, even if the START condition is not positioned according to the proper format. The I2C registers, however, are not reset.

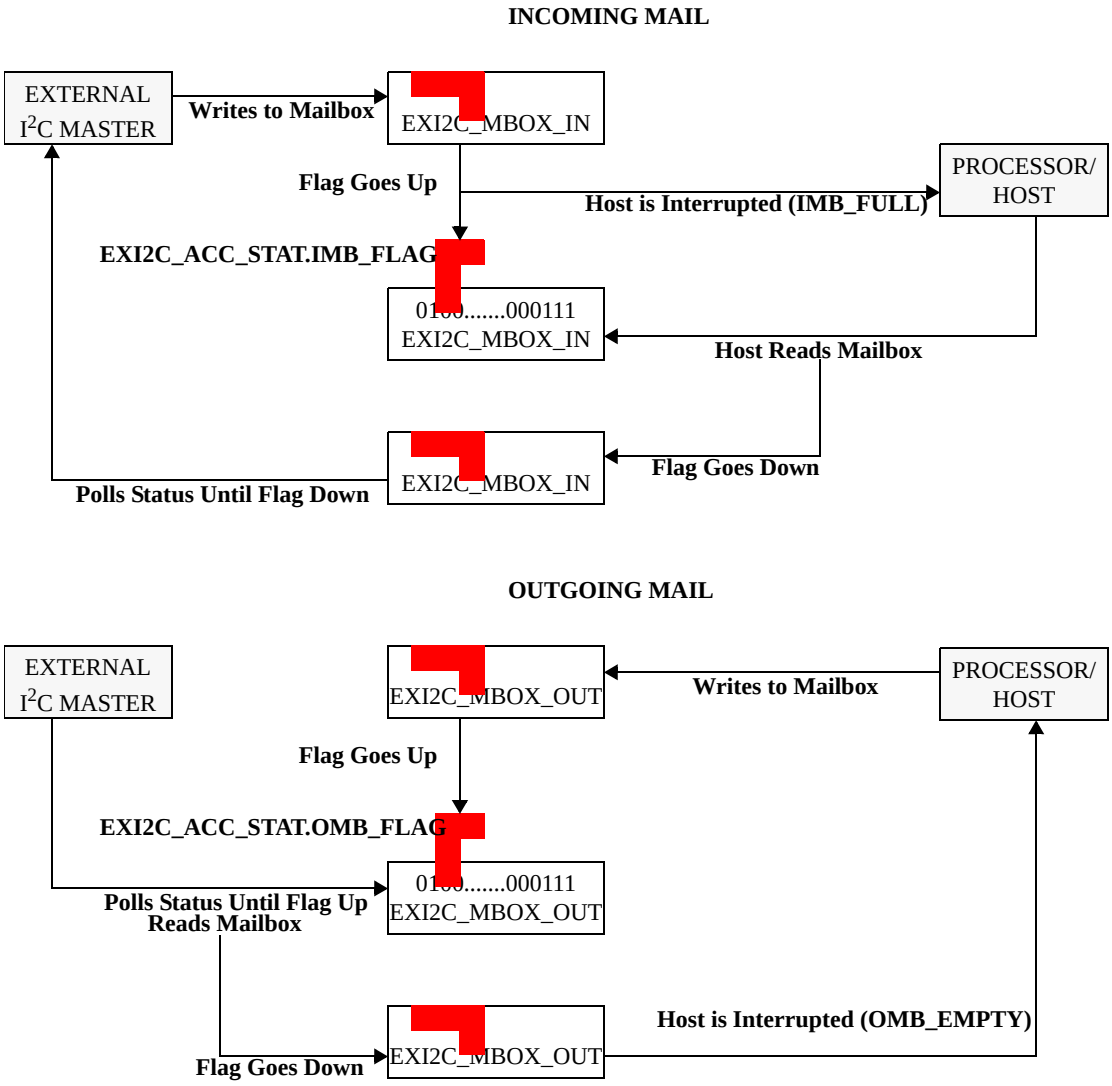
16.6 Mailboxes

As part of the peripheral address space on the Tsi620, the following registers act as I2C mailboxes for communicating information between an external I2C master and a processor or host:

- "Externally Visible I2C Incoming Mailbox Register" for data incoming from an external I2C master to the processor or host.
- "Externally Visible I2C Outgoing Mailbox Register" for data outgoing from the processor or host to an external I2C master.

In addition, flags in the "Externally Visible I2C Slave Access Status Register" are accessible by an external host to examine the mailbox status. Figure 39 shows the use of I2C mailboxes.

Figure 39: I2C Mailbox Operation



The incoming and outgoing I2C mailbox registers are discussed further in the following sections.

16.6.1 Incoming Mailbox

To send data to the processor/host, an external I2C master writes data to the incoming mailbox register, EXI2C_MBOX_IN, through the Tsi620 slave interface. When the Stop condition is seen (indicating the external master is completed writing to the mailbox), the slave interface sets the IMB_FLAG in the "Externally Visible I2C Slave Access Status Register", and the processor/host is interrupted to let it know the mailbox is full by setting IMB_FULL status in the "I2C Interrupt Status Register". An optional interrupt can be sent to the Interrupt Controller if enabled in IMB_FULL of the "I2C Interrupt Enable Register".

In response to the interrupt, the processor/host reads the incoming mailbox to retrieve the data. This process of reading the register clears the IMB_FLAG in the status register. The external I2C master can poll the status register through the slave interface, and when it sees the flag go to 0, it knows the processor/host has read the data and it is safe to write more. If the external I2C master writes more data before earlier data is read, the old data is overwritten. In this case, depending on timing, the processor/host may read a mixture of old and new data.

16.6.2 Outgoing Mailbox

To send data to an external I2C master, the processor/host writes data to the outgoing mailbox register, EXI2C_MBOX_OUT. This sets the OMB_FLAG in the "Externally Visible I2C Slave Access Status Register", which the external I2C master can poll through the slave interface. When the flag goes up (1), the external I2C master reads the outgoing mailbox register through the slave interface.

Once the Stop condition is seen (indicating the external master has completed reading the mailbox), the slave interface clears the OMB_FLAG in the status register, EXI2C_ACC_STAT, and interrupts the processor/host with an OMB_EMPTY in the "I2C Interrupt Status Register" to let it know the mailbox has been read and it is safe to write more data. An optional interrupt can be sent to the Interrupt Controller if enabled in OMB_EMPTY of "I2C Interrupt Enable Register". If the processor/host writes more data to the mailbox before the data is read, the old data is overwritten. In this case, depending on timing, the external I2C master may read a mixture of old and new data.

16.7 SMBus Support

The I2C Interface provides limited functionality for SMBus applications. The Tsi620 can act as an SMBus Host and communicate to other SMBus slave devices through a subset of the SMBus protocols (see "SMBus Protocol Support").

As a host, the Tsi620 cannot effectively receive a SMBus Host Notify message sent by another non-host SMBus device acting as a master. In addition, the Tsi620 cannot effectively act as a non-host SMBus device and receive commands from an external SMHost. Although the Tsi620 responds as a slave to the SMBus protocols, they are processed relative to the slave interface functionality. The SMBus command code is assumed to be a peripheral address, and data written to or read through the slave interface will depend on the peripheral address selected.

16.7.1 Unsupported SMBus Features

The Tsi620 does not support the following SMBus features:

- Non-host response to external SMBus host protocols, except for Alert Response Protocol
- Address Resolution Protocol (ARP) or any related commands
- SMBSUS# (suspend mode signal pin)
- SMBALERT# (alert response signal pin). External devices can signal the alert using the Tsi620's INTn or GPIO input pins. Software can then use the Alert Response Address protocol to determine the source of the alert.
- Packet Error Code (PEC). The Tsi620 does not generate, check, or expect PECs
- Process Call command

- Block write command with more than 4 data bytes
- Block read command
- Block write-block read process call command
- SMBus host notify protocol as a SMBus host device in slave mode

16.7.2 SMBus Protocol Support

The Tsi620 master interface functionality supports a subset of the SMBus Protocols (see Figure 40). In all cases, the Tsi620 masters a transaction to another SMBus device. All register and register field references are to the following I2C master interface registers:

- "I2C Master Configuration Register"
- "I2C Master Control Register"
- "I2C Master Receive Data Register"
- "I2C Master Transmit Data Register"

Use of the Quick Command with Read requires the target device to support the command. Under normal I2C protocol, the slave device returns data on the bus following the ACK, so the master could not always generate the required Stop condition. The target device must release the bus following the ACK if it is responding to this command.

Figure 40: SMBus Protocol Support

SMBus Quick Command with Write, PA_SIZE*=0, SIZE*=0, WRITE*=1

S	SlaveAdr	Wr	A	P

SMBus Quick Command with Read, PA_SIZE=0, SIZE=0, WRITE=0

S	SlaveAdr	Rd	A	P

SMBus Send Byte, PA_SIZE=1, SIZE=0, WRITE=1

S	SlaveAdr	Wr	A	PA0=Data	A	P

SMBus Receive Byte, PA_SIZE=0, SIZE=1, DORDER*=1, WRITE=0

S	SlaveAdr	Rd	A	RD0=Data	N	P

SMBus Write Byte, PA_SIZE=1, SIZE=1, DORDER=1, WRITE=1

S	SlaveAdr	Wr	A	PA0=Cmd	A	TD0	A	P

SMBus Write Word, PA_SIZE=1, SIZE=2, DORDER=1, WRITE=1

S	SlaveAdr	Wr	A	PA0=Cmd	A	TD0	A	TD1	A	P

SMBus Read Byte, PA_SIZE=1, SIZE=1, DORDER=1, WRITE=0

S	SlaveAdr	Wr	A	PA0=Cmd	A	R	SlaveAdr	Rd	A	RD0	N	P

SMBus Read Word, PA_SIZE=1, SIZE=2, DORDER=1, WRITE=0

S	SlaveAdr	Wr	A	PA0=Cmd	A	R	SlaveAdr	Rd	A	RD0	A	RD1	N	P

SMBus Write Block (NB = 1-4 bytes), PA_SIZE=2, SIZE=NB, DORDER=1, WRITE=1

S	SlaveAdr	Wr	A	PA1=Cmd	A	PA0=NB	A	TD0	A	- - - - -	TD(NB-1)	A	P

SMBus Host Notify Protocol, PA_SIZE=1, SIZE=2, DORDER=1, WRITE=1

S	SlaveAdr	Wr	A	PA0=DevAdr	A	TD0	A	TD1	A	P

SlaveAdr = SMBus device address set in DEV_ADDR
PA0 = LSB of PADDR
PA1 = MSB of PADDR
TD0 = LSB of I2C_MST_WDATA
TD3 = MSB of I2C_MST_TDATA
RD0 = LSB of I2C_MST_RDATA (data returned here)
RD3 = MSB of I2C_MST_RDATA (data returned here)

S = Start Condition
P = Stop Condition
R = Restart Condition
A = ACK
N = NACK
Unshaded = Master is driving the bus
Shaded = Slave is driving the bus

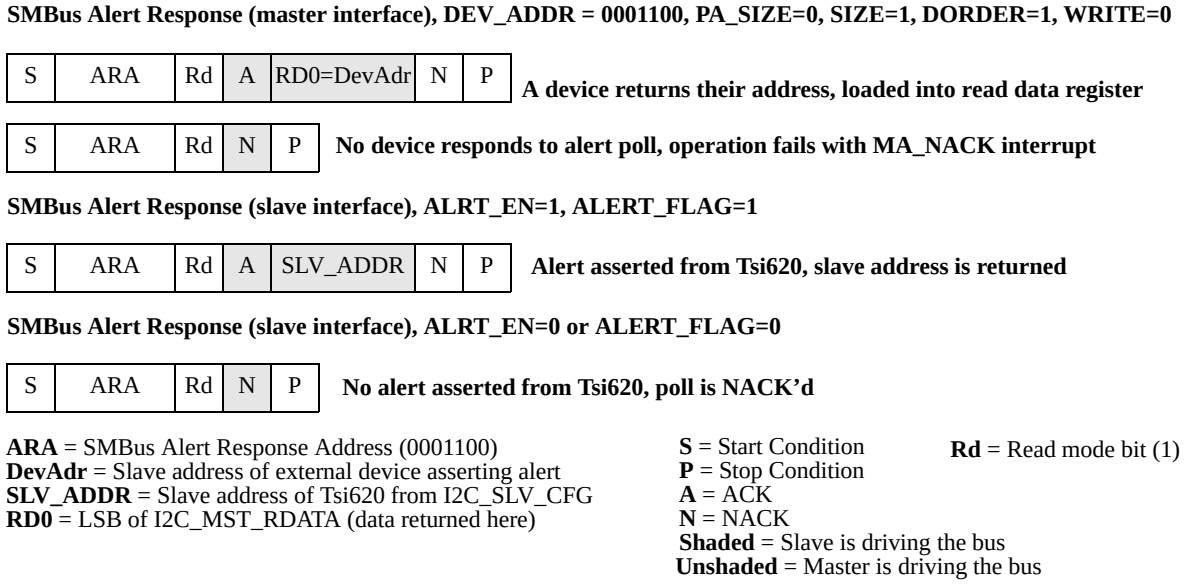
Wr = Write mode bit (0)
Rd = Read mode bit (1)

* For information about SIZE and WRITE, see "I2C Master Control Register". For information about PA_SIZE and DORDER, see "I2C Master Configuration Register".

16.7.3 SMBus Alert Response Protocol Support

The Tsi620 supports the SMBus Alert Response Protocol as either master or slave. As a master, an external device can be polled using a master read operation. As a slave, the Tsi620 slave interface responds to the Alert Response Address with the Tsi620's slave device address based on the value of ALERT_FLAG in the "Externally Visible I2C Slave Access Status Register", if enabled in ALRT_EN of "I2C Slave Configuration Register". Once the alert response is given and the Tsi620's slave device address is returned, the ALERT_FLAG is de-asserted. For the register fields indicated in Figure 40, reference the master interface registers I2C_MST_CFG, I2C_MST_CNTRL, and I2C_MST_RDATA, as well as the slave configuration register I2C_SLV_CFG.

Figure 41: SMBus Alert Response Protocol

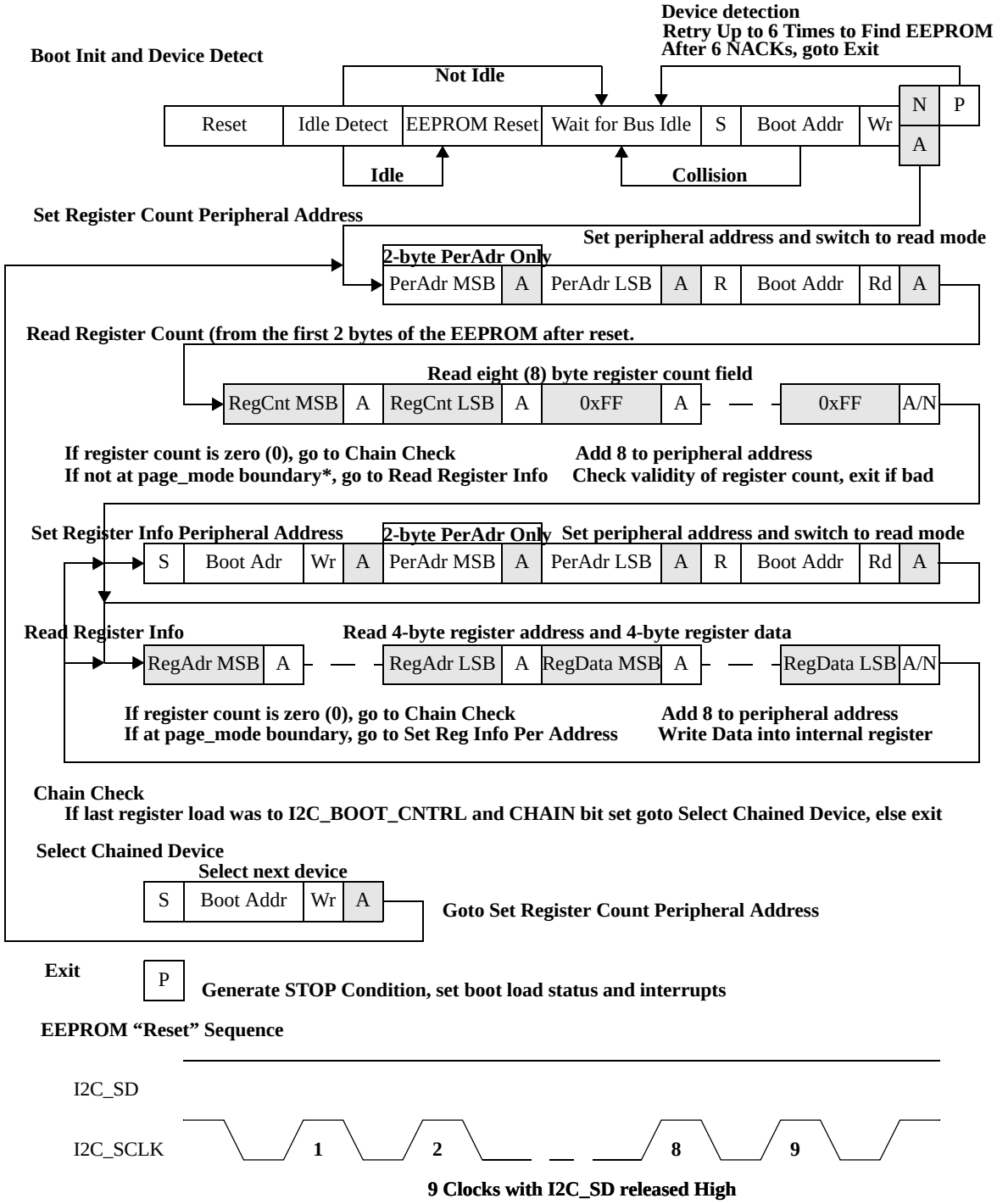


* For information about DEV_ADDR, DORDER, and PA_SIZE, see "I2C Master Configuration Register". For more information about SIZE and WRITE, see "I2C Master Control Register". For more information about ALRT_EN, see "I2C Slave Configuration Register".

16.8 Boot Load Sequence

Unless the I2C_DISABLE pin is held high, the Tsi620 will perform a post-chip reset register initialization sequence where it reads data from one or more external EEPROM devices (for more information about I2C_DISABLE, see "Power-up Options" in "Signal Descriptions". This data initializes the Tsi620's internal registers. After a block reset, a register initialization sequence is performed according to the I2C_BOOT bit in the "Block Reset Control Register". The register initialization sequence follows the steps shown in Figure 42. The register initialization sequence is controlled by the contents of the I2C_BOOT_CNTRL register.

Figure 42: Boot Load Sequence



* For information about page mode boundary, see PAGE_MODE in “I2C Boot Control Register”. For information about PA_SIZE, see “I2C Master Configuration Register”.

16.8.1 Idle Detect

Upon exit from reset, it is unknown if another master is active. The Idle Detect period determines if the I2C_SCLK signal remains high long enough (roughly 50 microseconds) that it is unlikely another master is active. If I2C_SCLK is seen low during this period, it is assumed another master is active, the EEPROM Reset phase is skipped, and boot sequence proceeds to the Wait for Bus Idle phase. This detection is performed whether or not the boot sequence is disabled. If the boot sequence is disabled, the BL_OK interrupt status is asserted immediately in the I2C_INT_STAT register, and an optional interrupt can be sent to the Interrupt Controller if enabled using BL_OK in the "I2C Interrupt Enable Register". If a master transaction is initiated before the idle detect completes, the transaction is started once the idle detect completes.

Upon exit from reset, if I2C_SDA is seen low the device assumes the bus is busy and does not attempt to reset the bus. Therefore the boot sequence will not take place.

16.8.2 EEPROM Reset Sequence

The EEPROM reset sequence is intended to cover the condition where a chip reset occurs while a transaction is active on the I^2C bus. In this case, because the Tsi620 I^2C master may have been reset and stopped generating the I2C_SCLK clock, one or more slave devices may be in a hung state where they are expecting a read or write to complete, and may be holding the I2C_SD signal low, preventing the generation of a STOP or START condition.

To try to force these devices out of their hung state, the Tsi620 allows the I2C_SD signal to stay high and generate 9 clock pulses on the I2C_SCLK signal. If no device was hung, this should not cause any problems because all devices are looking for a START condition. If a device was in the middle of a receiving a byte, the remainder of the byte will appear to have all 1s, and the device can generate an ACK or NACK. It is possible it may look to the device as if part of another byte is being sent, but because this is the master transmitting part of the protocol, the device will have released its control on the I2C_SD signal, so the master can force a START or STOP condition, even in the middle of the byte. If a device was in the middle of sending a byte, the clocks pulses will allow it to finish the transmission. The I2C_SD left high by the master (the Tsi620) will appear as a NACK to the device and it will not try to transmit another byte, but will leave the I2C_SD signal free so that another master can force a START or STOP condition.

This sequence is sent only once after a reset, and only if the Idle Detect phase was successful, and the Tsi620 believes it is not interfering with another master.

16.8.3 Wait for Bus Idle

Before attempting to access an EEPROM device, the boot loader waits for the bus to be idle. This is either the result of a successful Idle Detect phase, or, if the Idle Detect phase failed, once a STOP condition is seen on the bus, indicating another master has released control. In addition, if the I2C_SCLK and I2C_SD signals are both high for longer than the idle detect period while waiting for a STOP condition, the bus is assumed idle and the boot load process proceeds.

16.8.4 EEPROM Device Detection

Once the bus is available, the Tsi620 tries to connect to the EEPROM. A START condition is generated followed by BOOT_ADDR from the "I2C Boot Control Register". If an ACK is received, which indicates the device is present, the sequence proceeds to the next phase. If there is a bus collision, the loader returns to the Wait for Bus Idle phase because another master has the bus. If a NACK is received, the process is retried from the Wait for Bus Idle phase up to 6 times in case the device was busy. If six NACKs are received, the boot load is aborted and the BL_FAIL interrupt status is asserted. An interrupt can also be sent to the Interrupt Controller if enabled using BL_FAIL in the "I2C Interrupt Enable Register".

16.8.5 Loading Register Data from EEPROM

Once the EEPROM is successfully addressed, the Tsi620 does not release the bus until the boot load is complete. First, the peripheral address is set. The address resets to 0, so the first EEPROM accessed must be loaded from address 0. The peripheral address is either 1 or 2 bytes depending on the state of the I2C_MA pin, which much be set appropriately depending on the type of EEPROM connected.

The boot loader then switches to read mode and reads the first 8 bytes, expecting to find a count of the number of registers to be initialized in the first 2 bytes, followed by 6 bytes of 0xFF. A validity check is completed on this field — if the number of registers exceeds the maximum (see "EEPROM Data Format"), or if any of the last 6 bytes are not 0xFF, it is assumed the EEPROM does not contain boot load data, the boot load is aborted and the BL_FAIL interrupt status is updated in the "I2C Interrupt Status Register". On these boot load status bits, the optional interrupt can be forwarded to the Interrupt Controller if enabled in the "I2C Interrupt Enable Register". If the register count was 0, the boot load is ended successfully and the BL_OK interrupt status is updated. An optional interrupt can also be forwarded to the Interrupt Controller if enabled in the "I2C Interrupt Enable Register". For information on the expected EEPROM data format used for boot loading, see "EEPROM Data Format".

The boot loader continues by reading eight bytes of data for each register to be loaded, and increments the peripheral address by 8. Depending on the PAGE_MODE field in the "I2C Boot Control Register", the peripheral address is periodically reset by issuing a Restart, re-selecting the boot device, and sending the updated peripheral address. On reset, the PAGE_MODE resets to a boundary of 8 such that initially the peripheral address is updated to the device after every register is loaded (see "Accelerating Boot Load"). In addition, for 1-byte peripheral addresses, if the BINC bit is 1, then when the peripheral address crosses a 256-byte boundary (that is, when the 1-byte address rolls over to 0x00), the LSB 3 bits of the BOOT_ADDR are incremented and the device is re-addressed. This supports those EEPROMs that use the lower 3 bits of their address as a 256-byte page indicator.

For each block of 8 bytes loaded, the first 4 bytes are the register address on the internal Tsi620 register bus, and the next 4 bytes are the 32-bit data value to be written to the register. No checking is completed for register address or data validity. As soon as all 8 bytes are read, the data is written to the internal address, the peripheral address count is updated, and the register count is decremented. Once the register count reaches 0 the boot load from the current EEPROM is complete, and, unless chaining is invoked, the boot load sequence is complete, a STOP condition is issued to release the bus, and the BL_OK interrupt status is updated. An optional interrupt can also be forwarded to the Interrupt Controller if enabled in the "I2C Interrupt Enable Register".

16.8.6 Chaining

The boot loader provides for booting from multiple EEPROMs, or from multiple sections within a single EEPROM (or any combination of both). This process is called *chaining*. Chaining is invoked during the boot load sequence when three conditions occur together:

- All the registers indicated by the register count are loaded
- The final register loaded was the "I2C Boot Control Register"
- The value loaded into the I2C_BOOT_CNTRL register had the CHAIN bit set

If these conditions are met, then the boot load sequence continues using the updated information in the I2C_BOOT_CNTRL register. This allows all aspects of the boot load to be changed – the device address, the peripheral address, and so forth. When a chain occurs, the boot load sequence addresses the new device and reads a new register count from the peripheral address. This address could be non-zero, so on a chain it is possible to start loading from other than address 0 in an EEPROM.

On a chain, it is important to set the peripheral address size (PSIZE), boot address increment (BINC) and page mode (PAGE_MODE) fields so they are valid for the new EEPROM; otherwise, the boot load process may be corrupted (for information about these bits, see "I2C Boot Control Register").

It may also be necessary to use the BOOT_UNLK field to change the lower 2 bits of the EEPROM address. By default, the BOOT_UNLK field is not set, so if the BOOT_ADDR field is changed, the lower 2 bits remain at their previous value. This way the power-up reset value is not inadvertently lost. If as part of the chaining process it is necessary to change those bits (such as if the boot load is being switched to a common EEPROM), then a two-step process is needed. The I2C_BOOT_CNTRL register should be written once with the BOOT_UNLK field set to 1, then written a second time with the correct information. The lower 2 bits of the BOOT_ADDR field are only allowed to change if the BOOT_UNLK field was a 1 before the register load.

16.8.7 EEPROM Data Format

Table 93 shows the EEPROM data format for boot loading. The first 8 bytes of the EEPROM contain the number of registers to be loaded during the boot procedure. This count is the 16-bit value in EEPROM location 0 (MSB) and location 1 (LSB). The I2C Interface is limited to 255 register loads in 1-byte address mode, and limited to 8 KB-1 register loads in 2-byte address mode. The remaining 6 bytes (memory locations 2 through 7) must be set to 0xFF or the register count validity check will fail and the boot load will be aborted.

When 1-byte address mode is selected, any number of registers greater than 255 (0x00FF) aborts the boot load from the EEPROM.

When 2-byte address mode is selected, any number of registers greater than 8191 (8 KB-1 = 0x1FFF) aborts the boot load from the EEPROM.

The register load data consists of 8-byte fields aligned to 8-byte peripheral address boundaries. The first 4 bytes are the internal register address and the second 4 bytes are the register data. Note that the address and data are ordered from MSB to LSB within increasing peripheral byte addresses.

Table 93: Format for Boot Loadable EEPROM

PerAdr	PerAdr+0	PerAdr+1	PerAdr+2	PerAdr+3
0x0	RegCnt(MSB)	RegCnt(LSB)	0xFF	0xFF
0x4	0xFF	0xFF	0xFF	0xFF
0x8	RegAdr(MSB)	RegAdr	RegAdr	RegAdr(LSB)
0xC	RegData(MSB)	RegData	RegData	RegData(LSB)
0x10	RegAdr(MSB)	RegAdr	RegAdr	RegAdr(LSB)
0x14	RegData(MSB)	RegData	RegData	RegData(LSB)
...	...	...	...	...

As an example, the following shows an EEPROM configured to load two registers and then complete – first the "I2C Master Configuration Register" at internal address 0x108, loaded with data value 0x0102_0304; then the "I2C Master Transmit Data Register" at internal address 0x114, loaded with data value 0x0506_0708.

Table 94: Sample EEPROM Loading Two Registers

PerAdr	PerAdr+0	PerAdr+1	PerAdr+2	PerAdr+3	Description
0x0	0x00	0x02	0xFF	0xFF	RegCnt = 2, must have 0xFFFF at end
0x4	0xFF	0xFF	0xFF	0xFF	Must be 0xFFFF_FFFF
0x8	0x00	0x01		0x08	RegAdr = 0x108 I2C_MST_CFG
0xC	0x01	0x02	0x03	0x04	RegData = 0x0102_0304
0x10	0x00	0x01		0x14	RegAdr = 0x114 I2C_MST_TDATA
0x14	0x05	0x06	0x07	0x08	RegData = 0x0506_0708
>= 0x18	xx	xx	xx	xx	Unused by Boot

As a second example, the following shows an EEPROM configured to first load the I2C_MST_CFG register then chain to address 0x80 in the same EEPROM and load the I2C_MST_TDATA register. Note that the chain requires loading the I2C_BOOT_CNTRL register. The new peripheral address is 0x80 >> 3 = 0x10, because the 3 LSBs must be zero and are not part of the PADDR field.

Table 95: Sample EEPROM With Chaining

PerAdr	PerAdr+0	PerAdr+1	PerAdr+2	PerAdr+3	Description
0x0	0x00	0x02	0xFF	0xFF	RegCnt = 2, must have 0xFFFF at end
0x4	0xFF	0xFF	0xFF	0xFF	Must be 0xFFFF_FFFF
0x8	0x00	0x01		0x08	RegAdr = 0x108 I2C_MST_CFG
0xC	0x01	0x02	0x03	0x04	RegData = 0x0102_0304
0x10	0x00	0x01		0x40	RegAdr = 0x140 I2C_BOOT_CNTRL
0x14	0x80	0x50	0x00	0x10	RegData = 0x8050_0010 CHAIN = 1 BOOT_ADDR = 1010000 PADDR = 0x10
0x18 - 0x7F	xx	xx	xx	xx	Unused by Boot
0x80	0x00	0x01	0xFF	0xFF	RegCnt = 1, must have 0xFFFF at end
0x84	0xFF	0xFF	0xFF	0xFF	Must be 0xFFFF_FFFF
0x88	0x00	0x01		0x14	RegAdr = 0x114 I2C_MST_TDATA
0x8C	0x05	0x06	0x07	0x08	RegData = 0x0506_0708
>= 0x90	xx	xx	xx	xx	Unused by Boot

16.8.8 I2C Boot Time

The time required to perform an I2C boot depends on the following:

- The number of registers that require configuration
- The number of devices contending for EEPROM or I2C bus access
- The number of chaining operations
- The clocking speeds of the master devices

Because many of these parameters are outside the control of the Tsi620, the boot time cannot be predicted with complete accuracy.

If there are no other devices contending for bus access, a 1-byte peripheral address is used, no boot acceleration techniques are used, and no retries are necessary for device detect, then boot time can be estimated as follows:

Boot_Time =

50 *u*s idle detect time +

(9 * ClkPer) EEPROM reset time +

(102 * (RegisterCount + 1) * ClkPer) register load time +

(1 * ClkPer) STOP time

Where:

ClkPer = clock period (resets to 10 *u*s for a 100 kHz clock)

RegisterCount is the sum of number of registers from the Register Count fields in the EEPROM (only one count field unless chaining is involved).

If a 2-byte peripheral address is used, then the “102” constant increases to “111”. The 102 constant comes from the sum of Start + (9-bit boot address) + (9-bit peripheral address) + Restart + (9-bit boot address) + (9-bit data byte * 8 bytes per register = 72 bits) = 101 clocks, but the Start and Restart take an extra 1/2 clock each, so an extra clock cycle is consumed.

For example, if 255 registers are read the boot time is:

Boot_Time = 50*u*s + (90*u*s EEPROM reset) + (10*u*s * 102 * 256 register load) + 10*u*s Stop

Boot_Time = 261,270 *u*s = slightly over 1/4 second

16.8.9 Accelerating Boot Load

If boot load time is a design concern, the following techniques may accelerate the boot load sequence:

1. If the EEPROM supports reading of a large block of data sequentially, change PAGE_MODE in “I2C Boot Control Register” as the first register load. Depending on the page size, this reduces the number of times the boot load re-addresses the device and resets the peripheral address. At the limit, if the “infinite” setting was chosen and the device did not wrap on any page boundaries, the 102 constant in the boot time formula in “I2C Boot Time” would be reduced to 72 cycles per register, with only one address phase initially or per chain operation.
2. If the EEPROM supports reading at higher than 100-kHz clock speeds, the timing parameters can be changed during boot load. The success of this depends on the bus properties because the Tsi620 does not contain the Schmitt Triggers or slope controlled outputs needed to guarantee conformance to the 400-kHz high-speed mode. However, it is possible that many configurations will be interoperable at higher speeds (for information on changing timing parameters, see “Bus Timing”). Timing parameters are reloaded upon a chain operation, so the technique is to program the timing parameters for the higher speed, set up the digital filters if required, and then invoke a chain operation using the same EEPROM but the next peripheral address. Everything from the chain onwards will be mastered at the higher speed.

16.9 Error Handling

The Tsi620 handles a number of I2C errors and reports them with status bits, as summarized in Table 96.

Table 96: I2C Error Handling

Error Cause	Access Type	Tsi620 Response	Interrupt Status Bit (Events) [a]
Master Access Errors			
Master arbitration timeout expired. Tsi620 could not successfully arbitrate for the I2C bus; Arbitration lost during device addressing phase.	Master read or write initiated using I2C_MST_CNTRL register	The I2C transaction is aborted.	MA_ATMO
Tsi620 determined that it lost arbitration for the I2C bus after the device addressing phase	Read or Write	The I2C transaction is aborted.	MA_COL
No device ACK'd the slave address, or target device NACK'd a peripheral address or write data byte.	Any read or write access during slave address phase or peripheral address phase, or any write access during the data phase.	Access aborted, STOP generated. The I2C_ACC_STAT register indicates where transaction was on error.	MA_NACK
Timeout expired (I2C_SCLK Low, Byte or Transaction). Target device was too slow, or some device was interfering with the I2C_SCLK signal.	Any transfer to or from the Tsi620	Access aborted. The I2C_ACC_STAT register indicates where transaction was on error. For Byte or Transaction, master issues STOP at first legal opportunity. For I2C_SCLK Low, bus is hung, software must recover.	MA_TMO (MSCLTO, MBTTO or MTRTO)
Slave Access Errors			
Peripheral Address selects reserved external address space	Read operation	Peripheral Address byte is acknowledged, 0x00 is returned as data.	SA_OK
	Write operation	Peripheral Address byte is acknowledged. Write data is ignored.	
Peripheral Address selects a defined register, but data burst continues into reserved address	Read operation	0x00 is returned as data.	SA_OK
	Write operation	Write data is ignored	

Table 96: I2C Error Handling (Continued)

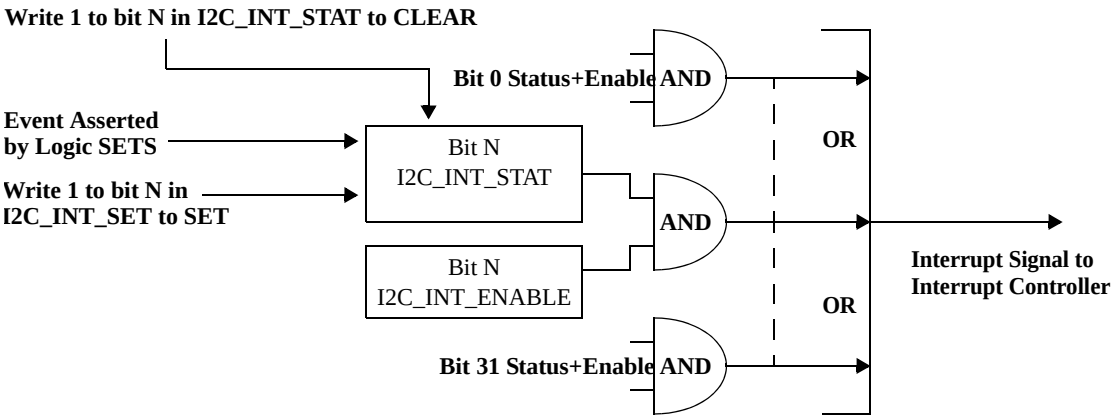
Error Cause	Access Type	Tsi620 Response	Interrupt Status Bit (Events) [a]
Programmed register address accesses a non-existent internal register block	Read operation	Read returns 0x00	SA_OK
	Write operation	Write data discarded	SA_OK
Internal register access when disabled	External master read to the EXI2C_REG_RDATA register, or write to the EXI2C_REG_WDATA register	Operation completes, returns existing RDATA or updates WDATA, but no internal register access generated.	SA_OK No SDW/SDR
Timeout expired (I2C_SCLK Low, Byte or Transaction). Target device was too slow, or some device was interfering with the I2C_SCLK signal.	Any transfer to or from the Tsi620	Slave releases I2C_SD and I2C_SCLK, goes into wait state.	SA_FAIL (SSCLTO, SBTTO or STRTO)
Protocol violation (collision detected)	Read data or Ack/Nack, when slave puts a 1 on the I2C_SD signal and another device holds the signal to 0.	Slave releases I2C_SD and I2C_SCLK, goes into wait state.	SA_FAIL (SCOL)
Register Initialization Loader Errors			
Failed to find EEPROM	Initialization read	Read operation retried up to 6 times before aborting. If not Ack'ed by the 6th try, status bits set	BL_FAIL (BLNOD)
Size field specifies more than 255 registers to load in 1-byte addressing mode, or 8 KB-1 registers in 2-byte addressing mode.	Initialization read	Initialization load aborted	BL_FAIL (BLSZ)
Register address selects non-existent register.	Register initialization write	Data discarded	None
Failed to arbitrate for I2C bus during boot load, boot load timer expired.	Initialization read	Initialization load aborted	BL_FAIL (BLTO)
Protocol error during boot load, including bytes 2-7 of a register count not containing 0xFF.	Initialization read	Initialization load aborted	BL_FAIL (BLERR)

a. To determine the setting of the interrupt status bits, see "I2C Interrupt Status Register".

16.10 Interrupt Handling

I2C interrupts are generated as shown in Figure 43. An I2C event detected by the I2C Interface sets a bit in the "I2C Interrupt Status Register" to a 1 to assert the interrupt. This bit is then anded with the corresponding bit in the "I2C Interrupt Enable Register" to determine if that interrupt is enabled. Any enabled interrupt status bit asserts the interrupt output signal to the Interrupt Controller. This signal stays asserted until all enabled bits in the interrupt status register are cleared.

Figure 43: I2C Interrupt Generation



The interrupt status bits are cleared by a write-one-to-clear operation to the Interrupt Status Register, provided the interrupt status register has first been read. For test purposes, bits in the Interrupt Status Register can also be set by a write-one-to-set operation to the "I2C Interrupt Set Register".

A bit that is set in the Interrupt Status Register is cleared by a write-1-to-clear operation only after the register has first been read, and then providing another event that would result the interrupt condition has not occurred since the read of the register (see "Events versus Interrupts".

16.11 Events versus Interrupts

Interrupts are generated by I2C events. Figure 44 shows the design of the event and interrupt logic. A single interrupt status bit may be derived from one or more events. The event registers provide control over the individual events that in turn produce the interrupt status. In the diagram, the shaded boxes represent virtual registers. These registers behave correctly when read or written, but can be constructed from combinational logic as opposed to flip-flops. Whether a register is virtual or not is inconsequential to their behavior from a software perspective. The distinction is shown only for exactness.

A new event is set in the "I2C New Event Register" when an event is asserted in the logic, or when a 1 is written to the register (or to the related interrupt bit in the "I2C Interrupt Set Register"). New events are ored with the I2C_SNAP_EVENT register to create the virtual I2C_EVENT register. A snapshot operation occurs when the "I2C Interrupt Status Register" is read. As a result of the snapshot, the new event register is "copied" to the snapshot register by oring the new events into the current snapshot state, then clearing the new event register. Each event is anded with the corresponding enable bit in the "I2C Enable Event Register", and then ored with any other enabled events that are related to a single interrupt status bit. The combined event state becomes the interrupt status bit in the Interrupt Status Register, and is then anded with the corresponding enable in the "I2C Interrupt Enable Register". All the enabled interrupt status bits are then ored together to become the single interrupt signal to the Interrupt Controller.

The new event and snapshot registers separate events that occurred prior to a read of the interrupt status register from those that occur during or after the read. When a 1 is written to the interrupt status register to clear an interrupt, all related events that are enabled are cleared in the snapshot register. Since events are copied to the snapshot register only when the interrupt status register is read, the read must be completed first for the write 1 to clear to have effect. If no new events have occurred, this write-1-to-clear de-asserts the interrupt status. If a new event has occurred, the event remains set in the new event register, so the interrupt status remains set.

For control purposes, software can read and clear the snapshot event bits directly, allowing individual events to be cleared while leaving any new events intact. Software can also select to read or clear events using the new event register. Reading the event register shows the "or" of the new and snapshot, and thus shows whether an event is asserting. Writing a 1 to an event bit clears both the snapshot and new events register bits, thus clearing out the event entirely, unless that event happens to be asserting again on the same cycle the clear is completed, thus setting it again.

As long as all event enables are set (the reset state), then the behavior is logical (see "Interrupt Handling").

Figure 44: I2C Event and Interrupt Logic

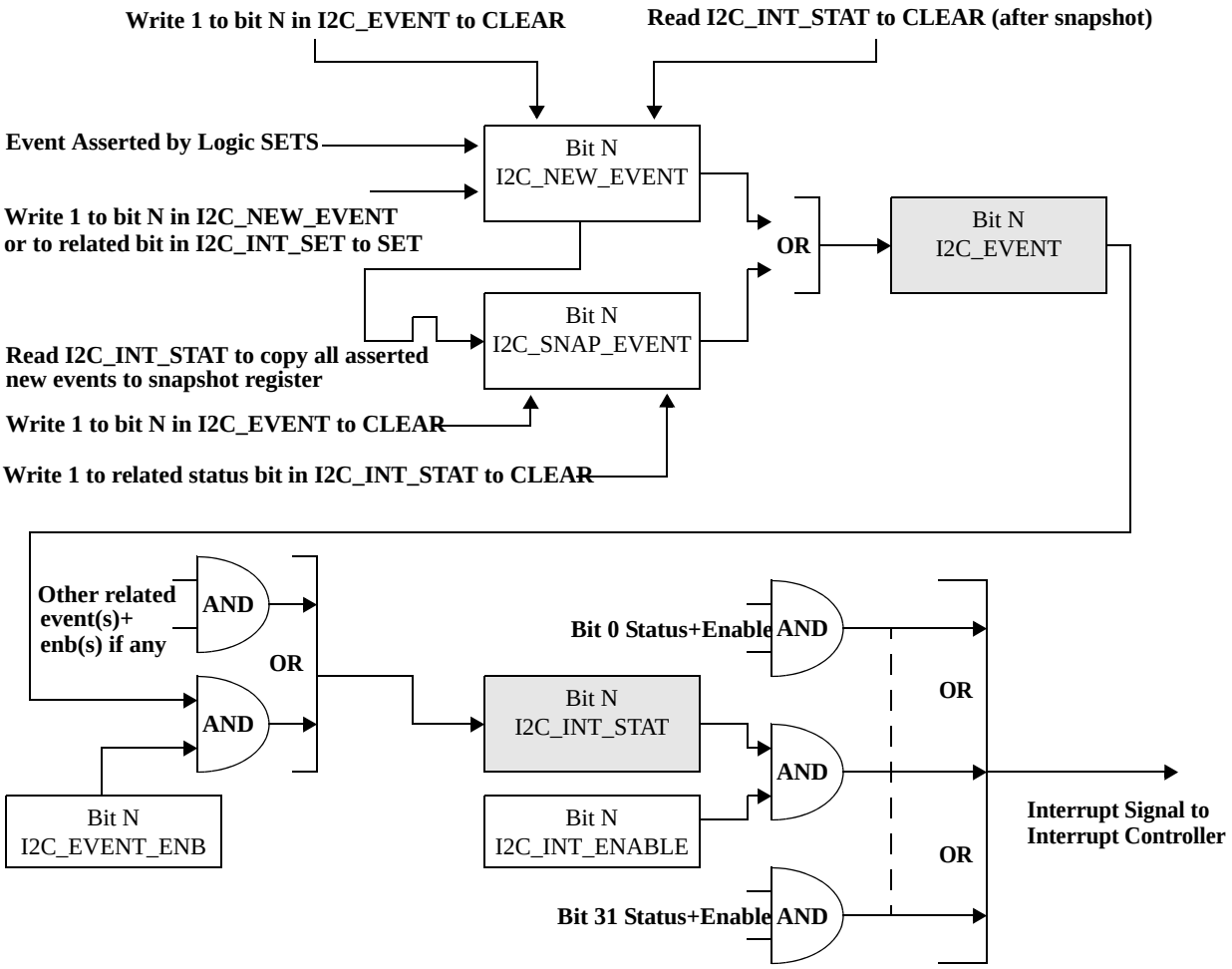


Table 97 shows the mapping of interrupts in the "I2C Interrupt Status Register" to the events in the "I2C Event and Event Snapshot Registers". Any asserted and enabled event sets the corresponding interrupt status, and clearing an asserted interrupt status bit clears all the related and enabled events.

Table 97: I2C Interrupt to Events Mapping

Interrupt Status Bit	Events Related to Interrupt
OMB_EMPTY (Outgoing Mailbox Empty)	OMBR (Outgoing Mailbox Read Event)
IMB_FULL (Incoming Mailbox Full)	IMBW (Incoming Mailbox Write Event)
BL_FAIL (Boot Load Fail)	BLTO (Boot Load Timeout Error) BLERR (Boot Load Error Event) BLSZ (Boot Load Size Error Event) BLNOD (Boot Load No Device Event)

Table 97: I2C Interrupt to Events Mapping (Continued)

Interrupt Status Bit	Events Related to Interrupt
BL_OK (Boot Load OK)	BLOK (Boot Load OK Event)
SA_FAIL (Slave Access Failed)	SCOL (Slave Collision Detect Event) STRTO (Slave Transaction Timeout Event) SBTTO (Slave Byte Timeout Event) SSCLTO (Slave I2C_SCLK Low Timeout Event)
SA_WRITE (Slave Access Write)	SDW (Slave Internal Register Write Done Event)
SA_READ (Slave Access Read)	SDR (Slave Internal Register Read Done Event)
SA_OK (Slave Access OK)	SD (Slave Transaction Done Event)
MA_DIAG (Master Diagnostic Event)	DTIMER (Diagnostic Timer Expired Event) DHIST (Diagnostic History Filling Event) DCMDD (Diagnostic Command Done Event)
MA_COL (Master Collusion)	MCOL (Master Collision Detect Event)
MA_TMO (Master Timeout)	MTRTO (Master Transaction Timeout Event) MBTTO (Master Byte Timeout Event) MSCLTO (Master I2C_SCLK Low Timeout Event)
MA_NACK (Master NACK)	MNACK (Master NACK Received Event)
MA_ATMO (Master Arbitration Timeout)	MARBTO (Master Arbitration Timeout Event)
MA_OK (Master Transaction OK)	MTD (Master Transaction Done Event)

16.12 Timeouts

The I2C Interface supports a number of timeout periods to detect a set of error conditions related to I2C operation. These timeouts, and the registers that configure them, include the following:

- I2C_SCLK low timeout (see "I2C_SCLK Low and Arbitration Timeout Register") – This timeout detects a situation where a device on the bus is stuck holding the clock low. Because the clock is stuck low, no progress can be made. If enabled, this timeout expiring will set either the SSCLTO or MSCLTO events and result in a SA_FAIL or MA_TMO interrupt status being updated in the I2C_INT_STAT register (depending on whether a master or slave operation was in progress). An optional interrupt can be sent to the Interrupt Controller if SA_FAIL or MA_TMO is enabled in the "I2C Interrupt Enable Register". This is an extreme failure. With I2C_SCLK held low, no Stop condition can be generated. Any operation is aborted, both I2C_SCLK and I2C_SD are released, and both master and slave revert to their monitor-for-bus-idle phase. It is up to software to decide how to handle this error. Because any operation was aborted without correct termination (no Stop), it is possible that the external device is left in an invalid state.

- Arbitration timeout (see "I2C_SCLK Low and Arbitration Timeout Register") – This timeout applies only to master transactions initiated by setting the START bit in the "I2C Master Control Register". Its purpose is to limit the length of time the master controller tries to gain ownership of the bus. The arbitration timer is disabled once the <Start><Slave Address><Read/Write> are successfully transmitted without detecting another master attempting a different transaction. If the Tsi620 I2C master subsequently loses ownership of the bus after this phase of the transaction, the transaction is aborted. If the Tsi620 I2C master detects another master corrupting the <Start><Slave Address><Read/Write> bits it has transmitted, the Tsi620 I2C master reverts to waiting for bus idle then tries again. The arbitration timeout continues to run in this case. If the arbitration timer expires before ownership is gained and the master is waiting for bus idle, then it aborts the operation and sets the MARBTO event, which causes a MA_ATMO interrupt status to be updated in the "I2C Interrupt Status Register". An optional interrupt can also be sent to the Interrupt Controller if the MA_ATMO is enabled in the "I2C Interrupt Enable Register".

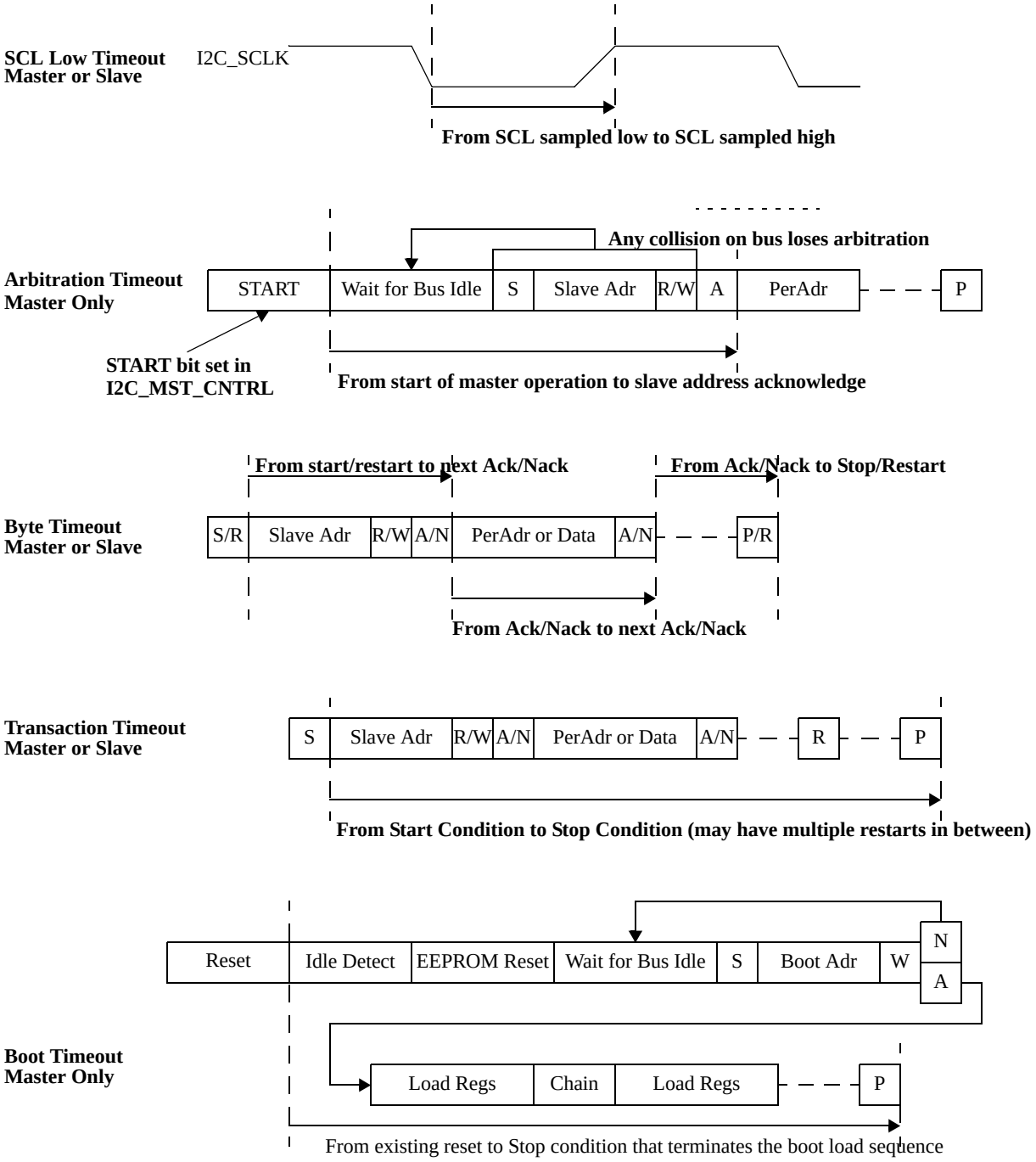
  If the Tsi620 I2C master was in the midst of transmitting the <Slave Address> when the timeout expires, it allows the <Slave Address> to complete. If an ACK or NACK is successfully received, the master continues as if the timeout had not expired. If another I2C master collides with <Slave Address>, the timeout immediately takes effect following the <Slave Address> bit where the collision took place.

- Byte timeout (see "I2C Byte/Transaction Timeout Register") – This timeout is disabled on reset. It detects a situation where one or more devices are stretching the clock enough to slow the transfer speed on the bus beyond some limit. This timeout is available primarily to detect a violation of the SMBus TLOW:MEXT time. The response to this timeout expiring depends on the phase of the transfer and whether it is detected by the master or slave interface. For a master transaction, the master continues to generate clocks until the next bit time where it would have control of the bus; that is, writing data or generating an Ack/Nack in response to a read byte. At that time, the master generates a Stop condition, aborts the operation and sets the MBTTO event, which causes an MA_TMO interrupt status to get updated in the "I2C Interrupt Status Register". An optional interrupt can also be sent to the Interrupt Controller if the MA_TMO bit is enabled in the "I2C Interrupt Enable Register". For a slave transaction, the slave waits for the start of the next bit time, releases the I2C_SD and I2C_SCLK signals and sets the SBTTO event, which causes an SA_FAIL interrupt status to get updated in the I2C_INT_STAT register. An optional interrupt can be sent to the Interrupt Controller if the SA_FAIL bit is enabled in the "I2C Interrupt Enable Register". The slave then reverts to looking for the next Start/Restart/Stop.

- Transaction timeout (see “I2C Byte/Transaction Timeout Register”) – This timeout is disabled on reset. It detects a situation where a master is keeping the bus for an extended period of time, as measured from the Start to Stop condition. This timeout is available primarily to detect a violation of the SMBus TLOW:SEXT time. The response to this timeout expiring is identical to a Byte timeout, with the exception that the events are MTRTO or STRTO for the master or slave respectively.
- Boot timeout (see “I2C Boot and Diagnostic Timer”) – This timeout detects a situation where the boot load sequence has not completed in a reasonable time. This could occur if the EEPROM was improperly programmed with an infinite chaining loop, the bus ownership is held by some other device, or some other anomalous situation resulting in any of the time-outs above. If the boot timeout expires before the normal end of the boot load sequence, the master interface reads until the next data byte and drives a Stop condition on the bus. It then sets the BLTO event, which causes a BL_FAIL interrupt status to get updated in the “I2C Interrupt Status Register”. An optional interrupt can also be sent to the Interrupt Controller if the BL_FAIL bit is enabled in the “I2C Interrupt Enable Register”. The boot_complete signal is asserted when the boot load timeout expires. If the boot timeout is not desired, then the EEPROM programming should immediately write the I2C_BOOT_DIAG_TIMER.COUNT to 0 to disable the timeout.

Figure 45 shows the relationship of the I2C time-outs to I2C operations.

Figure 45: I2C Timeout Periods



16.13 Bus Timing

Figure 46 shows the relationship of the bus timing parameters to the generation of the I2C_SCLK and I2C_SD signals on the I2C bus. These parameters are configured in the following registers:

- "I2C Start Condition Setup/Hold Timing Register"
- "I2C Stop/Idle Timing Register"
- "I2C_SD Setup and Hold Timing Register"
- "I2C Stop/Idle Timing Register"
- "I2C_SCLK High and Low Timing Register"
- "I2C_SCLK Minimum High and Low Timing Register"

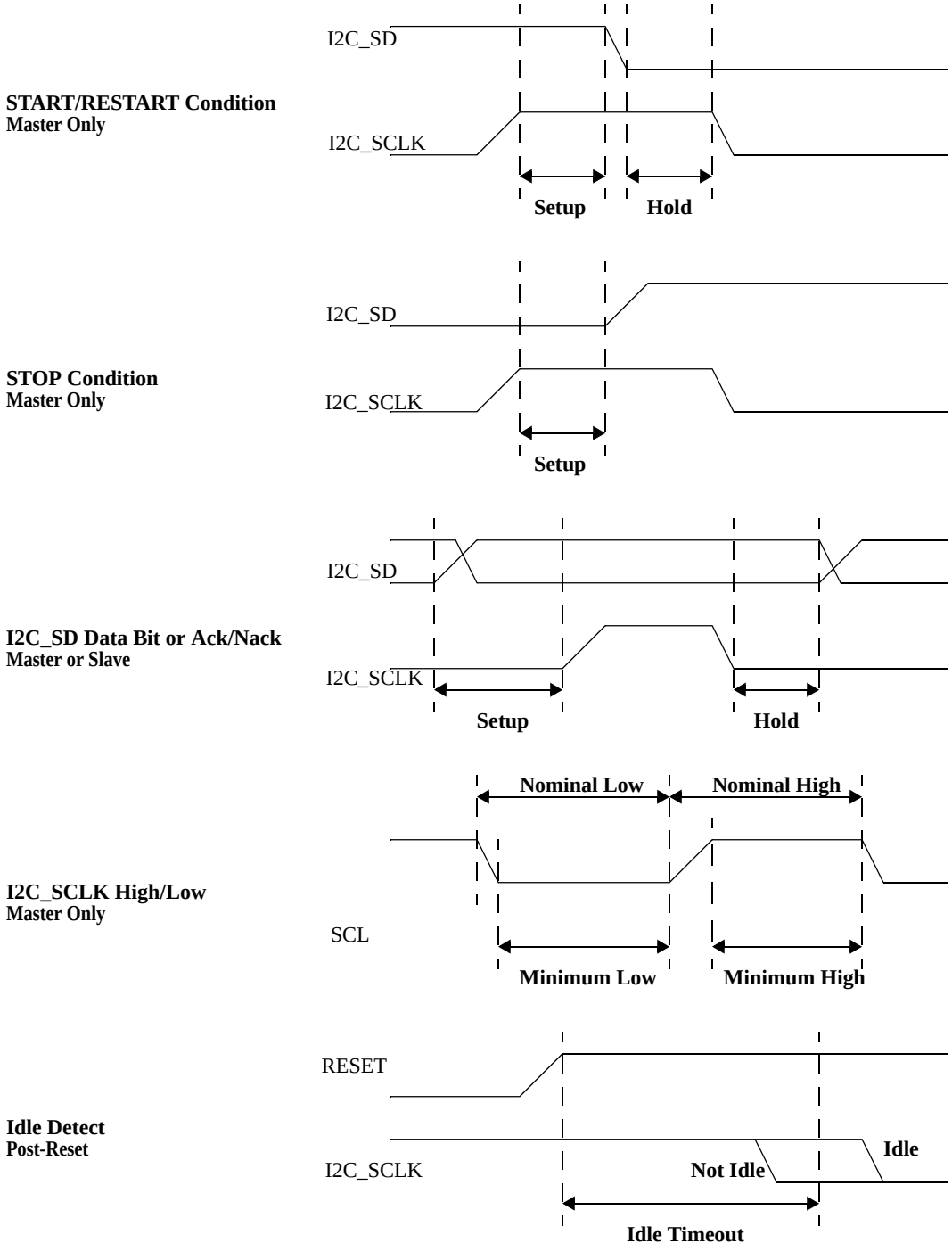
The bus timing resets to 100-kHz operation. By reprogramming these registers, other bus speeds can be configured. Speeds above 100 kHz are not guaranteed to conform to the *I2C Specification* because of the absence of Schmitt triggers on the input of the I2C_SD and I2C_SCLK signals, and the absence of slope controlled outputs for the I2C_SD and I2C_SCLK signals. It is up to the board or system designer to decide on the applicability of operation at speeds above 100 kHz.

Bus timing does not normally change during a transaction, even if these registers are changed. The timing registers are sampled at certain times to prevent this from occurring. The following are the times when timing adjustments take effect:

- When reset using the "I2C Reset Register"
- At the start of a master transaction through the "I2C Master Control Register", when the START condition is generated
- Upon a chain operation during boot load

Timing parameters are discussed further in the following sections.

Figure 46: I2C Bus Timing Diagrams



16.13.1 Start/Restart Condition Setup and Hold

The Start/Restart Condition is generated by a master. As shown in Figure 46, the Start Setup time defines the minimum period both the I2C_SD and I2C_SCLK signals must be seen high (1) before the I2C_SD signal is pulled low (0) to trigger the Start. The I2C_SD signal must also have fulfilled the I2C_SD Setup time prior to the rising edge of I2C_SCLK. Once the I2C_SD signal is seen low (0), the Start Hold time is the minimum period the I2C_SCLK signal must continue to remain high (1) before it is pulled low (0). These parameters are used by the Tsi620 as a master when generating the Start condition. These times may be violated by an external master or slave pulling the I2C_SD or I2C_SCLK signals low before the setup/hold periods are expired, which may result in an arbitration loss or collision.

16.13.2 Stop Condition Setup

The Stop Condition is generated by a master. As shown in Figure 46, the Stop Setup time defines the minimum period the I2C_SD must be seen low (0) and the I2C_SCLK signal must be seen high (1) before the I2C_SD signal is released high (1) to trigger the Start. The I2C_SD signal must also have fulfilled the I2C_SD Setup time prior to the rising edge of I2C_SCLK. There is no separate Stop Hold parameter, as the only valid condition following a Stop would be a Start; therefore, the Start Setup fulfills the same use as a Stop Hold or Stop-to-Start buffer time. This parameter is used by the Tsi620 as a master when generating the Stop condition. If the I2C_SCLK signal was prematurely pulled low (0) by an external master or slave, this would be seen as a collision event.

16.13.3 I2C_SD Setup and Hold

Either a master or a slave can be in control of the I2C_SD signal, depending on the phase of the data transfer protocol. As shown in Figure 46, the I2C_SD Setup time defines the minimum period the I2C_SD signal must set to the desired state while I2C_SCLK is low (0) before the I2C_SCLK signal is release high (1) to generate the high period of the clock. The I2C_SD Hold time defines the minimum period the I2C_SD signal is left unchanged after the falling edge of I2C_SCLK (I2C_SCLK seen low). The I2C_SD hold time may be violated by another device pulling I2C_SD low, but this is not an error, as it normally indicates another device with a different design.

The I2C_SD setup time is not as defined in the *I2C Specification*. The setup time parameter encompasses both the maximum rise/fall time of the I2C_SD signal plus the output hold time and must be set accordingly. There is no feedback check that the I2C_SD signal goes to the desired state, as this could result in I2C_SCLK being held low erroneously. If another device is also controlling I2C_SD, the likely result is an arbitration loss or collision.

16.13.4 I2C_SCLK Nominal and Minimum Periods

These parameters are used by the Tsi620 as a master to generate the I2C_SCLK clock. The master must obey the minimum times to conform to the *I2C Specification*, and must also attempt to regulate the overall I2C_SCLK frequency to a defined period. From Figure 46, it can be seen that the logic measures the minimum periods high/low from the detected rising/falling edges of the I2C_SCLK signal to the point where I2C_SCLK is driven low or released high to generate the opposing edge. In conjunction, a separate nominal period timer measures from driven low to released high, and released high to driven low. Both timers must expire if unaffected by external devices. If another device pulls the I2C_SCLK signal low prematurely in the high period, the high period timers are expired and the lower period timers restart for the low period, so the actual low period may be stretched by the nominal timer. If another device holds the I2C_SCLK signal low longer in the low period than the nominal low period, the high period nominal timer will likely expire early and the minimum high period timer will control the high period when the clock is finally released.

16.13.5 Idle Detect Period

This is a master-only parameter that is used in two cases. First, upon exit from reset it is unknown if another master is active. The Idle Detect timeout determines if the I2C_SCLK signal remains high long enough (roughly 50 microseconds) that it is unlikely another master is active. If I2C_SCLK is seen low during this period, it is assumed another master is active, and the master enters the Wait for Bus Idle phase. If the idle detect period expires without I2C_SCLK seen low, then it is assumed the bus is idle and the master is free to generate a Start Condition if needed.

Second, during the Wait for Bus Idle phase, it is possible that an external master that has claimed the bus ceases activity without issuing a STOP condition. When a master operation is started but the bus is currently seen busy, the idle detect timer monitors the I2C_SCLK and I2C_SD signals. If the I2C_SCLK and I2C_SD signals both remain high longer than the idle detect period, the bus is then assumed idle even though a STOP had not been seen, and the master logic will attempt the requested transaction.

17. JTAG Interface

Topics discussed include the following:

17.1 Overview

The JTAG Interface is compliant with IEEE 1149.6 B*oundary Scan Testing of Advanced Digital Networks*, as well as IEEE 1149.1 *Standard Test Access Port and Boundary Scan Architecture* standards. There are five standard pins associated with the interface (TMS, TCK, TDI, TDO and TRST_b) which allow full control of the internal TAP (Test Access Port) Controller. The JTAG Interface has the following features:

- Contains a 5-pin Test Access Port (TAP) Controller, with support for the following registers:
  - Instruction register (IR)
  - Boundary scan register
  - Bypass register
  - Device ID register
  - User test data register (DR)
- Supports debug access of Tsi620's configuration registers
- Supports the following instruction opcodes:
  - Sample/Preload
  - Extest
  - EXTEST_PULSE *(1149.6)*
  - EXTEST_TRAIN *(1149.6)*
  - Bypass
  - IDCODE
  - Clamp
  - User data select

17.2 JTAG Device Identification Number

The JTAG device identification number for the Tsi620 is 0x00620167.

17.3 JTAG Register Access

Users can read and write Tsi620's registers through the JTAG Interface. The interface can access registers to debug issues that can affect register accesses. Register access through the JTAG Interface can also be used in normal mode to do extensive read and write accesses on the performance registers without affecting normal traffic in the device or during initialization.

Prior to using the IEEE Register Access Command feature, the Tsi620 must be reset by driving TRST_b low.

A user-defined command enables the read and write capabilities of the interface. The command is in the IEEE 1149.1 Instruction Register (IR) in the Tsi620.

- IEEE Register Access Command (IRAC)

There must be IEEE 1149.1 capability on the board to use the IEEE 1149.1 register access feature.

17.3.1 Format

The format used to access the registers is shown in Figures 47 and 48. The address shown in the figure is the RapidIO offset.

Figure 47: Register Access From JTAG — Serial Data In

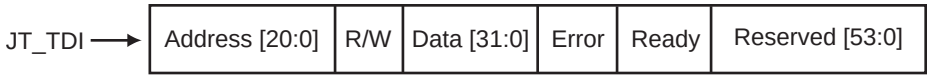
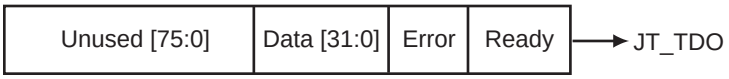


Figure 48: Register Access From JTAG — Serial Data Out



17.3.2 Write Access to Registers from the JTAG Interface

The following steps are required to write to a register through the JTAG Interface:

1. Move to the Tap Controller "Shift-IR" state and program the instruction register with IRAC instruction. This is completed by shifting in IR length instruction register, which is 123 bits of all ones except for the second last bit (for example, ...1111_1101).
2. Move to the "Shift-DR" state and shift the data[31:0], R/W = 1 and the address[20:0] serially in the TDI pin. To prevent corruption of unused bits, the full DR bits have to be written with the following values:
   - DR[109:89] = ADDR[20:0]
   - DR[88] = R/W
   - DR[87:56] = DATA[31:0]

- DR[55:54] = 0b0
- DR[53:0] = 0b0

3. Move to the “Run-test idle” state and loop in this state for a minimum of 20 TCK cycles.
4. Move to the “Shift-DR” state again and shift-in 110 zero bits to DR[109:0], while at the same time verify the Ready and Error bits that are being shifted-out as the first two bits.
5. Go back to step two to perform another write.

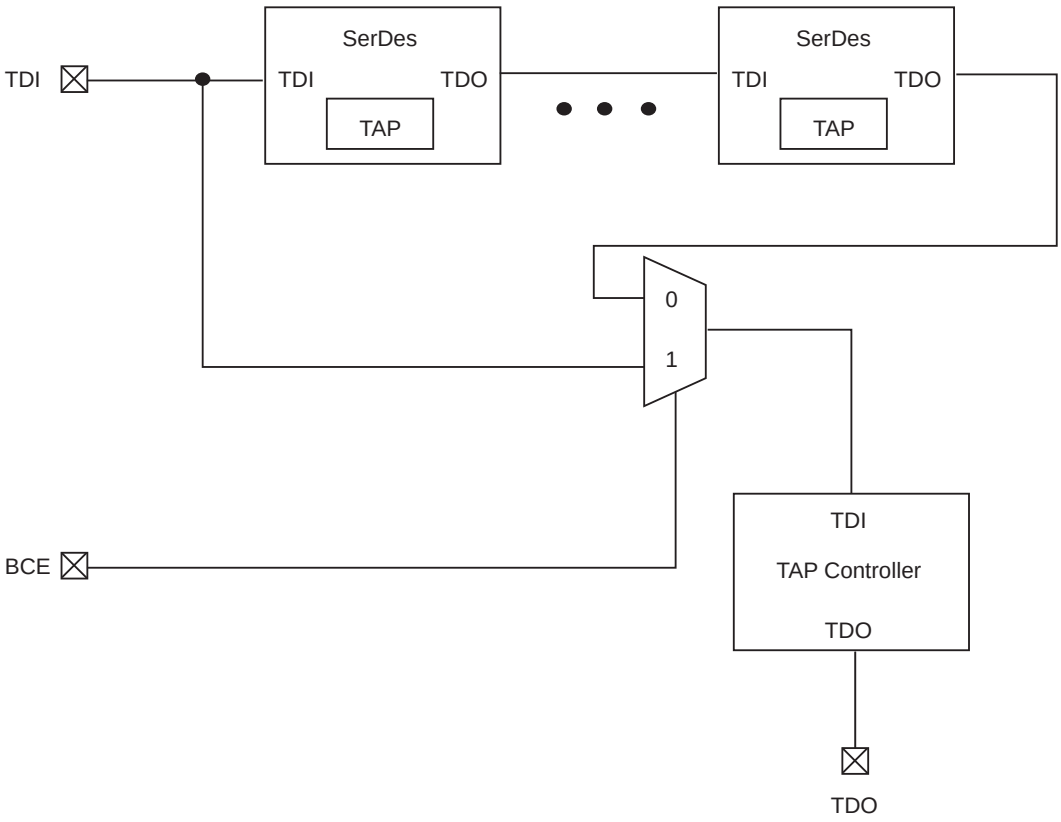
17.3.3 Read Access to Registers from the JTAG Interface

The following steps are required to read a register through the JTAG Interface:

1. Move to the Tap Controller “Shift-IR” state and program the instruction register with IRAC instruction.

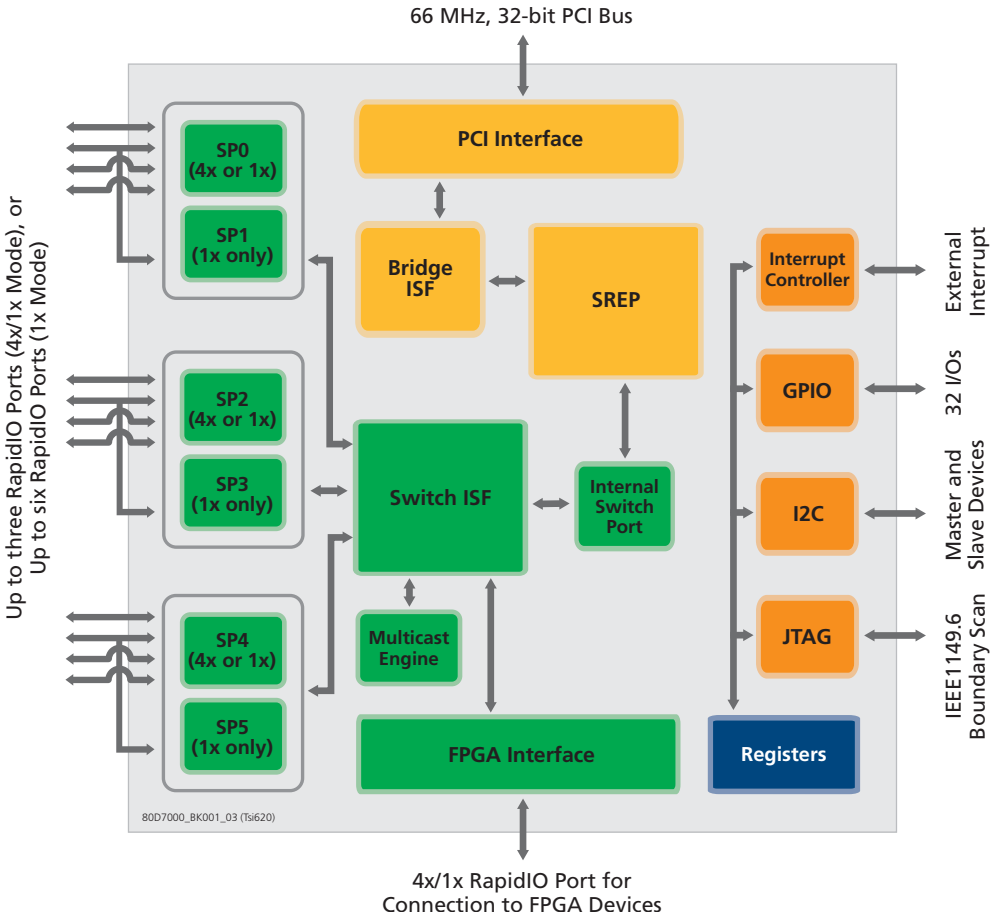
   This step is optional if the instruction register is already programmed during the write cycle.
2. Move to the “Shift-DR” state and shift the R/W = 0 and the address[20:0] serially in the TDI pin. To prevent corruption of unused bits, the full DR bits have to be written. The following values must be written:
   - DR[109:89] = ADDR[20:0]
   - DR[88] = R/W
   - DR[87:56] = DATA[31:0]
   - DR[55:54] = 0b0
   - DR[53:0] = 0b0
3. Move to the “Run-test idle” state and loop in this state for a minimum of 20 TCK cycles.
4. Move to the “Shift-DR” state and shift in 110 bits of 0. The first two bits of data shifted out are the Error and Ready bits. The next 32 bits are data. The remainder of the shifted out data, the Unused bits in Figure 48, can be discarded.
5. The Error and Ready bits are shifted out at the same time.
6. Verify that the Error bit is at logic low and the Ready bit is at logic high.
7. Go back to step two to perform another read.

Figure 49: Conceptual Diagram of Daisy Chaining the TAP Controllers



PART 5: OTHER TOPICS

This part of the document discusses other features and functions that are integral to the Tsi620, including signal descriptions, clocking, reset, and initialization options.



18. Signal Descriptions

This chapter provides signal descriptions and recommended termination information for the Tsi620. Topics discussed include the following:

18.1 Overview

The following conventions are used in this chapter:

- Signals with the suffix "_p" are the positive half of a differential pair.
- Signals with the suffix "_n" are the negative half of a differential pair.
- Signals with the suffix "_b" are active low.

Signals are classified according to the types defined in the following table.

Table 98: Signal Types

Pin Type	Definition
I	Input
O	Output
I/O	Input/Output
OD	Open Drain
SRIO	Differential driver/receiver defined by *RapidIO Interconnect Specification (Revision 1.3)*

Table 98: Signal Types (Continued)

Pin Type	Definition
CML	Defined by *RapidIO Interconnect Specification (Revision 1.3)*
PU	Pulled Up internal to the Tsi620
PD	Pulled Down internal to the Tsi620
HSTL	HSTL I/O with 1.5V thresholds.
LVTTL	CMOS I/O with LVTTL thresholds
Hyst	Hysteresis
Core Power	Core supply
Core Ground	Ground for core logic
I/O Power	I/O supply

18.2 RapidIO Signals

Table 99: RapidIO Signals

Pin Name	Type	Description	Recommended Termination
Port n - 1x/4x Mode RapidIO Port (n+1) - 1x Mode RapidIO, where n = 0, 2, 4			
Serial Port n/n+1 Transmit, where n = 0, 2, 4			
SP{n}_TA_p	O, SRIO	Port n Lane A Differential Non-inverting Transmit Data output (4x mode) Port n Differential Non-inverting Transmit Data output (1x mode)	Leave all unused ports/lanes as no-connect.
SP{n}_TA_n	O, SRIO	Port n Lane A Differential Inverting Transmit Data output (4x mode) Port n Differential Inverting Transmit Data output (1x mode)	Leave all unused ports/lanes as no-connect.
SP{n}_TB_p	O, SRIO	Port n Lane B Differential Non-inverting Transmit Data output (4x mode) Port n+1 Differential Non-inverting Transmit Data output (1x mode)	No termination required.
SP{n}_TB_n	O, SRIO	Port n Lane B Differential Inverting Transmit Data output (4x mode) Port n+1 Differential Inverting Transmit Data output (1x mode)	No termination required.

Table 99: RapidIO Signals (Continued)

Pin Name	Type	Description	Recommended Termination
SP{n}_TC_p	O, SRIO	Port n Lane C Differential Non-inverting Transmit Data output (4x mode)	No termination required.
SP{n}_TC_n	O, SRIO	Port n Lane C Differential Inverting Transmit Data output (4x mode)	No termination required.
SP{n}_TD_p	O, SRIO	Port n Lane D Differential Non-inverting Transmit Data output (4x mode)	No termination required.
SP{n}_TD_n	O, SRIO	Port n Lane D Differential Inverting Transmit Data output (4x mode)	No termination required.
Serial Port n/n+1 Receive, where n = 0, 2, 4			
SP{n}_RA_p	I, SRIO	Port n Lane A Differential Non-inverting Receive Data input (4x node) Port n Differential Non-inverting Receive Data input (1x mode)	DC blocking capacitor of 0.1uF in series.
SP{n}_RA_n	I, SRIO	Port n Lane A Differential Inverting Receive Data input (4x node) Port n Differential Inverting Receive Data input (1x mode)	DC blocking capacitor of 0.1uF in series.
SP{n}_RB_p	I, SRIO	Port n Lane B Differential Non-inverting Receive Data input (4x mode) Port n+1 Differential Non-inverting Receive Data input (1x mode)	DC blocking capacitor of 0.1uF in series.
SP{n}_RB_n	I, SRIO	Port n Lane B Differential Inverting Receive Data input (4x mode) Port n+1 Differential Inverting Receive Data input (1x mode)	DC blocking capacitor of 0.1uF in series.
SP{n}_RC_p	I, SRIO	Port n Lane C Differential Non-inverting Receive Data input (4x mode)	DC blocking capacitor of 0.1uF in series.
SP{n}_RC_n	I, SRIO	Port n Lane C Differential Inverting Receive Data input (4x mode)	DC blocking capacitor of 0.1uF in series.
SP{n}_RD_p	I, SRIO	Port n Lane D Differential Non-inverting Receive Data input (4x mode)	DC blocking capacitor of 0.1uF in series.
SP{n}_RD_n	I, SRIO	Port n Lane D Differential Inverting Receive Data input (4x mode)	DC blocking capacitor of 0.1uF in series.
Serial Port Configuration			
SP{n}_REXT n = 0, 2, 4	Analog	Used to connect a resistor to VSS to provide a reference current for the driver and equalization circuits.	Must be connected to VSS with a 191-ohm (1%) resistor.

Table 99: RapidIO Signals (Continued)

Pin Name	Type	Description	Recommended Termination
SP{n}_MODE_SEL n = 0, 2, 4	I/O, LVTTL, PD	See SP{n}_MODE_SEL in "Power-up Configuration Signals".	
SP6_MODE_SEL	I/O, LVTTL, PD	See SP6_MODE_SEL in "Power-up Configuration Signals".	
SP{n}_PWRDN, n = 2, 4	I/O, LVTTL, PU	See SP{n}_PWRDN in "Power-up Configuration Signals".	
SP{n+1}_PWRDN, n = 0, 2, 4	I/O, LVTTL, PU	See SP{n+1}_PWRDN in "Power-up Configuration Signals".	
SP6_PWRDN	I/O, LVTTL, PU	See SP6_PWRDN in "Power-up Configuration Signals".	
Serial Port Speed Select			
SP_CLK_SEL[1,0]	I/O, LVTTL, [PU,PD]	See SP_CLK_SEL[1,0] in "Power-up Configuration Signals".	
SP_IO_SPEED[1,0]	I/O, LVTTL, [PU,PD]	See SP_IO_SPEED in "Power-up Configuration Signals".	
Serial Port Lane Ordering Select			
SP_RX_SWAP	I, LVTTL, PD	See SP_RX_SWAP/SP_TX_SWAP in "Power-up Configuration Signals".	
SP_TX_SWAP	I, LVTTL, PD		
SREP Mode Select			
SP_HOST	I, LVTTL, PD	See SP_HOST in "Power-up Configuration Signals".	
SP_MAST_EN	I, LVTTL, PD	See SP_MAST_EN in "Power-up Configuration Signals".	

18.3 FPGA Interface Signals

Table 100: FPGA Interface Signals

Pin Name	Type	Description	Recommended Termination
SP6_RXD[31:0]	I/O, HSTL	Data received from the FPGA Interface link partner. These signals are organized into groups of 8 data signals, whereby each group corresponds to a RapidIO lane. SP6_RXCLK is center-aligned with these signals. The output capabilities of these signals are used only for test purposes.	Use Altera Stratix III on-die termination. Transmitter configured as a class II 25 ohm series termination.
SP6_RXCLK	I/O, HSTL	This is a center-aligned clocking signal for the FPGA Interface receive side. The output capabilities of this signal is used only for test purposes.	Use Altera Stratix III on-die termination. Transmitter configured as a class II 25 ohm series termination.
SP6_RXCTL[3:0]	I/O, HSTL	Indicates whether each 8-bit lane of SP6_RXD[31:0] is a data character or a special character. [a] 0 = Lane contains a data character 1 = Lane contains a special (K) character SP6_RXCLK is center-aligned with this signal. The output capabilities of these signals are only used for test purposes.	Use Altera Stratix III on-die termination. Transmitter configured as a class II 25 ohm series termination.
SP6_RX_ERROR	I/O, HSTL	Indicates whether the data being received is an error. 0 = Data is correct 1 = Data is in error The output capabilities of this signal is used only for test purposes.	Use Altera Stratix III on-die termination. Transmitter configured as a class II 25 ohm series termination.
SP6_TXD[31:0]	O, HSTL	Data transmitted to the FPGA Interface link partner. These signals are organized into groups of 8 data signals, whereby each group corresponds to a RapidIO lane. SP6_TXCLK is center-aligned with these signals.	Use Altera Stratix III on-die termination. Receiver configured as a class I 50 ohm parallel termination.
SP6_TXCLK	O, HSTL	This is a center-aligned clocking signal for the FPGA Interface transmit side.	Use Altera Stratix III on-die termination. Receiver configured as a class I 50 ohm parallel termination.

Table 100: FPGA Interface Signals (Continued)

Pin Name	Type	Description	Recommended Termination
SP6_TXCTL[3:0]	O, HSTL	Indicates whether each 8-bit lane of PS6_RXD[31:0] is a data character or a special character. [a] 0 = Lane contains a data character 1 = Lane contains a special (K) character SP6_TXCLK is center-aligned with this signal.	Use Altera Stratix III on-die termination. Receiver configured as a class I 50 ohm parallel termination.
SP6_PHY_DISABLE	O, HSTL	Indicates whether the data being transmitted is an error. 0 = Data is correct 1 = Data is in error This signal is asynchronous to other signals.	Use Altera Stratix III on-die termination. Receiver configured as a class I 50 ohm parallel termination.

a. For more information on the encoding of character values in RapidIO, see the *RapidIO Interconnect Specification (Revision 1.3) Part 6: 1x/4x LP-Serial Specification*.

18.4 Multicast Event Control Symbol Signals

Table 101: Multicast Signals

Pin Name	Type	Description	Recommended Termination
MCES	I/O, LVTTL, PD	Multicast Event Control Symbol (MCES) signal. As an input, an edge (rising or falling) triggers a MCES to be sent to all ports. As an output, this pin toggles its value every time an MCES is received by any port that is enabled for Multicast even control symbols. The maximum frequency for this signal as an input or output is 1 MHz. This pin is only reset by assertion of CHIP_RST_b. No other reset will affect this pin.	No termination required. This pin must not be driven by an external source until all power supply rails are stable.

18.5 PCI Signals

Table 102: PCI Signals

Name	Pin Type	Description[a]	Recommended Termination
PCI_AD[31:0]	PCI Bidir	Address/Data Bus. These multiplexed signals provide a 32-bit address and data bus.	No termination required.
PCI_CBEn[3:0]	PCI Bidir	Command/Byte Enables. These multiplexed signals indicate the current transaction type.	No termination required.

Table 102: PCI Signals (Continued)

Name	Pin Type	Description[a]	Recommended Termination
PCI_CLK	PCI In	PCI Clock. This signal provides timing for all devices on the PCI bus.	If PCI_CLKO[n] is used to supply the Tsi620 PCI_CLK signal, use a 33-Ohm series resistor.
PCI_DEVSELn	PCI Bidir	Device Select. A target device asserts this signal when it decodes its address on the bus. The master samples the signal at the beginning of a transaction, and the target rescinds it at the end of the transaction.	Pull up using a 8.2K resistor to VDD_PCI.[b]
PCI_FRAMEn	PCI Bidir	Frame. The current initiator drives this signal to indicate the start and duration of a transaction, and the bus the bus target samples it. The bus master rescinds the signal at the end of the transaction.	Pull up using a 8.2K resistor to VDD_PCI.[b]
PCI_GNTn[4:1]	PCI Bidir	Multi-function signal (Bus Grant). These signals grant access to the PCI bus; however, they are used differently depending on whether or not the Tsi620 PCI arbiter is used. If the PCI arbiter is used, then PCI_GNTn[4:1] are outputs used by the Tsi620 to grant access to the bus. If an external arbiter is used, PCI_GNTn[1] is an input that is driven by the arbiter to grant the Tsi620 access to the bus, while PCI_GNTn[4:2] should be pulled high.	Pull up using a 8.2K resistor to VDD_PCI.
PCI_IDSEL	PCI In	Initialization Device Select. This signal is used as a chip select during configuration read and write transactions.	Resistively couple using a 200-ohm resistor to AD line.
PCI_INTAn	PCI Bidir OD	Interrupt A.	Pull up using a 8.2K resistor to VDD_PCI.[c]
PCI_INTBn	PCI Bidir OD	Interrupt B.	Pull up using a 8.2K resistor to VDD_PCI.[c]
PCI_INTCn	PCI Bidir OD	Interrupt C.	Pull up using a 8.2K resistor to VDD_PCI.[c]
PCI_INTDn	PCI Bidir OD	Interrupt D.	Pull up using a 8.2K resistor to VDD_PCI.[c]
PCI_IRDYn	PCI Bidir	Initiator Ready. The bus master asserts this signal to indicate it is ready to complete the current transaction.	Pull up using a 8.2K resistor to VDD_PCI.[b]
PCI_PAR	PCI Bidir	Parity. This signal carries even parity across PCI_AD[31:0] and PCI_CBEn[3:0]. The bus master asserts this signal for the address and write data phases. The bus target assets it for read data phases.	No termination required.

Table 102: PCI Signals (Continued)

Name	Pin Type	Description[a]	Recommended Termination
PCI_PERRn	PCI Bidir	Parity Error. This signal indicates a parity error occurred during the current data phase. The bus target that receives the data asserts this signal.	Pull up using a 8.2K resistor to VDD_PCI.[b]
PCI_PMEn	PCI Bidir OD	Power Management Event. This signal indicates a power management event occurred.	Pull up using a 8.2K resistor to VDD_PCI.[b]
PCI_REQn[4:1]	PCI Bidir	Bus Request. These signals request access to the PCI bus; however, they are used differently depending on whether or not the Tsi620 PCI arbiter is used. If the PCI arbiter is used, then PCI_REQn[4:1] are inputs used by external masters to request access to the bus. If an external arbiter is used, PCI_REQn[1] is an output used by the Tsi620 to request access to the bus, while PCI_REQn[4:2] should be pulled high.	Pull up using a 8.2K resistor to VDD_PCI.[b]
PCI_RSTn	PCI Bidir	Reset. This signal performs an asynchronous reset of the PCI Interface. If asserted, it forces all PCI configuration registers, master and target state machines, and output drivers, to an initialized state.	Pull up using a 8.2K resistor to VDD_PCI.[b]
PCI_RSTDIR	PCI In	See PCI_RSTDIR in "Power-up Configuration Signals".	
PCI_SERRn	PCI Bidir OD	System Error. This signal indicates an address parity error occurred.	Pull up using a 8.2K resistor to VDD_PCI.[b]
PCI_STOPn	PCI Bidir	Stop. A bus target asserts this signal to indicate it wants to stop the active transaction on the current data phase.	Pull up using a 8.2K resistor to VDD_PCI.[b]
PCI_TRDYn	PCI Bidir	Target Ready. The bus target asserts this signal to indicate it is ready to complete the current data phase.	Pull up using a 8.2K resistor to VDD_PCI.[b]
PCI Clocking			
PCI_M66EN	PCI In	See PCI_M66EN in "Power-up Configuration Signals".	
PCI_CLKO[0:4]	PCI Out	PCI Clock outputs The Tsi620 provides five PCI clock output signals to act as PCI reference clock signals for devices on the PCI bus (see "Clock Generation").	Pull down to VSS_IO using a 50-Ohm series resistor. Unused clocks can be left unconnected.
PCI Configuration Signals			
PCI_ARBEN	PCI In	See PCI_ARBEN in "Power-up Configuration Signals".	

Table 102: PCI Signals (Continued)

Name	Pin Type	Description[a]	Recommended Termination
PCI_PLL_BYPASS	PCI In	See PCI_PLL_BYPASS in "Power-up Configuration Signals".	
PCI_HOLD_BOOT	PCI In	See PCI_HOLD_BOOT in "Power-up Configuration Signals".	

a. The Tsi620 does not support CompactPCI Hot swap.

b. These pull-ups must exist somewhere in the system; usually an add-in card can rely on the motherboard to provide these terminations.

c. When PCI interrupts are configured as outputs, the interrupt outputs of the Tsi620 Interrupt Controller are driven to the PCI_INTn outputs of the Tsi620.

18.6 I^2C Signals

Table 103: I^2C Signals

Pin Name	Type	Description	Recommended Termination
I2C_SCLK	I/O, OD, LVTTL, PU	I^2C input/output clock, up to 100 kHz.	No termination required. An internal pull-up may be used for logic 1. If a higher edge rate is required, pull up to VDD_IO through a minimum 470-ohm resistor.
I2C_SD	I/O, OD, LVTTL, PU	I^2C input and output data bus (bidirectional open drain).	No termination required. An internal pull-up may be used for logic 1. If a higher edge rate is required, pull up to VDD_IO through a minimum 470-ohm resistor.
I2C_MA	I, LVTTL, PU	See I2C_MA in "Power-up Configuration Signals".	
I2C_DISABLE	I, LVTTL, PD	See I2C_DISABLE in "Power-up Configuration Signals".	
I2C_SEL	I, LVTTL, PU	See I2C_SEL in "Power-up Configuration Signals".	
I2C_SA[6:0]	I, LVTTL, PU	See I2C_SA[6:0] in "Power-up Configuration Signals".	
I2C_SLAVE	I, LVTTL, PU	See I2C_SLAVE in "Power-up Configuration Signals".	

18.7 JTAG Signals

Table 104: JTAG Signals

Pin Name	Type	Description	Recommended Termination
BCE	I, LVTTL, PU	Boundary Scan compatibility enabled pin. This input assists with 1149.6 testing. It must be tied to VDD_PCI during normal operation of the device, and during JTAG accesses of the device's registers.	This signal should have the capability to be pulled up or pulled down. • The default setting is to be pulled up. • Pulling the signal down enables the signal analyzer functionality on the SerDes. • A 10K resistor to VDD_PCI should be used.
TCK	I, LVTTL, PD	IEEE 1149.1/1149.6 Test Access Port. Clock input.	If this signal is not used, pull up to VDD_PCI using 10K resistor.
TMS	I, LVTTL, PU	IEEE 1149.1/1149.6 Test Access Port. Test Mode Select	If this signal is not used, pull up to VDD_PCI using a 10K resistor.
TDO	O, LVTTL	IEEE 1149.1/1149.6 Test Access Port. Serial Data Output	No connect if JTAG is not used. Also, pull up to VDD_PCI using a 10K resistor if not used.
TDI	I, LVTTL, PU	IEEE 1149.1/1149.6 Test Access Port. Serial Data Input	If the signal is not used or if a higher edge rate is required, pull up to VDD_PCI through a 10K resistor.
TRST_b	I, LVTTL, PU	IEEE 1149.1/1149.6 Test Access Port (TAP) Reset Input This input must be asserted during the assertion of CHIP_RST_b. Afterwards, it can be left in either state. Combine the CHIP_RST_b and TRST_b signals with an AND gate and use the output to drive the TRST_b pin.	If not used, tie to VSS_PCI using a 10K resistor.

18.8 Interrupt Signals

Table 105: Interrupt Signals

Pin Name	Type	Description	Recommended Termination
INT_b	O, OD, LVTTL	Interrupt signal. For more information on the control of this pin, see "Interrupt Controller".	Pull up to VDD_PCI using a 10K resistor.
RST_IRQ_b	O, OD, LVTTL	Reset notification. This signal is asserted when any Tsi620 interface (PCI or RapidIO) receives a valid reset request, and the interface is configured to request software intervention. For more information on configuring reset requests to assert the RST_IRQ_b signal, see "Resets".	Pull up to VDD_PCI using a 10K resistor.

18.9 GPIO Signals

Table 106: GPIO Signals

Name	Type	Description	Recommended Termination
GPIO[31:0]	3.3 Bidir	General Purpose I/Os. These signals can be configured as input, output, or open-drain under program control. As inputs an interrupt on change of state is provided.	Terminate individual pins to VDD using 10K resistors. Note: Although this is not required, it is recommended to support debug operations.

18.10 Clock and Reset Signals

Table 107: Clock and Reset Signals

Pin Name	Type	Description	Recommended Termination
S_CLK_p	I, CML	Differential reference clock, maximum frequency 156.25 MHz. The clock is used as a reference clock for the SerDes and as a source for the internal logic clocks. These signals are internally terminated. The S_CLK inputs should be AC coupled.	AC coupling capacitor of 0.1uF required.
S_CLK_n	I, CML		

Table 107: Clock and Reset Signals

Pin Name	Type	Description	Recommended Termination
CHIP_RST_b	I, LVTTL, Hyst, PU	Schmidt-triggered reset of all Tsi620 blocks. Asynchronous active-low reset for the Tsi620 (see "Chip Reset").	No termination required.
BLK_RST_b	I, LVTTL, Hyst, PU	Schmidt-triggered reset of all Tsi620 blocks, as controlled by the "Block Reset Control Register". Asynchronous active-low reset for the Tsi620. The action of this reset is controlled through the "Block Reset Control Register".	No termination required.

18.11 Power-up Configuration Signals

Table 108: Power-up Configuration Signals

Pin Name	Pin Type	Description[a]	Recommended Termination[b]
SP_HOST	I, LVTTL, PD	Configures whether or not the SREP is a RapidIO host, or a RapidIO endpoint. This is captured by the HOST bit in the "SREP General Control CSR". It also controls the reset value of destination ID for the SREP. 0 = SREP is an agent or endpoint device 1 = SREP is a RapidIO Host If the HOST bit is 0, the SREP is not a host. The default values for the destination ID are 0xFF/0xFFFF for endpoints that do not support the system boot ROM, and 0xFE/0xFFFE for endpoints that support the system boot ROM. When HOST is 0, I2C_SA[6] initializes the least significant bit of the DEST_ID and LG_DEST_ID fields of the "SREP Base Device ID CSR". Other bits in these fields are 1. If the HOST bit is 1, then the SREP is a host. The least significant 7 bits of the DEST_ID and LG_DEST_ID fields of the "SREP Base Device ID CSR" are determined by the I2C_SA[6:0] bits. Other bits in these fields are 0. This signal must remain stable for 10 RapidIO Reference Clock cycles after CHIP_RST_b is de-asserted in order to be sampled correctly. It is ignored after reset.	If the system host is found on the PCI bus, then pull up this signal to VDD_PCI using a 10K resistor. This configures the SREP RapidIO destination ID using the I2C_SA[6:0] signals.

Table 108: Power-up Configuration Signals (Continued)

Pin Name	Pin Type	Description[a]	Recommended Termination[b]
SP_MAST_EN	I, LVTTL, PD	Configures whether or not the SREP is allowed to issue RapidIO Request packets. This signal configures the reset value of the MAST_EN bit in the "SREP General Control CSR". 0 = SREP cannot issue RapidIO Request transactions 1 = SREP can issue RapidIO Request transactions This signal must remain stable for 10 RapidIO Reference Clock cycles after CHIP_RST_b is de-asserted in order to be sampled correctly. It is ignored after reset.	If the system host is found on the PCI bus, and the system host should be allowed to issue requests immediately after a reset, then pull up to VDD_PCI using a 10K resistor.
SP{n}_MODE_SEL	I/O, LVTTL, PD	Selects the RapidIO operating mode for ports n and n+1 (n = 0, 2, 4). 0 = Port n operating in 4x mode (Port n+1 not available) 1 = Ports n and n+1 operating in 1x mode The output capability of this pin is used only in test mode. This signal must remain stable for 10 RapidIO Reference Clock cycles after CHIP_RST_b is de-asserted in order to be sampled correctly. It is ignored after reset. Note: The "RapidIO SMAC x Digital Loopback and Clock Selection Register" will override and determine the operating mode for the corresponding ports.	Pin must be tied off according to the required configuration using either a 10K pull-up to VDD_PCI or a 10K pull-down to VSS_IO. An internal pull-down may be used for logic 0.
SP6_MODE_SEL	I/O, LVTTL, PD	Selects the FPGA Interface port operating mode. 0 = Port 6 operates in 4x mode 1 = Port 6 operates in 1x mode The output capability of this pin is used only in test mode. This signal must remain stable for 10 RapidIO Reference Clock cycles after CHIP_RST_b is de-asserted in order to be sampled correctly. It is ignored after reset. Note: It is not possible to use the "RapidIO SMAC x Digital Loopback and Clock Selection Register" to override the operating mode for the FPGA Interface port.	Pin must be tied off according to the required configuration using either a 10K pull-up to VDD_PCI or a 10K pull-down to VSS_IO. An internal pull-down may be used for logic 0.

Table 108: Power-up Configuration Signals (Continued)

Pin Name	Pin Type	Description[a]	Recommended Termination[b]
SP_IO_SPEED[1:0]	I/O, LVTTL, [PU, PD]	Serial Port Transmit and Receive operating frequency select. These pins select the power-up frequency for all RapidIO ports. 00 = 1.25 Gbps 01 = 2.5 Gbps 10 = 3.125 Gbps (default) 11 = Illegal The following are applicable to this power-up option: • These signals must remain stable for 10 RapidIO Reference Clock cycles after CHIP_RST_b is de-asserted in order to be sampled correctly. • These signals are ignored after reset; software can override the port frequency setting in the "RapidIO SMAC x Digital Loopback and Clock Selection Register". • The SP_IO_SPEED[1:0] setting is equal to the IO_SPEED field in the "RapidIO SMAC x Digital Loopback and Clock Selection Register". • Clock speeds are achieved with a 125- or 156.25-MHz reference clock. Other reference clock speeds produce different output port speeds. • The output capability of these pins is used only in test mode.	Pin must be tied off according to the required configuration using either a 10K pull-up to VDD_PCI or a 10K pull-down to VSS_IO. For SP_IO_SPEED[1], an internal pull-down can be used for logic 0. For SP_IO_SPEED[0], an internal pull-up can be used for logic 1.
SP_CLK_SEL[1:0]	I/O, LVTTL, [PD, PD]	Reference clock speed 00 = 125-MHz Reference clock (default) 01 = 156.25-MHz Reference clock 10, 11 = Reserved The following notes are applicable to this power-up option: • These signals configure the MPLL settings for the RapidIO SerDes. • These signals must remain stable for 10 RapidIO Reference Clock cycles after CHIP_RST_b is de-asserted in order to be sampled correctly. • These signals are ignored after reset; software can override the port frequency setting in the "RapidIO SMAC x Digital Loopback and Clock Selection Register". • The output capability of this pin is used only in test mode.	Pin must be tied off according to the required configuration using either a 10K pull-up to VDD_PCI or a 10K pull-down to VSS_IO. For SP_CLK_SEL[1:0], an internal pull-down can be used for logic 0.

Table 108: Power-up Configuration Signals (Continued)

Pin Name	Pin Type	Description[a]	Recommended Termination[b]
SP{n}_PWRDN n = 2, 4	I/O, LVTTL, PU	Port n Transmit and Receive Power Down Control. This signal controls the state of Port n. It also controls the power down of Ports n+1. The PWRDN controls the state of all four lanes (A/B/C/D) of the SerDes. 0 = Port n Powered Up. Port n+1 controlled by SP{n+1}_PWRDN. 1 = Port n Powered Down. Port n+1 Powered Down. Override SP{n}_PWRDN using PWDN_x4 field in the "RapidIO SMAC x Digital Loopback and Clock Selection Register". The output capability of this pin is used only in test mode. This signal must remain stable for 10 RapidIO Reference Clock cycles after CHIP_RST_b is de-asserted in order to be sampled correctly. It is ignored after reset. Note: Port 0 must remain powered up at all times. Note: This input exists for the FPGA Interface (n = 2, 4, 6).	Pin must be tied off according to the required configuration using either a 10K pull-up to VDD_PCI or a 10K pull-down to VSS_IO. An internal pull-up may be used for logic 1.
SP{n+1}_PWRDN n=0, 2, 4	I/O, LVTTL, PU	Port Transmit and Receive Power Down Control. This signal controls the state of Port n+1. Note that Port n+1 is never used when 4x mode is selected for a RapidIO port, and it must be powered down. 0 = Port n+1 powered up 1 = Port n+1 powered down Override SP{n+1}_PWRDN using PWDN_x1 field in "RapidIO SMAC x Digital Loopback and Clock Selection Register". The output capability of this pin is only used in test mode. This signal must remain stable for 10 RapidIO Reference Clock cycles after CHIP_RST_b is de-asserted in order to be sampled correctly. It is ignored after reset. Note: Port 0 must remain powered up at all times.	Pin must be tied off according to the required configuration using either a 10K pull-up to VDD_PCI or a 10K pull-down to VSS_IO. An internal pull-up may be used for logic 1.

Table 108: Power-up Configuration Signals (Continued)

Pin Name	Pin Type	Description[a]	Recommended Termination[b]
SP6_PWRDN	I/O, LVTTL, PU	Port 6 Transmit and Receive Power Down Control. This signal controls the state of all four lanes (A/B/C/D) of Port 6, the FPGA Interface. 0 = Port 6 powered up 1 = Port 6 powered down Override SP6_PWRDN using PWDN_x4 field in the "RapidIO SMAC x Digital Loopback and Clock Selection Register". The output capability of this pin is used only in test mode. This signal must remain stable for 10 RapidIO Reference Clock cycles after CHIP_RST_b is de-asserted in order to be sampled correctly. It is ignored after reset.	Pin must be tied off according to the required configuration using either a 10K pull-up to VDD_PCI or a 10K pull-down to VSS_IO. An internal pull-up may be used for logic 1.
SP_RX_SWAP SP_TX_SWAP	I, LVTTL, PD	Configures the order of 4x receive/transmit lanes on RapidIO ports [0,2,4]. These signals do not apply to ports 1, 3, or 5, or the FPGA Interface. 0 = A, B, C, D 1 = D, C, B, A Override SP_RX(TX)_SWAP using RX(TX)_SWAP field in the "RapidIO SMAC x Digital Loopback and Clock Selection Register". These signals must remain stable for 10 RapidIO Reference Clock cycles after CHIP_RST_b is de-asserted in order to be sampled correctly. They are ignored after reset. Note: Ports that require the use of lane swapping will function only as x4 ports. The reconfiguration of a swapped port to dual x1 will result in the inability to connect to a x1 link partner.	No termination required. An internal pull-down can be used for logic 0. Pull up to VDD_PCI through 10K if external pull-up is desired. Pull down to VSS_IO through a 10K resistor if an external pull-down is desired.
I2C_MA	I, LVTTL, PU	I^2C Multibyte Address. When driven high, the I^2C Interface expects multi-byte peripheral addressing; when driven low, single-byte peripheral address is expected. This signal must remain stable for 10 RapidIO Reference Clock cycles after CHIP_RST_b is de-asserted in order to be sampled correctly. It is ignored after reset.	No termination required. An internal pull-up may be used for logic 1. Pull up to VDD_PCI through 10K resistor if an external pull-up is desired. Pull down to VSS_IO to change the logic state.

Table 108: Power-up Configuration Signals (Continued)

Pin Name	Pin Type	Description[a]	Recommended Termination[b]
I2C_SA[6:0]	I, LVTTL, PU	I^2C Slave Address pins. The values on these pins represent the values for the 7-bit address of the Tsi620 when acting as an I^2C slave. These signals, in combination with the SP_HOST signal, determine the destination ID value in "SREP Base Device ID CSR" (see SP_HOST in this table). These signals, in combination with the I2C_SEL signal, determine the address of the EEPROM to boot from (see I2C_SEL in this table). The values on these pins can be overridden after a reset by writing to the "I2C Slave Configuration Register", and the "SREP Base Device ID CSR". These signals must remain stable for 10 RapidIO Reference Clock cycles after CHIP_RST_b is de-asserted in order to be sampled correctly. they are ignored after reset.	No termination required. An internal pull-up may be used for logic 1. Pull up to VDD_PCI through 10K resistor if an external pull-up is desired. Pull down to VSS_IO to change the logic state.
I2C_SLAVE	I, LVTTL, PU	I^2C Slave mode enable. This signal enables the I^2C Interface to respond to the I^2C slave address, and allows an external I^2C master to read and write Tsi620 registers. This pin controls the reset value of the SLV_EN bit in the "I2C Slave Configuration Register". 0 = Tsi620 does not respond to accesses on the I^2C bus 1 = Tsi620 responds to accesses on the I^2C bus This signal must remain stable for 10 RapidIO Reference Clock cycles after CHIP_RST_b is de-asserted in order to be sampled correctly. It is ignored after reset.	No termination required. An internal pull-up may be used for logic 1. Pull up to VDD_PCI through 10K resistor if an external pull-up is desired. Pull down to VSS_IO to change the logic state.
I2C_SEL	I, LVTTL, PU	I^2C Pin Select. Together with the I2C_SA[1,0] pins, the Tsi620 will determine the lower 2 bits of the 7-bit address of the EEPROM address it boots from. When asserted, the I2C_SA[1,0] values also are used as the lower 2 bits of the EEPROM address. When de-asserted, the I2C_SA[1,0] pins are ignored and the lower 2 bits of the EEPROM address are default to 00. The values of the EEPROM address can be overridden by software after reset. This signal must remain stable for 10 RapidIO Reference Clock cycles after CHIP_RST_b is de-asserted in order to be sampled correctly. Ignored after reset.	No termination required. An internal pull-up may be used for logic 1. Pull up to VDD_PCI through 10K resistor if an external pull-up is desired. Pull down to VSS_IO to change the logic state.

Table 108: Power-up Configuration Signals (Continued)

Pin Name	Pin Type	Description[a]	Recommended Termination[b]
I2C_DISABLE	I, LVTTL, PD	Disable I^2C register loading after reset. When asserted, the Tsi620 does not attempt to load register values from the I^2C bus. 0 = Enable I^2C register loading 1 = Disable I^2C register loading This signal must remain stable for 10 RapidIO Reference Clock cycles *before and after* a transition a transition of CHIP_RST_b or BLK_RST_b.	No termination required. Pull up to VDD_PCI through a 10K resistor if I2C loading is not required.
PCI_RSTDIR	PCI In	Reset Direction. This signal determines the direction of PCI initialization. 0 = PCI_RSTn is an input. The PCI Interface detects initialization driven by another PCI device. 1 = PCI_RSTn is an output. The PCI Interface drives initialization. This includes driving the PCI_RSTn signal, and driving the PCI bus clock.	If the Tsi620 is a secondary device on the PCI bus, pull down to VSS. If the Tsi620 is the primary device on the PCI bus, pull up to VDD_PCI.
PCI_M66EN	PCI In	66-MHz Enable. It determines whether the PCI bus is low speed or high speed. 0 = PCI bus operates in the 25-33 MHz range 1 = PCI bus operates in the 50-66 MHz range This signal must be stable and valid around the rising edge of CHIP_RST_b, BLK_RST_b, and PCI_RSTn.	Pull up to VDD_PCI using a 8.2K-ohm resistor.
PCI_PLL_BYPASS	PCI In	PCI PLL Bypass. This signal controls whether or not the PLL used to generate the PCI_CLKO[4:0] signals is bypassed. 0 = PCI PLL is enabled 1 = PCI PLL is bypassed This signal affects the output frequency of the PCI_CLKO[4:0] signals (see "Clock Generation"). This signal must remain stable for 10 RapidIO Reference Clock cycles after CHIP_RST_b is de-asserted in order to be sampled correctly. It is ignored after reset.	If logic 0 is required, pull down to VSS. If logic 1 is required, pull up to VDD_PCI.

Table 108: Power-up Configuration Signals (Continued)

Pin Name	Pin Type	Description[a]	Recommended Termination[b]
PCI_ARBEN	PCI In	PCI arbiter enable. This pin controls whether or not the Tsi620 PCI arbiter is used (see "Bus Arbitration"). 0 = PCI arbiter is not used. The PCI Interface requests bus ownership using PCI_REQ1, and is granted ownership by PCI_GNT1. 1 = PCI arbiter is used. PCI_REQn signals are used by external PCI masters to request bus ownership, and PCI_GNTn signals are used by the Tsi620 PCI arbiter to grand bus ownership to external PCI masters. This signal must remain stable for 10 RapidIO Reference Clock cycles after CHIP_RST_b is de-asserted in order to be sampled correctly. Ignored after reset.	If logic 0 is required, pull down to VSS. If logic 1 is required, pull up to VDD_PCI.
PCI_HOLD_BOOT	PCI In	This pin controls whether the PCI Interface responds immediately after a reset, or is released by software control from the RapidIO Interface. 0 = Release PCI reset immediately after Tsi620 reset is completed 1 = Hold PCI Interface in reset until software clears the SOFT_RESET bit in the "PCI Miscellaneous Control and Status Register". This signal must remain stable for 10 RapidIO Reference Clock cycles after CHIP_RST_b is de-asserted in order to be sampled correctly. Ignored after reset.	If logic 0 is required, pull down to VSS. If logic 1 is required, pull up to VDD_PCI.

a. Power-up signals are only latched when the CHIP_RST_b pin is asserted.

b. Even if internal termination is used for Tsi620's power-up configuration signals, IDT recommends that for debug reasons provision be made to pull the signals in the opposite direction from the internal termination.

18.12 Power Supply Signals

Table 109: Power Supply Signals

Pin Name	Type	Description	Decoupling Requirements
PCI_PLL_AVDD	-	PCI PLL analog power	a
PCI_PLL_AVSS	-	PCI PLL analog ground	a
CLKGEN_PLL_ AVDD	-	PCI Clock Generator analog power	a
CLKGEN_PLL_ AVSS	-	PCI Clock Generator analog ground	a
COMMON SUPPLY			
VDD_PCI	-	PCI I/O and LVTTL I/O power	a
VDD_HSTL	-	FPGA Interface I/O power	a
SP6_VREF	-	Reference power supply for differential stage of input receiver. (VRF 1.8V option). This is required to be a low-noise supply.	a
VSS_IO	-	Common ground returns for I/Os	a
VDD	-	Common 1.2V supply for core digital logic	a
VSS	-	Common ground returns for core digital logic	a
SP_AVDD	-	Analog power supply for RapidIO ports	a
SP_VDD	-	Digital logic power supply for RapidIO ports	a

a. This information will be available in a future version of this document.

19. Clock, Reset, Power-up, and Initialization Options

Topics discussed include the following:

- “Clocks”
- “Resets”
- “Power-up”
- “Initialization”

19.1 Clocks

The Tsi620 has three input clock signals (see also Table 110):

- A differential clock input for RapidIO, S_CLK_p/n
- A single-ended clock input for PCI, PCI_CLK
- A single-ended clock input for the FPGA Interface, SP6_RXCLK

Table 110: Input Clock Signals

Clock Signal	Frequency	Duty Cycle		Jitter	Skew
		Minimum	Maximum		
S_CLK_p/n	125 - 156.25 MHz +/- 100 PPM	40%	60%	0-320 ps	3 ps RMS
PCI_CLK	25 - 66MHz	40%	60%	+/- 100 ps	N/A
SP6_RXCLK	62.5, 125, or 156.25 MHz +/- 100 PPM	40%	60%	175 ps (peak to peak)	N/A

The Tsi620 has three output clock signals (see also Table 110):

- Five PCI clock outputs, PCI_CLKO[0:4]
- A single-ended clock output for the FPGA Interface, SP6_TXCLK
- An I^2C clock signal, I2C_SCLK

Table 111: Output Clock Signals

Clock Signal	Frequency	Duty Cycle		Jitter	Skew
		Minimum	Maximum		
PCI_CLKO[0:4]	33 - 66MHz	40%	60%	+/- 100 ps	58 ps
SP6_TXCLK	62.5, 125, or 156.25 MHz +/- 100 PPM	40%	60%	175 ps (peak to peak)	N/A
I2C_SCLK	<100 KHz	40%	60%		N/A

The Tsi620 requires a RapidIO Reference Clock, S_CLK_p/n. A RapidIO Reference clock frequency of either 125 or 156.25 MHz supports standard RapidIO lane frequencies. Other reference clock frequencies can support lane rates for other standards (see "Support for Non-standard Baud Rates").

The RapidIO Reference Clock drives the Switch Ports, the Switch ISF, the Multicast Engine, the SREP, and the Bridge ISF. The RapidIO Reference Clock, divided by 2, drives the register bus, the Switch Utility block, and the I^2C block. The Switch Port clock logic is initialized to use either a 125 or 156.25 MHz reference clock based on the setting of the SP_CLK_SEL input pin (see "RapidIO Signals").

The RapidIO Reference Clock is also used as the input to an SSPLL, whose output frequency drives the five PCI_CLKO pins. When the PCI_RSTDIR pin is 0, indicating that the Tsi620 drives the PCI bus clock and reset signals, the SSPLL divides 125 MHz or 156.25 MHz reference clock frequencies down to 33.33 MHz or 66.66 MHz. The PCI_M66EN input signal determines whether 33.33 MHz or 66.66 MHz is chosen. One of PCI_CLKO pins can be connected on the board to the PCI_CLK input pin for the PCI de-skew PLL. The PCI de-skew PLL drives the PCI block clock tree.

When the PCI_RSTDIR pin is 1, indicating that the Tsi620 does not drive the PCI bus clock and reset signals, the SSPLL is bypassed. The SSPLL is also bypassed if the PCI block is powered down using the PWRDWN field of the "Clock Generator PLL0 Control Register 0", or if the Internal Switch Port and the SREP are powered down at the same time (see "Bridge Shutdown"). The frequencies output on the PCI_CLKO pins when PCI_RSTDIR is 1 or PWRDWN is 1 (see Table 112).

Note that when the SSPLL is bypassed, the clock divider logic divides the reference clock frequency by a factor of 2, 4, or 6 to drive the PCI_CLKO pins, depending on the settings of the PCI_M66EN and SP_IO_SPEED[1-0] pins.

Table 112: PCI_CLKO Output Frequencies

PCI_RSTDIR == 0 OR PWRDWN == 1 (SSPLL Bypassed)	RapidIO Reference Clock Frequency (MHz)	PCI_M66EN Signal	PCI_CLKO Output Frequency (MHz)
0	125	0	33.33
0	125	1	66.66
0	156.25	0	33.33
0	156.25	1	66.66
1	125	0	31.25
1	125	1	62.5
1	156.25	0	26.04
1	156.25	1	39.06

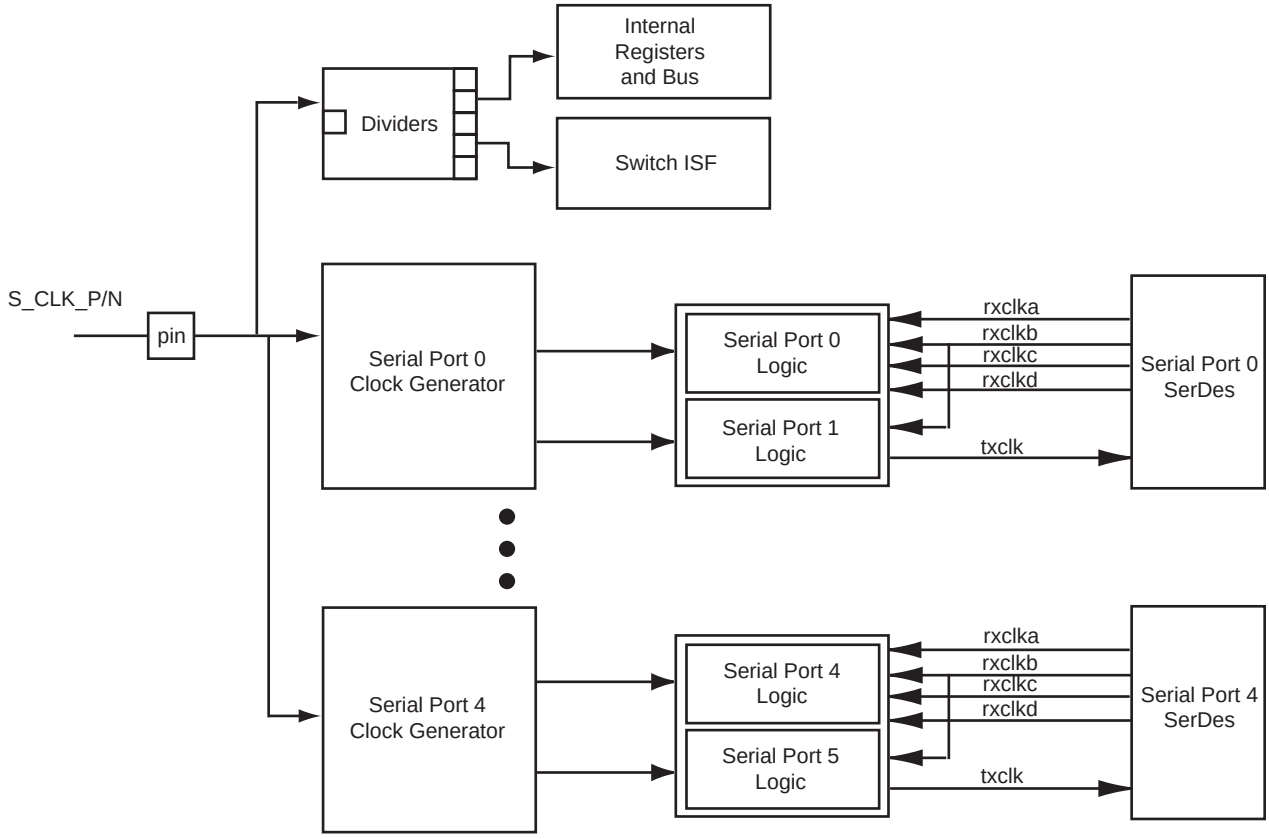
The FPGA Interface accepts an input clock as part of its receive interface. This drives the portion of the FPGA Interface receive path up to the clock compensation buffer. All FPGA Interface logic outside of this portion of the receive path is driven by a clock provided by the DPLL associated with the FPGA Interface. The FPGA Interface's DPLL is driven off of the RapidIO reference clock (see "Clocking").

The register bus and the I^2C block are driven from the RapidIO Reference Clock. The register bus and the I^2C block receive a clock signal that is half the frequency of the RapidIO Reference Clock. The I^2C block clock settings are initialized based a 156.25 MHz reference clock to allow the I^2C Interface to operate at ~100 KHz. If the reference clock is different from 156.25 MHz, the I^2C bus operates at a frequency that is slower than 100 KHz.

In addition to the two reference clocks, each RapidIO ingress port contains independent receive clock domains, one for each lane. The receive clock is extracted from the 8B/10B encoding on each lane.

19.1.1 SMAC Clock Architecture

Figure 50: SMAC Clock Architecture



The SMAC relies on the reference clock, S_CLK_P/N, to generate most clocks inside the SMAC. This reference clock is connected to each SerDes. The SerDes outputs a clock signal that drives all logic within the MAC, except for a small portion of the receive path. On the Receive side, each SerDes recovers clocks from the data stream. This clock drives the logic that consumes the clock compensation sequence, which handles clock frequency mismatches between the transmitter and receiver. In x4 mode, 4 different synchronous clocks are extracted (rxclka..d). In x1 mode, either one (rxclka) or two (rxclka..b) clocks are recovered.

On the Transmit side, the SerDes uses the reference clock signal (S_CLK_p/n) as the reference frequency to derive tx_clk. An extra clock (sys_clk) is also sourced from the SerDes to the MAC.

19.1.2 SerDes Clocks

The RapidIO SerDes in the Tsi620 uses the same external reference clock, S_CLK_P/N. Depending on the pin or register setup, the SerDes generates the appropriate clocks for serializing/deserializing the data as well as the clocks for the internal logic. On the Receive side, each lane of the SerDes recovers its own clocks. These clocks can be powered down using the “RapidIO SMAC x Digital Loopback and Clock Selection Register”.

19.1.3 Reference Clocks

The two reference clocks are described in Table 113.

Table 113: Input Reference Clocks

Clock Input Pin	Type	Frequency (MHz)[a]	Clock Domains[b]
S_CLK_P/N	Differential	122.88, 125, 153.6 156.25	Reference clock for all Switch logic (SMAC, Switch ISF, Multicast Engine), SREP, Bridge ISF, I^2C Interface, and register bus Masters and Slaves. Standard RapidIO lane rates are supported with 125 and 156.25 MHz reference clocks.
PCI_CLK	Single ended	25–66	Reference clock for the PCI Interface when the Interface is the Primary device on the PCI bus.

a. For electrical characteristics information, see "Electrical Characteristics".

b. For information on configuring the clock rate of RapidIO ports, see "Clocking".

19.1.4 Clock Domains

The Tsi620 contains eight clock domains that are generated from the two input reference clocks. These domains are explained in Table 114.

Table 114: Clock Domains

Clock Domain	Clock Source	Frequency (MHz)[a]	Description
Internal Register Domain	S_CLK_P/N Divided by 2	61.44, 62.5, 76.8, 78.125	This clock domain includes the register bus, and the register bus master and slave sub-blocks within each functional block. Registers for each block usually use the same clock as that block. The domain uses the input S_CLK_P/N divided by 2.
PCI Clock Domain	S_CLK_P/N with SSPLL, And/Or PCI_CLK input	25–66	This clock domain includes the portion of the PCI Interface that translates PCI transactions to and from Bridge ISF transactions.
ISF Clock Domain (applies to Switch ISF and Bridge ISF)	S_CLK_P/N	122.88, 125, 153.6, 156.25	This clock domain includes the Switch ISF, the Multicast Engine, the Internal Switch Port, the SREP, Bridge ISF and the portion of each block that communicates with the Switch ISF or the Bridge ISF. The domain uses the S_CLK_P/N.
FPGA Interface Receive Domain	Pin on receive interface	62.5, 125, or 156.25	For more information, see "Clocking".
FPGA Interface Transmit Domain	S_CLK_P/N		

Table 114: Clock Domains (Continued)

Clock Domain	Clock Source	Frequency (MHz)[a]	Description
Serial Receive Domain[b]	Individual RapidIO Received lanes	125/250/312.5	These clocks are the recovered clocks from the SerDes. Each lane will have its own recovered clocks. Its rates depend on the data baud rates. Frequencies quoted are for standard RapidIO line rates. Non-standard reference clock frequencies result in different frequencies for the Serial Receive Domain (see "Support for Non-standard Baud Rates").
Serial Transmit Domain	S_CLK_P/N	125/250/312.5	This clock domain clocks all the RapidIO transmit ports. The S_CLK_P/N input clocks the transmit logic. This clock generates the high-speed clock that outputs the serial data on output pins, SP{0..5}_T{A..D}_P/N. The maximum data rate for this domain is 3.125 Gbps per lane. The maximum data rate requires a 125 or 156.25 MHz SL_CLK_P/N reference clock.
I^2C Domain	S_CLK_P/N divided by 2, Further divided by 782	100 kHz	This clock domain drives the I^2C output clock pin, I2C_SCLK. This domain is generated by dividing the S_CLK_P/N input by a programmable value (see "I2C Interface"). The majority of the I^2C logic runs in the Internal Register Domain. Division by 782 produces the 100 kHz frequency when a 156.25 MHz RapidIO reference clock is supplied. Slower reference clocks require adjustment of the I^2C clock divider registers to achieve 100 kHz operation.

a. For electrical characteristics information, see "Electrical Characteristics".

b. This is also known as SYS_CLK in the document.

19.1.5 Clock Gating

When a RapidIO port is powered down using the PWDN_X1/X4 bits in the "RapidIO SMAC x Digital Loopback and Clock Selection Register", the clock to that RapidIO port is gated to prevent the port from consuming power.

When the Tsi620 Bridge is powered down, the clock to the PCI, Bridge ISF, and SREP is gated to prevent the Tsi620 Bridge from consuming power (see "Bridge Shutdown").

19.2 Resets

The Tsi620 supports six types of resets:

- "Chip Reset"
- "Block Reset"
- "Switch Reset"
- "Bridge Reset"

- "I2C Boot"
- "JTAG Reset"

The Tsi620 can be completely reset using either the CHIP_RST_b or BLK_RST_b pins. CHIP_RST_b resets all Tsi620 interfaces and clears all sticky bits in the interfaces. BLK_RST_b resets specific blocks as controlled by the "Block Reset Control Register". For more information on the operation of these signals, see "Chip Reset" and "Block Reset".

To facilitate a software-controlled shutdown before performing the reset of the hardware, the Tsi620 allows all reset inputs, except CHIP_RST_b, to be mapped to an interrupt output, RST_IRQ_b. RST_IRQ_b can be routed to the appropriate processor to interrupt software. The software then performs any operations necessary to make the reset safe. After the software prepares for a safe reset, the software entity must perform a reset of the appropriate Tsi620 blocks using the "Block Reset Control Register".

If RST_IRQ_b must be routed to either PCI INTA or to a RapidIO port-write, RST_IRQ_b can be connected to a GPIO signal configured as an interrupt input. For more information on configuring GPIO signals, see "GPIO as Event Sources".

GPIO interrupt inputs can then be routed to PCI INTA, INT_b, or a RapidIO port-write using the "Event Routing Register 2".

To prevent possible corruption of devices on the I^2C bus, it is recommended that software wait for an I^2C operation to complete before completing reset (see "Tsi620 as I2C Master").

Internal logic must automatically sequence the removal of reset in all internal blocks to meet their requirements; no additional software programming is required.

If a Chip, Block, Switch, or Bridge reset is in progress, then subsequent Block, Switch, or Bridge reset requests are ignored. Chip reset requests from the DO_RESET bit in the "Block Reset Control Register" are also ignored until the current reset is completed.

The assertion of the CHIP_RST_b pin causes a Chip Reset, regardless of the current reset state.

19.2.1 Chip Reset

A chip reset resets all Tsi620 blocks except the JTAG block. This type of reset is equivalent to a power-up reset. Sticky bits are cleared, and all configuration registers are reset to their original power-on states.

A chip reset is triggered by one of the following:

- CHIP_RST_b is asserted
- BLK_RST_b input pin is asserted when the CHIP_RESET bit is 1 in the "Block Reset Control Register"
- DO_RESET bit is set to 1 in the "Block Reset Control Register" when the CHIP_RESET bit is set to 1 in the same register

The operation of the device is controlled by the power-up configuration pins (see "Power-up").

The MCES signal is reset only on assertion of CHIP_RST_b. The signal is not reset by any other reset.

An I^2C Boot occurs after a chip reset when the I2C_DISABLE pin has a value of 0.

Lookup tables in the Switch Ports, and Lookup table entries in the SREP and PCI block, are left in an undefined state after reset. It is recommended that all Lookup tables and Lookup Table entries be completely initialized after a chip reset to ensure deterministic operation.

Power-up configuration signals are only latched when the CHIP_RST_b signal is asserted.

19.2.2 Block Reset

A block reset resets all Tsi620 blocks according to the settings in the "Block Reset Control Register". This type of reset is equivalent to the simultaneous execution of a "Switch Reset" and a "Bridge Reset".

A block reset is triggered by one of the following:

- BLK_RST_b input pin is asserted when CHIP_RESET bit is 0 in the "Block Reset Control Register"
- DO_RESET bit is set to 1 in the "Block Reset Control Register" when the CHIP_RESET bit is set to 0 in the same register

The default settings in the "Block Reset Control Register" cause all blocks in the Tsi620 to be reset on assertion of BLK_RST_b, followed by an I^2C Boot. By default, sticky bits are preserved by a block reset.

All packets in flight between the Internal Switch Port and the SREP are discarded when a block reset occurs and either the SRIO or SREP bits are set in the "Block Reset Control Register".

Note that the RapidIO Reference Clock must remain stable before, during, and after the assertion of BLK_RST_b, since the Tsi620 PLLs are not reset by a block reset.

An I^2C Boot occurs after a block reset when the I2C bit and the I2C_BOOT bits are set in the "Block Reset Control Register".

The I2C_BOOT bit must only be set if all blocks are reset by a block reset. The SREP, SRIO, PCI, BISF, SISF, and I2C bits must all be set to 1 in the "Block Reset Control Register" if the I2C_BOOT bit is set.

If the I2C_DISABLE pin is set, no register values are loaded from I2C, regardless of the status of the I2C_BOOT bit in the "Block Reset Control Register".

19.2.3 Switch Reset

The Tsi620 Switch includes all RapidIO ports, the Switch ISF, the Multicast Engine, the Internal Switch Port, and the Switch Utility block. The blocks within the Tsi620 Switch that are reset is controlled by the "Block Reset Control Register". A switch reset is triggered by one of the following:

- A valid reset request, which consists of four Reset Request Control Symbols, is received on any of the RapidIO ports [RapidIO ports 0-5, the FPGA Interface (port 6), or the Internal Switch Port (port 8)]
- A bridge reset is propagated to a switch reset

If the SELF_RST bit in the "RapidIO Port x Mode CSR" is set to 0 and a reset request is received on that port, the behavior of the port depends on the setting of the RCS_INT_EN bit in the same register.

- If RCS_INT_EN is set to 0, the reset request is ignored.
- If RCS_INT_EN is 1, then the RST_IRQ_b interrupt is asserted. No reset occurs. Software must prepare the system for a reset, and then generate that reset using the "Block Reset Control Register". For more information on how resets are handled by software, see "Software Reset of Switch and Bridge".

The use of the SELF_RST and RCS_INT_EN bits in the "RapidIO Port x Mode CSR" allows system designers to control which RapidIO ports, if any, are legitimate sources of a switch reset.

If the SELF_RST bit in the "RapidIO Port x Mode CSR" is set to 1 and a reset request is received on that port, then:

- All RapidIO ports, the Multicast Engine, the Switch ISF, and the I^2C Interface are reset according to the settings of the bits in the "Block Reset Control Register".

The SISF and SRIO bits in the "Block Reset Control Register" must be set to 1 to correctly handle a switch reset request.

All packets in flight between the Internal Switch Port and the SREP are discarded when a switch reset occurs.

- The RST_IRQ_b signal remains asserted for at least four RapidIO Reference clock cycles.
- If the PGTSW bit is set to 1 in the "Block Reset Control Register", then the SREP, the PCI Interface, and the Bridge ISF are reset according to the bit settings in the "Block Reset Control Register".
- An I^2C boot occurs if the I2C_BOOT bit is set in the "Block Reset Control Register". All state machines and the configuration registers are reset to their original power-on states.

The I2C_BOOT bit must only be set if all blocks are reset by the switch reset. The SREP, SRIO, PCI, BISF, SISF, PGTSW and I2C bits must all be set to 1 in the "Block Reset Control Register" if the I2C_BOOT bit is set.

If the I2C_DISABLE pin is set, no register values are loaded from I2C, regardless of the status of the I2C_BOOT bit in the "Block Reset Control Register".

Lookup tables in all ports are left in an undefined state after a switch reset. It is recommended that lookup tables be completely initialized after a reset to ensure deterministic operation.

If the PGTSW bit is set to 1 in the "Block Reset Control Register", then the LUTs located in the SREP and in the PCI Interface must be initialized.

19.2.3.1 Port Resets

Each Tsi620 RapidIO port can be independently reset, including the Internal Switch Port (Port 8). The ports can be reset using one of the following methods:

- Setting the SOFT_RST_X1/X4 bit in the "RapidIO SMAC x Digital Loopback and Clock Selection Register"
- Setting the PWDN_X1/X4 bit in the same register

Note that the Internal Switch Port (Port 8) only supports the PWDN_X4 bit. The port power-down feature is described in "Port Power Down".

After a port is reset, the contents of the ports configuration registers revert to reset default values, SerDes related registers and the LUT for the reset port must be re-configured; register values are not loaded from I^2C on a port reset.

19.2.3.2 Generating a RapidIO Reset Request to a Peer Device

Software can reset a peer device using the following steps:

1. Determine which RapidIO port is connected to the peer to be reset.
2. Change the LUT contents of the RapidIO port to ensure that no packets are being routed to the link partner that is to be reset.
3. Lock out the port using PORT_LOCKOUT in the "RapidIO Serial Port x Control CSR". This ensures that any traffic received from the peer device is dropped, and any traffic still in flight to the peer device is dropped.

   The ackIDs expected by each link partner are out of synchronization because the reset has set the next ackID expected and transmitted by the link partner to be 0. No packets other than the maintenance write in step 8 can be sent on the link before step 8 is completed; otherwise, a fatal link error due to an ackID mismatch results.

4. Use the "RapidIO Serial Port x Link Maintenance Request CSR" to transmit four reset control symbols in a row.
5. Write 0 to the OUTBOUND field of the "RapidIO Serial Port x Local ackID Status CSR".

6. Unlock the port using the PORT_LOCKOUT field within the "RapidIO Serial Port x Control CSR".

   The Tsi620 port will be in input-error stopped state due to errors caused by the loss of lane synchronization (LOLS). The response packet for the maintenance write packet in step 6 is therefore not acknowledged immediately by the Tsi620. The link partner will time out waiting for a packet acknowledge control symbol, and will enter output-error stopped state. To recover, the link partner sends a link-request/input-status control symbol to the Tsi620 port. This clears the input-error stopped state on the Tsi620. The Tsi620 responds with a link-response/status control symbol, which the link partner accepts and exits the output-error stopped state. The maintenance response packet is then retransmitted and accepted by the Tsi620.

   The link-response timer value in the link partner must be configured to a short value after reset through hardware dependent means. If this is not completed, the Tsi620 recovery from input-error stopped state is delayed, and the step 6 maintenance write may time out at the originating entity.

7. Perform a maintenance write to the OUTBOUND field of the "RapidIO Serial Port x Local ackID Status CSR" of the peer device.

19.2.3.3 Software Reset of Switch and Bridge

Software can reset the switch or the bridge portion of the Tsi620 using the "Block Reset Control Register".

Specific blocks cannot be reset when the CHIP_RESET bit in the "Block Reset Control Register" is set.

Lookup tables in the RapidIO ports and Lookup table entries in the SREP and PCI Interface are left in an undefined state after these blocks are reset. It is recommended that all Lookup tables and Lookup table entries be completely initialized after a reset to ensure deterministic operation.

To perform a software reset of the Tsi620 Switch:

1. Set the SRIO and SISF bits to 1 in the "Block Reset Control Register".

   If the RapidIO ports must be reset, then the SISF and SRIO bits in the "Block Reset Control Register" must be set to 1. It is a programming error to set only one of the SISF and SRIO bits.

   The I2C and I2C_BOOT bits can also be set depending on the need to initialize Tsi620 Switch register values from I2C.

2. Set the PGTBR, PBTSW, SREP, PCI, and Bridge ISF bits to 0 to avoid resetting the Tsi620 Bridge.
3. Set the DO_RESET bit to trigger the reset.

To perform a software reset of the Tsi620 Bridge:

1. Set the SREP, PCI, and Bridge ISF bits to 1 in the "Block Reset Control Register".

   The operation of the Tsi620 is not defined when any of the SREP, PCI, or Bridge ISF blocks is not reset.

   The I2C and I2C_BOOT bits can also be set depending on the need to initialize Tsi620 Switch register values from I2C.

   If the I2C_DISABLE pin is set, no register values are loaded from I2C, regardless of the status of the I2C_BOOT bit in the "Block Reset Control Register".

2. Set the PGTBR, PBTSW, SRIO, and SISF bits to 0 to avoid resetting the Tsi620 Switch.
3. Set the DO_RESET bit to trigger the reset.

   If the I2C_BOOT bit is set, then all registers that are initialized during an I2C Boot are initialized, including registers in blocks that were not reset.

   When the Tsi620 Bridge is reset using software control from the PCI Interface, the Tsi620 does not respond to requests until the reset is completed.

   When the Tsi620 Switch is reset under software control from a RapidIO port, the RapidIO links are unavailable until the reset is completed.

19.2.4 Bridge Reset

The Tsi620 Bridge includes the SREP, the Bridge ISF, and the PCI Interface. A bridge reset is triggered by one of the following:

- A valid reset request is received by the SREP
- The PCI_RSTn signal is asserted on the PCI Interface — PCI_RSTn is an input — when the PCI Interface is a secondary device on the PCI bus
- A switch reset is propagated to a bridge reset

The Tsi620 Bridge must be reset as a unit.

An interrupt can be generated instead of a reset when a valid reset request is received by the SREP, or when the PCI_RSTn signal is asserted. If the SREP receives a reset request from the Internal Switch Port, and the SELF_RST bit in the "SREP Mode CSR" is set to 0, then RST_IRQ_b is asserted. This causes the SREP_RESET_RX bit to be set in the "Block Reset Control Register" (see "Reset Control Symbol Processing").

Similarly, if the PCI_RSTn signal is asserted and the PCI_SELF_RST bit in the "Block Reset Control Register" is 0, then RST_IRQ_b is asserted. This causes the PCI_RESET_RX bit to be set in the "Block Event Status Register". Software can handle the interrupt and then generate a reset (see "Software Reset of Switch and Bridge").

On the other hand, if a reset is to be generated immediately, (either the SELF_RST bit in the "SREP Mode CSR" is set to 1, or the PCI_SELF_RST bit in the "Block Reset Control Register" is set to 1), and a reset is requested, then:

- The RST_IRQ_b signal remains asserted for at least four RapidIO Reference clock cycles.
- The SREP, the PCI block, and the Bridge ISF are reset according to the bit settings in the "Block Reset Control Register".

All packets in flight between the Internal Switch Port and the SREP are discarded when a bridge reset occurs.

The SREP, PCI, and BISF bits must all be set to 1 in the "Block Reset Control Register" to correctly handle a bridge reset request.

- If the PGTBR bit is set in the "Block Reset Control Register", then a switch reset occurs. For more information on the handling of switch resets, see "Switch Reset".
- An I^2C boot occurs if the I2C_BOOT bit is set in the "Block Reset Control Register". All configuration registers are reset to their original power-on states.

The I2C_BOOT bit must only be set if all blocks are reset by the bridge reset. The SREP, SRIO, PCI, BISF, SISF, PGTBR and I2C bits must all be set to 1 in the "Block Reset Control Register" if the I2C_BOOT bit is set.

If the I2C_DISABLE pin is set, no register values are loaded from I^2C regardless of the status of the I2C_BOOT bit in the "Block Reset Control Register".

Lookup tables in SREP and in the PCI Interface are left in an undefined state after a bridge reset. It is recommended that lookup tables be completely initialized after a reset to ensure deterministic operation.

If the PGTBR bit is set to 1 in the "Block Reset Control Register", then the LUTs located in all ports must be initialized.

19.2.5 I^2C Boot

If the I^2C Interface is reset, the I^2C Boot capability can initialize the Tsi620's register values. When all blocks are taken out of reset, the I^2C Interface performs automatic reads from an external EEPROM device to load the initial configuration of the Tsi620 (see "I2C Interface").

The I^2C Interface can only be reset if all other blocks are reset. It is a programming error to reset the I^2C Interface when any other block is not reset.

External I^2C devices are not reset by the Tsi620, so the I^2C bus could be left in an undefined state if the Tsi620 is reset during initial configuration. It is recommended that resets of the Tsi620 occur at a rate that ensures that register loading from an I^2C device has completed before another reset is issued.

Alternatively, software can ensure that all I^2C operations are completed before another reset is issued (see "Tsi620 as I2C Master").

19.2.6 JTAG Reset

The JTAG TAP Controller's reset is independent of the Tsi620 functional resets. The TAP Controller can be reset using the TRST_b signal, or by holding the TMS signal asserted for more than five TCK cycles.

To ensure predictable operation of the Tsi620, for power-up reset, CHIP_RESET and TRST_b must be asserted at the same time. After power-up, the TAP Controller can be reset at any time since it will not affect the Tsi620's operation.

Normal functional reset is still required to reset the device's internal registers. Reset of the Tsi620 does not reset the TAP Controller.

The TAP Controller must be reset on power-up, whether or not it is going to be used, to ensure predictable operation of the Tsi620.

19.2.7 Recommended System Designs

The Tsi620 reset scheme supports three general system designs: PCI-based control, RapidIO-based control, and PCI/RapidIO shared control. These designs are explained in the following sections.

19.2.7.1 Control Functionality on PCI

The most common system design places the system control functions on the PCI bus. In this case, the PCI-based control functions manage the reset of the Tsi620. When the system control functionality resides on the PCI bus, then the following conditions should hold:

- The Tsi620 is a secondary device on the PCI bus, and therefore PCI_RSTn is an input signal for the Tsi620.
- Resets from RapidIO should be ignored.

In this case, PCI_RSTn, CHIP_RST_b, and/or BLOCK_RST_b can drive the reset of the Tsi620. RapidIO reset requests should be disabled (see "Switch Reset" and "Bridge Reset"). The bridge reset, as triggered by the PCI_RSTn signal, should be configured to propagate to a switch reset by setting the PGTBR bit in the "Block Reset Control Register". The SREP, PCI, and Bridge ISF should all be set to ensure that the Tsi620 Bridge is reset correctly.

19.2.7.2 Control Functionality on RapidIO

A similar design to the PCI system places the system control functions on a RapidIO port. In this case, the RapidIO-based control functions manage the reset of the Tsi620. When the system control functionality resides on a RapidIO port, then the following conditions should hold:

- The Tsi620 is the primary device on the PCI bus, and therefore PCI_RSTn is an output signal for the Tsi620.
- Resets from RapidIO should be accepted only on the port(s) that the system control functionality is connected to.

In this case, RapidIO reset requests should be disabled (see "Switch Reset" and "Bridge Reset") for those RapidIO ports that should not be able to cause a reset. The receipt of a RapidIO reset request on ports that the system control functionality is connected to should cause a Switch Reset. The switch reset should be configured to propagate to a bridge reset by setting the PGTSW bit in the "Block Reset Control Register". The SREP, PCI, and Bridge ISF bits should all be set to ensure that the Tsi620 Bridge is reset correctly. This causes the PCI Interface to drive the PCI_RSTn signal, causing a reset of all devices on the PCI bus.

19.2.7.3 Control Functionality Shared Between RapidIO and PCI

When system control functionality is shared between entities residing on PCI and RapidIO, it is recommended that the PCI entity control the bridge reset and the RapidIO entity control the switch reset. The PCI entity is the primary device on the PCI bus, and can generate resets by asserting the PCI_RSTn signal. These resets are handled as a bridge reset. The bridge reset is not propagated to the Tsi620 Switch.

Similarly, the RapidIO entity can cause a reset of the Tsi620 Switch by sending a RapidIO reset request to any RapidIO port on the Tsi620. This reset request should be handled as a switch reset, which resets all RapidIO ports. The switch reset is not propagated to the Tsi620 Bridge.

After either a bridge or switch reset, half of the Tsi620 must be reprogrammed before it can accept traffic. The following options can prevent traffic from being handled by uninitialized hardware depending on the requirements of the system.

- Set the PORT_LOCKOUT bit on both sides of the Internal Switch Port/SREP link. This causes all Bridge-Switch traffic to be retained until the bit is cleared. Congestion can occur in the system, however, and errors may be seen.
- After a reset is issued, set the PORT_DIS bit on both sides of the link. This causes all Bridge-Switch traffic to be discarded until the PORT_DIS bit is cleared. When traffic is discarded, errors may be seen.
- Resets can be coordinated between the PCI and RapidIO entities by allowing reset requests sent through the Internal Switch Port/SREP to cause interrupts to be generated. The receipt of a reset request would indicate that the other side is about to initiate a reset, and to prepare for it. A number of registers, including the "RapidIO Host Base Device ID Lock CSR", "SREP Host Base Device ID Lock CSR", and the "SREP Scratch n Register", can implement a deterministic handshaking algorithm.
- Permit the PCI and RapidIO entities to reset the entire board, and therefore reset all of Tsi620 through the CHIP_RST_b or BLK_RST_b signals.

19.3 Power-up

The Tsi620 Switch has the following types of power-up option pins: default port speed (SP_IO_SPEED[1-0]), port power-up and power-down (SPn_PWRDN), and mode selection (SPn_MODE_SEL). An additional power-up option is the SP_CLK_SEL indication for the RapidIO Reference Clock. These power-up options are discussed in the following sections.

The power-up option pins must be stable for 10 RapidIO Reference clock cycles after CHIP_RST_b is de-asserted.

19.3.1 Power-up Option Signals

Power-up options are latched at reset for initializing the Tsi620. The power-up option pins are listed in Table 108. All power-up option pins have to remain stable for 10 RapidIO Reference Clock cycles after CHIP_RST_b is de-asserted in order to be sampled correctly. These signals are ignored after reset and software can override the settings.

The power-up signals do have internal pull ups or pull downs, however external resistors are recommended on all power-up option signals. For more information on the internal pull-up/pull-down settings for the Tsi620's power-up signals, see "Power-up Configuration Signals".

The latched values of the power-up option pins are available in the "Clock Generator RapidIO Power-up Status Register" and the "Clock Generator Tsi620 Power-up Status Register".

19.3.2 Default Port Speed

When the SP_IO_SPEED[1,0] pins are left unconnected in the board, the device internal pull-up configures the Tsi620 to the maximum speed (3.125 Gbaud). The speed can be overridden by the IO_SPEED field in the "RapidIO SMAC x Digital Loopback and Clock Selection Register").

It is strongly recommended to drive the SP_IO_SPEED[1,0] with known values instead of relying on the internal default values in order to set the default speeds of the device.

19.3.3 Port Power-up and Power-down

The power-up and power-down is overridden by the PWDN_X1 and PWDN_X4 fields in the "RapidIO SMAC x Digital Loopback and Clock Selection Register".

19.4 Initialization

The Tsi620 can be initialized from the RapidIO ports or from the PCI Interface.

19.4.1 Initializing the Tsi620 from RapidIO

To initialize the Tsi620 from the RapidIO Interface, the power-up options must be selected to allow communication on the port to which the RapidIO system host is attached. The software will explore and initialize the Tsi620 Switch using RapidIO Maintenance transactions, initializing lookup tables and other error reporting functions. The objective is to route RapidIO packets from the RapidIO ports to the Internal Switch Port, and then on to the SREP.

Once packets can be routed to the SREP, the SREP can be configured using maintenance transactions. An early part of this configuration may be to allow memory-mapped access to all Tsi620 registers. This allows the PCI Interface of the Tsi620 to be initialized.

For information on how to memory map registers on the SREP, see “Overview of Device Register Map”.

Some system designs require that the PCI Interface be locked out until completion of discovery/initialization of the Tsi620 Switch and SREP. The PCI Interface retries all Configuration Type 1 and Memory accesses if the LOCKOUT bit is set in the “PCI Miscellaneous Control and Status Register”. This register can be cleared by software from the RapidIO Interface once the Tsi620 completes initialization.

19.4.2 Initializing the Tsi620 from PCI

The PCI Interface can initialize the Tsi620. All the Tsi620 registers can be memory-mapped through the PCI Interface. For information on how to memory map registers on the Tsi620 PCI Interface, see “Overview of Device Register Map”.

Once this is completed, all PCI, SREP, and Tsi620 Switch registers can be accessed directly from the PCI bus.

Switch registers cannot be accessed using Maintenance transactions generated from the SREP.

19.4.3 Initializing the SREP

The initialization of the SREP depends on the direction the block is being initialized from — register bus or RapidIO. This chapter assumes initialization is performed from the register bus. Steps that can be skipped when initialization occurs from RapidIO are identified.

19.4.3.1 Establishing Link Sanity

If the system has a RapidIO host, it will be necessary to initialize the Tsi620 Switch before accessing the SREP’s registers.

19.4.3.2 Saving and Clearing Error Status Information

Once the port is powered up, error information from before the reset can be captured, if necessary, to determine the state of the system before the reset. The following registers have bits that assist in hardware debug:

- "SREP Logical and Transport Layer Error Detect CSR"
- "SREP Logical and Transport Layer Address Capture CSR"
- "SREP Logical and Transport Layer Device ID Capture CSR"
- "SREP Logical and Transport Layer Control Capture CSR"
- "SREP R2I Event Status Register"
- "SREP R2I Error ISF Command Attributes Capture CSR"
- "SREP R2I Error ISF Logical Error Decomposition Attributes Capture CSR"
- "SREP R2I BAR and LUT Parity Error Status Register"
- "SREP ISF Logical Error Detect CSR"
- "SREP ISF Logical Error Upper Attributes Capture CSR"
- "SREP ISF Logical Error Middle Attributes Capture CSR"
- "SREP ISF Logical Error Lower Attributes Capture CSR"
- "SREP ISF Logical Error RapidIO Routing Attributes Capture CSR"
- "SREP ISF Logical Error RapidIO Physical Attributes Capture CSR"
- "SREP ISF Logical Error RapidIO Lower Address Capture CSR"
- "SREP I2R BAR and LUT Parity Error Status Register"
- "SREP ISF ECC Error Status Register"
- "SREP Interrupt Status Register"
- "SREP Port-Write Status Register"

All errors should be cleared by accesses to the following registers:

- "SREP Logical and Transport Layer Error Detect CSR"
- "SREP R2I Event Status Register"
- "SREP ISF Logical Error Detect CSR"
- "SREP Interrupt Status Register"

19.4.3.3 Configuring Transaction Bridging

If the SREP is being configured from RapidIO, the first step should be to memory map and protect the registers by accessing the following registers:

- "SREP Local Configuration Space Base Address CSR"
- "SREP R2I LUT and Parity Control Register"
- "SREP Register Access Source ID Checking Control Register"

- "SREP Register Access Small Source ID Checking Control Register"
- "SREP Large Register Access Source ID Checking Register"

In addition, the destination ID of the SREP should be configured through the following registers:

- "SREP Base Device ID CSR"
- "SREP Destination ID Checking Control Register"
- "SREP Large Secondary Destination ID Checking Control Register"

Transaction bridging must be configured separately for two directions: RapidIO-to-Bridge ISF, and Bridge ISF-to-RapidIO. To configure transaction bridging in the RapidIO-to-Bridge ISF direction, the following registers must be accessed:

- "SREP Processing Element Logical Layer Control CSR"
- "SREP R2I Base Address Register x LUT Control CSR"
- "SREP R2I Base Address Register x Lower"
- "SREP R2I LUT and Parity Control Register"
- "SREP R2I Upper LUT Entry Translation Address Register"
- "SREP R2I Lower LUT Entry Translation Address Register"
- "SREP R2I ISF Request Priority Control Register"
- "SREP R2I ISF Response Priority Control Register"
- "SREP R2I RapidIO Miscellaneous Control CSR"

To control RapidIO-to-Bridge ISF performance, the following registers must be accessed:

- "SREP R2I ISF Watermarks Control Register"
- "SREP R2I ISF Buffer Release Control Register"
- "SREP R2I Watermarks Register"
- "SREP R2I Transaction Time-To-Live Register"
- "SREP ISF Response Timeout Register"

To configure transactions in the Bridge ISF-to-RapidIO direction, the following registers must be accessed:

- "SREP General Control CSR"
- "SREP Response Timeout Control CSR"
- "SREP I2R Base Address Register x LUT Entry CSR"
- "SREP I2R Base Address Register x Upper"
- "SREP I2R Base Address Register x Lower"
- "SREP I2R Doorbell BAR Upper"
- "SREP I2R Doorbell BAR Lower"
- "SREP I2R LUT and BAR Parity Control Register"

- "SREP I2R Upper LUT Entry Translation Register"
- "SREP I2R Lower LUT Entry Translation Address Register"
- "SREP I2R LUT Translation Parameters Register"
- "SREP R2I RapidIO Miscellaneous Control CSR"

To control Bridge ISF-to-RapidIO performance, the following registers must be accessed:

- "SREP I2R Transaction Time-To-Live Register"
- "SREP I2R Buffer Release Control Register"
- "SREP R2R Queue Watermarks Control Register"
- "SREP R2R Queue Buffer Release Control Register"
- "SREP Response Timeout Control CSR"

19.4.3.4 Initializing Error Management

Error management initialization is performed through accessing the registers associated with each error, as defined in "Event Notification and Register Hierarchy". These registers control the checking of events, as well as the notification strategy for the events (either interrupts or port-writes).

To configure port-write transmission, the following registers must be initialized:

- "SREP Port-Write Target Device ID CSR"
- "SREP Port-Write Parameters Register"
- "SREP Component Tag CSR"

To enable performance monitoring, the following registers must be accessed:

- "SREP ISF Performance Statistics Counter 0 and 1 Control Register"
- "SREP ISF Performance Statistics Counter 2 and 3 Control Register"
- "SREP ISF Performance Statistics Counter 4 and 5 Control Register"

20. Electrical Characteristics

Topics discussed include the following:

- "Absolute Maximum Ratings"
- "Recommended Operating Conditions"
- "Power Characteristics"
- "AC/DC Signal Characteristics"

20.1 Absolute Maximum Ratings

Table 115 lists the absolute maximum ratings.

Operating the Tsi620 beyond the operating conditions is not recommended. Stressing the device beyond the absolute maximum ratings can cause permanent damage.

Table 115: Absolute Maximum Ratings

Symbol	Parameter	Minimum	Maximum	Unit
$T_{STORAGE}$	Storage temperature	-40	125	°C
T_C	Case temperature under bias	-40	120	°C
Voltage with respect with ground				
PCI PLL_AVDD	PCI PLL analog power	-0.5	4.1	V
CLKGEN PLL_AVDD	SSPLL analog power	-0.5	2.0	V
VDD_PCI(3.3V)	PCI I/O power	-0.5	4.1	V
VDD_HSTL(1.5V)	FPGA Interface I/O power	-0.5	1.9	V
SP6_VREF	Reference power supply for differential stage of input receiver (VRF 1.5v option)	0	1.125	V
SP_AVDD	3.3 V analog supply voltage	-0.5	4.6	V
V_{DD}, SP_VDD, REF_AVDD	1.2 V DC supply voltage	-0.3	1.7	V
$V_{I\_SP\{n\}-R\{A-D\}\_\{p,n\}}$	SerDes port receiver input voltage	-0.3	3	V
$V_{O\_SP\{n\}-T\{A-D\}\_\{p,n\}}$	SerDes port VM Transmitter output voltage	-0.3	3	V
V_{I_LVTTL}	LVTTL input voltage	-0.5	VDD_PCI +0.5	V

Table 115: Absolute Maximum Ratings (Continued)

Symbol	Parameter	Minimum	Maximum	Unit
V_{O_LVTTL}	LVTTL output or I/O voltage	-0.5	VDD_PCI +0.5	V
V_{ESD_HBM}	Maximum ESD voltage discharge tolerance for Human Body Model (HBM). Test conditions per JEDEC standard - JESD22-A114-B.	-	2000	V
V_{ESD_CDM}	Maximum ESD voltage discharge tolerance for Charged Device Model (CDM). Test Conditions per JEDEC standard - JESD22-C101-A.	-	500	V

20.2 Recommended Operating Conditions

Table 116 lists the recommended operating conditions.

Continued exposure of the Tsi620 to the maximum limits of the specified junction temperature can affect the device's reliability. Subjecting the device to temperatures beyond the maximum/minimum limits can result in a permanent failure of the device.

Table 116: Recommended Operating Conditions

Symbol	Parameter	Minimum	Maximum	Unit
T_j	Junction temperature	-40	120	°C
T_A	Ambient temperature – Commercial	0	70	°C
	Ambient temperature – Industrial	-40	85	°C
PCI PLL_AVDD	PCI PLL analog power	3.0	3.6	V
CLKGEN PLL_AVDD	SSPLL analog power	1.14	1.26	V
V_{DD_PCI}(3.3V)	PCI I/O power	3.0	3.6	V
V_{DD_HSTL}(1.5V)	FPGA Interface I/O power	1.4	1.6	V
SP6_VREF	Reference power supply for differential stage of input receiver (VRF 1.5v option). Note: Must be half of VDD_HSTL.	0.67	0.82	V
SP_AVDD	3.3 V analog supply voltage	2.97	3.63	V
V_{DD}, SP_VDD	1.2 V DC supply voltage	1.14	1.26	V
I_{SP_VDD}	SerDes digital supply current	-	351	mA

Table 116: Recommended Operating Conditions (Continued)

Symbol	Parameter	Minimum	Maximum	Unit
I_{SP_AVDD}	3.3 V SerDes supply current	-	TBD	mA
I_{VDD}	1.2 V Core supply current	-	TBD	mA
I_{VDD_PCI}	3.3V PCI supply current	-	TBD	mA
$V_{ripple1}$	Power supply ripple for voltage supplies: SP_VDD and VDD	-	100	mV_{pp}
$V_{ripple2}$	Power supply ripple for voltage supplies: SP{n}_AVDD, REF_AVDD	-	50	mV_{pp}
I_{REXT}	External reference resistor current	-	10	uA

20.3 Power Characteristics

The following sections describe the Tsi620's power dissipation and power sequencing.

20.3.1 Power Consumption

The power consumption values provided are dependent on device configuration. The line rate, port configuration, and traffic, all affect the Tsi620's power consumption.

Table 117: Power Consumption — Three 4x Ports, 4x XGMII, PCI 66 MHz

Line Rate[a]	1.25 GBaud	2.5 GBaud	3.125 GBaud
SPn_VDD	0.313	0.333	0.404
SPn_AVDD	0.696	0.757	0.865
VDD_CORE	1.099	1.421	1.564
HSTL	0.250	0.285	0.285
VREF	0.023	0.023	0.023
VDD_IO	0.818	0.818	0.817
Total Power Consumption (W)	3.199	3.636	3.958

a. Line rate applies to RapidIO links and XGMII interface. RapidIO port traffic is at 100 percent.

Table 118: Power Consumption — Six 1x Ports, 1x XGMII, PCI 66 MHz

Line Rate[a]	1.25 GBaud	2.5 GBaud	3.125 GBaud
SPn_VDD	0.281	0.290	0.352
SPn_AVDD	0.629	0.655	0.746
VDD_CORE	1.160	1.513	1.695
HSTL	0.094	0.099	0.096
VREF	0.023	0.023	0.023
VDD_IO	0.817	0.823	0.817
Total Power Consumption (W)	3.005	3.402	3.729

a. Line rate applies to RapidIO links and XGMII interface. RapidIO port traffic is at 100 percent.

20.3.2 Power Sequencing

The Tsi620 must have the supplies powered up in the following order:

1. VDD (1.2 V) must be powered up first.
2. SP_VDD (1.2 V) and REF_AVDD (1.2 V) should power up at approximately the same time as VDD.
3. Delays between the powering up of VDD, SP_VDD, and REF_AVDD are acceptable.
4. No more than 50 ms after VDD is at a valid level, VDD_PCI (3.3 V) should be powered up to a valid level.
5. VDD_PCI (3.3V) must not power up before VDD (1.2 V).
6. SP_AVDD (3.3V) should power up at approximately the same time as VDD_PCI.
7. Delays between powering up VDD_PCI and SP_AVDD are acceptable.
8. SP_AVDD must not power up before SP_VDD.

It is recommended that there not be more than 50 ms between ramping of the 1.2 V and 3.3 V supplies. The power supply ramp rates must be kept between 10 V/s and 1x10E6 V/s to minimize power current spikes during power up.

If it is necessary to sequence the power supplies in a different order than the one recommended, the following precautions must be taken:

- Any power-up option pins must be current limited with 10 K ohms to VDD_PCI or VSS_IO as required to set the desired logic level.
- Power-up option pins that are controlled by a logic device must not be driven until all power supply rails to the Tsi620 are stable.

20.3.2.1 Power down

Power down is the reverse sequence of power up:

1. VDD_PCI (3.3V) and SP{n}_AVDD.
2. VDD (1.2V), SP_VDD, and REF_AVDD power-down at the same time, or all rails falling simultaneously.

20.4 AC/DC Signal Characteristics

This section describes the AC and DC signal characteristics for the Tsi620.

20.4.1 SerDes Receiver (SP{n}_RD_p/n)

Table 119 lists the electrical characteristics for the SerDes Receiver.

Table 119: SerDes Receiver Electrical Characteristics

Symbol	Parameter	Min.	Typ.	Max.	Unit	Notes
Z_{DI}	RX Differential Input impedance	90	100	110	Ohm	-
V_{DIFFI}	RX Differential Input Voltage	170	-	1600	mV	-
L_{CR}	RX Common Mode Return Loss	-	-	6	dB	Over a range 100MHz to 0.8* Baud Frequency
L_{DR}	RX Differential Return Loss	-	-	10	dB	Over a range 100MHz to 0.8* Baud Frequency
V_{LOS}	RX Loss of Input Differential Level	55	-	-	mV	Port Receiver Input level below which Low Signal input is detected
$T_{RX\_ch\_skew}$	RX Channel to Channel Skew Tolerance	-	-	24	ns	Between channels in a given x4 port @ 1.25/2.5Gbps
		-	-	22	ns	Between channels in a given x4 port @ 3.125Gbps
$R_{TR,}R_{TF}$	RX Input Rise/Fall times	-	-	160	ps	Between 20% and 80% levels

20.4.2 SerDes Transmitter (SP{n}_TD_p/n)

Table 120 lists the electrical characteristics for the SerDes Transmitter.

Table 120: SerDes Transmitter Electrical Characteristics

Symbol	Parameter	Min.	Typ.	Max.	Unit	Notes
Z_{SEO}	TX Single-Ended Output impedance	45	50	55	Ohm	-
Z_{DO}	TX Differential Output Impedance	90	100	110	Ohm	-
V_{SW}	TX Output Voltage Swing (Single-ended)	425		600	mVp-p	V_{SW} (in mV) = Z_{SEO}/2 x Inom x RIdr/Inom, where Ridr/Inom is the Idr to Inom ratio.
V_{DIFFO}	TX Differential Output Voltage Amplitude	-	$2*V_{SW}$		mVp-p	-
V_{OL}	TX Output Low-level Voltage	-	1.2 - V_{SW}		V	-
V_{OH}	TX Output High-level Voltage	-	1.2		V	-
V_{TCM}	TX common-mode Voltage	-	1.2 - V_{SW}/2		V	-
L_{DR1}	TX Differential Return Loss	-	-	10	dB	For (Baud Frequency)/10<Freq(f)< 625MHz and
L_{DR2}	TX Differential Return Loss	-	-	10 + |10log(f/625MHz)|	dB	For 625MHz<=Freq(f)<= Baud Frequency
T_{TX_skew}	TX Differential signal skew	-	-	15	ps	Skew between _p and _n signals on a give Serial channel
T_{TR},T_{TF}	TX Output Rise/Fall times	80	-	110	ps	Between 20% and 80% levels

20.4.3 Reference Clock (S_CLK_p/n)

Table 121 lists the electrical characteristics for the differential SerDes Reference clock input (S_CLK_p/n) in the Tsi620.

Table 121: Reference Clock (S_CLK_p/n) Electrical Characteristics

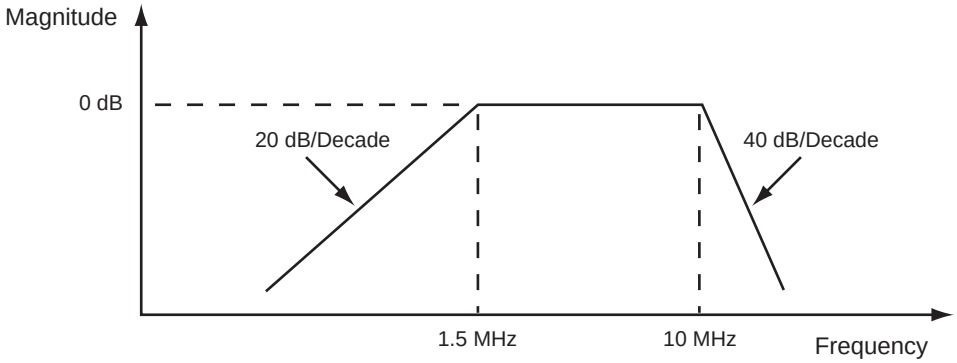
Symbol	Parameter	Min.	Typ.	Max.	Unit	Notes
V_{SW}	Input voltage swing	0.1	0.5	1	V	-
V_{DIFF}	Differential input voltage swing	$V_{DIFF} = V_{SW} * 2$			V	-

Table 121: Reference Clock (S_CLK_p/n) Electrical Characteristics (Continued)

Symbol	Parameter	Min.	Typ.	Max.	Unit	Notes
V_{CM}	Differential Input Common Mode Range ((S_CLK_p + S_CLK_n)/2)	175	-	2000	mV	The S_CLK_p/n must be AC coupled.
Fin	Input Clock Frequency	156.25	-	156.25	MHz	-
$F_{S\_CLK\_P/N}$	Reference Clock Frequency Stability	-100	-	+100	ppm	PPM with respect to 156.25 MHz.
Fin_DC	Reference Clock Duty Cycle	40	50	60	%	-
T_{skew}	Reference Clock Skew	-	-	0.32	ns	Between _p and _n inputs.
T_{R_SCLK}, T_{F_SCLK}	S_CLK_p/n Input Rise/Fall Time	-	-	1	ns	-
$J_{CLK-REF}$	Total Phase Jitter, rms	-	-	3	ps_{rms}	[a]
Zin	Input Impedance	80	100	114	ohms	-

a. Total Permissible Phase Jitter on the Reference Clock is 3 ps rms. This value is specified with the assumption that the measurement is completed with a 20 GSamples/s scope with more than 1 million samples taken. The zero-crossing times of each rising edges are recorded and an average Reference Clock is calculated. This average period may be subtracted from each sequential, instantaneous period to find the difference between each reference clock rising edge and the ideal placement to produce the Phase Jitter Sequence. The PSD of the phase jitter is calculated and integrated after being weighted with the transfer function shown in Figure 51. The square root of the resulting integral is the rms Total Phase Jitter.

Figure 51: Weighing Function for RMS Phase Jitter Calculation



20.4.4 PCI Interface Signals

Table 122 contains AC and DC operating characteristics for the PCI Interface.

Table 122: PCI Interface DC Operating Characteristics

Symbol	Parameter	Condition	Minimum	Maximum	Units	Notes
V_{IL_PCI}	PCI Input Low Voltage	-	-0.5	$0.35V_{DD\_PC}$	V	-
V_{IH_PCI}	PCI Input High Voltage	-	$0.5V_{DD\_PC}$	V_{DD_PCI} + 0.5	V	-
C_{IN_PCI}	Input Pin Capacitance	-	-	8.8	pF	-
C_{CLK_PCI}	Clock Pin Capacitance PCI_CLK	-	-	7.5	pF	-
L_{IN_PCI}	Input Pin Inductance	-	-	8.3	nH	-
L_{CLK_PCI}	Clock Pin Inductance PCI_CLK	-	-	4.9	nH	-

Table 123: PCI Clock (PCI_CLK) Specification

Symbol	Parameter	PCI		Units	Notes
		Min.	Max.		
T_{F_PCI}	PCI Clock Frequency	25	66	MHz	a
T_{C_PCI}	PCI Clock Cycle Time	15	40	ns	a, b
T_{CH_PCI}	PCI Clock High Time	6	-	ns	-
T_{CL_PCI}	PCI Clock Low Time	6	-	ns	-
T_{SR_PCI}	PCI Clock Slew Rate	1	6	V/ns	c

a. The clock frequency may not change beyond the spread-spectrum limits except while chip reset is asserted.

b. The minimum clock period must not be violated for any single clock cycle.

c. This slew rate must be met across the minimum peak-to-peak portion of the clock waveform.

Table 124: AC Specifications for PCI Interface

Symbol	Parameter	PCI 66		PCI 33		Units	Notes
		Min.	Max.	Min.	Max.		
T_{OV1}	Clock to Output Valid Delay for bused signals	1	6	2	11	ns	a, b, c
T_{OV2}	Clock to Output Valid Delay for point to point signals	2	6	2	12	ns	a, b, c
T_{OF}	Clock to Output Float Delay	-	14	-	28	ns	a, d
T_{IS1}	Input Setup to clock for bused signals	3	-	7	-	ns	c, e, f
T_{IS2}	Input Setup to clock for point to point signals	5	-	10,12	-	ns	c, d
T_{IH1}	Input Hold time from clock	0	-	0	-	ns	d
T_{RST}	Reset Active Time	1	-	1	-	ms	
T_{RF}	Reset Active to output float delay	-	40	-	40	ns	-
T_{IS3}	P[x]_REQ64_B to Reset setup time	10	-	10	-	clocks	-
T_{IH2}	Reset to P[x]_REQ64_B hold time	0	50	0	50	ns	-

a. See the timing measurement conditions in Figure 53.

b. See Figure 54, Figure 55, and Figure 56.

c. Setup time for point-to-point signals applies to P[x]_REQ_B and P[x]_GNT_B only. All other signals are bused.

d. For purposes of Active/Float timing measurements, the HI-Z or “Off” state is defined to be when the total current delivered through the component pin is less than or equal to the leakage current specification.

e. See the timing measurement conditions in Figure 52.

f. Setup time applies only when the device is not driving the pin. Devices cannot drive and receive signals at the same time.

Table 125: PCI Output Clock Specifications

Symbol	Parameter	Min.	Nom.	Max.	Units	Notes
T_{SKEW_CGPCI}	Skew between any two CG_PCI_CLKO outputs	-	-	58	ps	-
T_{JPER_CGPCI}	CG_PCI_CLKO Clock period jitter	-225	-	225	ps	e, d

20.4.4.1 PCI Interface Signal Timing Diagrams

Figure 52: Input Timing Measurement Waveforms

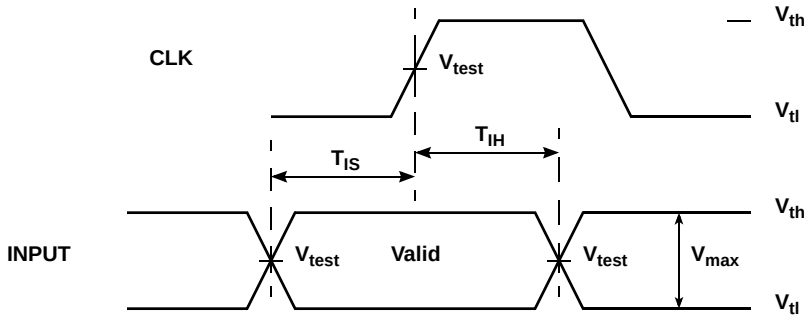


Figure 53: Output Timing Measurement Waveforms

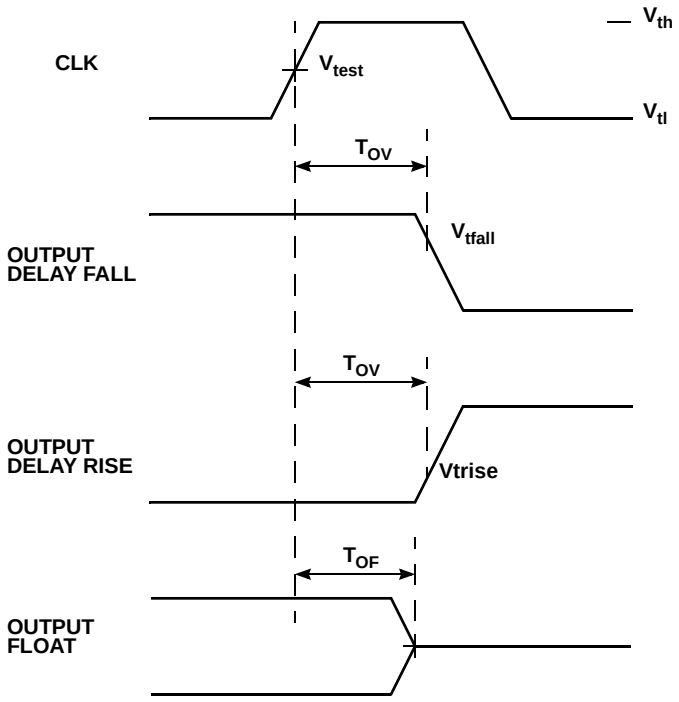


Figure 54: PCI $T_{OV\ (max)}$ Rising Edge AC Test Load

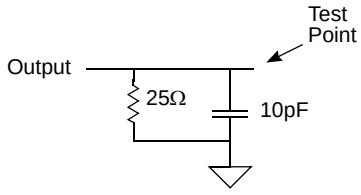


Figure 55: PCI $T_{OV\ (max)}$ Falling Edge AC Test Load

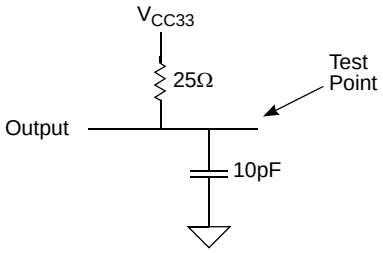
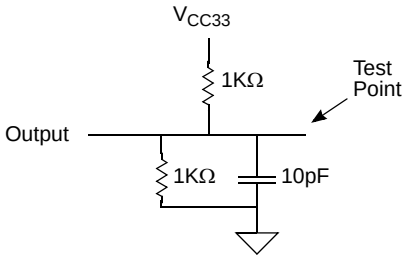


Figure 56: PCI $T_{OV\ (min)}$ AC Test Load



20.4.5 Interrupt Controller Signal Timing

The following table lists the AC specifications for the Interrupt Controller.

Table 126: AC Specifications for Interrupt Controller

Symbol	Parameter	Min.	Max.	Units	Notes
T_{EDGE}	INT rise/fall time	-	0.5	Switch Fabric clock period	a

a. INT and GPIO inputs provide only 5mV hysteresis and a single ISF (Bridge and Switch) clock period of de-bounce.

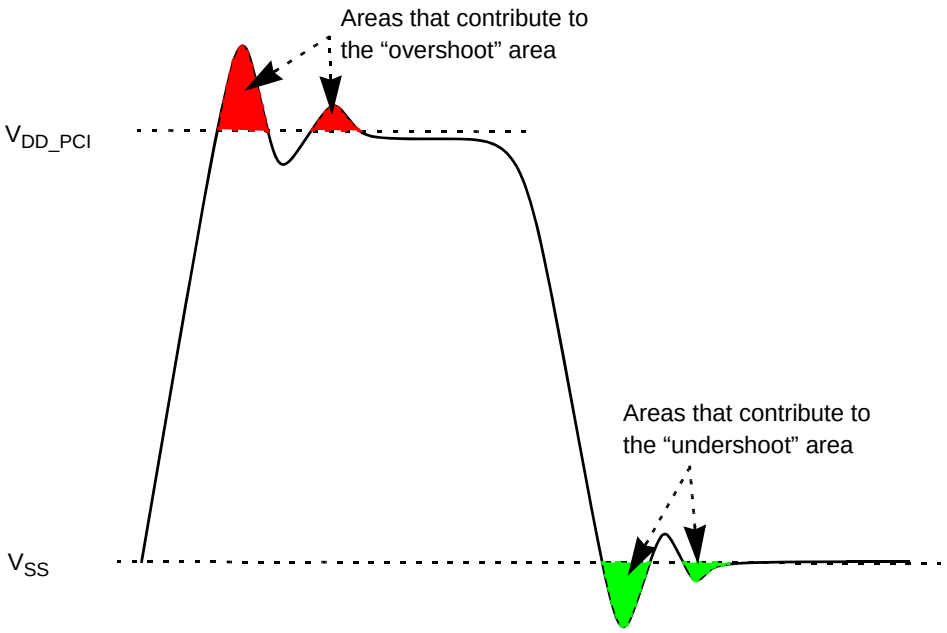
Table 127: AC Overshoot/Undershoot Specification for PCI, Interrupt, and GPIO Pins

Symbol	Parameter	Max.	Units	Notes
-	Maximum allowable peak overshoot above V_{DD_PCI}	+0.5	V	-
-	Maximum allowable peak undershoot below ground	-0.5	V	-

Table 127: AC Overshoot/Undershoot Specification for PCI, Interrupt, and GPIO Pins (Continued)

Symbol	Parameter	Max.	Units	Notes
-	Maximum area under the overshoot signal above V_{DD_PCI}	0.75	V-ns	-
-	Maximum area above the undershoot signal below V_{SS}	0.75	V-ns	-

Figure 57: Interrupt Controller Overshoot/Undershoot



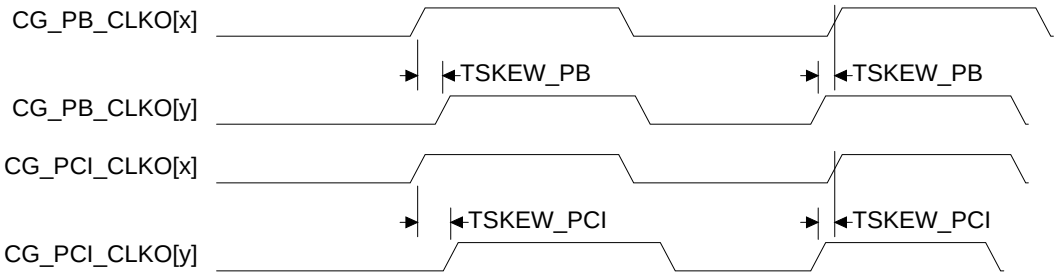
20.4.6 Clock Generator AC Signal Timing

Table 128: Clock Generator Clock Specifications

Symbol	Parameter	Min.	Nom.	Max.	Units	Notes
T_{F_CG}	Reference Clock Frequency (PLL enabled - 25MHz mode)	-	25	-	MHz	a
T_{F_CG}	Reference Clock Frequency (PLL enabled - 33MHz mode)	-	33.33	-	MHz	
T_{FA_CG}	Reference clock Frequency accuracy	-100	-	+100	ppm	b
T_{C_CG}	Reference Clock Cycle Period (PLL enabled)	30	-	-	ns	-
T_{CH_CG}	Reference Clock High Time	10	-	-	ns	-
T_{CL_CG}	Reference Clock Low Time	10	-	-	ns	-
T_{SR_CG}	Reference Clock Slew Rate	1	-	6	V/ns	c
T_{JPER_CG}	Reference Clock period jitter	-	-	1.75	%	d
T_{F_NOPLL}	Reference Clock Frequency (PLL disabled)	-	-	133.333	MHz	-
T_{C_NOPLL}	Reference Clock Cycle Period (PLL disabled)	7.5	-	-	ns	-
T_{CH_NOPLL}	Reference Clock High Time (PLL disabled)	3	-	-	ns	-
T_{CL_NOPLL}	Reference Clock Low Time (PLL disabled)	3	-	-	ns	-
T_{SR_NOPLL}	Reference Clock Slew Rate (PLL disabled)	1	-	6	V/ns	-
T_{JPER_NOPLL}	Reference Clock period jitter (PLL disabled)	-	-	2	%	-
T_{DUTY_CGPB}	Clock Generator output duty cycle	45	-	55	%	-
T_{SKEW_CGPB}	Skew between any two CG_PB_CLKO outputs	-	-	27	ps	-
T_{JPER_CGPB}	CG_PB_CLKO Clock period jitter	-100	-	100	ps	e, d
T_{DUTY_CGPCI}	Clock Generator output duty cycle	45	-	55	%	-
T_{SKEW_CGPCI}	Skew between any two CG_PCI_CLKO outputs	-	-	58	ps	-
T_{JPER_CGPCI}	CG_PCI_CLKO Clock period jitter	-225	-	225	ps	e, d

a. Reference clock frequency must not exceed the limits set by the spread spectrum requirements except while reset is active.
b. A 33.33MHz or 25MHz oscillator with +/- 100ppm accuracy meets the requirements for T_{F_CG} and T_{FA_CG}.
c. Measured from between V_{IL} and V_{IH}.
d. The time difference between a measured cycle period and the ideal cycle period.
e. CG_REF must meet all specified clock input requirements.

Figure 58: Clock Generator Interface Timing



20.4.7 HSTL (FPGA Interface) Signals

Table 131 lists the electrical characteristics for the 1.5 V digital HSTL Interface pins on the Tsi620.

Table 129: HSTL Electrical Characteristics

Symbol	Parameter	Min.	Typ.	Max.	Unit	Notes
V_{IHT_HSTL}	HSTL Input Logic Threshold High	V_{REF_HSTL}+0.1	-	-	V	All inputs for FPGA Interface, DC
V_{ILT_HSTL}	HSTL Input Logic Threshold Low	-	-	V_{REF_HSTL}-0.1	V	All inputs for FPGA Interface, DC
V_{IHAC_HSTL}	HSTL Input Logic Threshold High, AC	V_{REF_HSTL} +0.20	-	-	V	All inputs for FPGA Interface, AC From JEDEC specification for HSTL I/O Standard
V_{ILAC_HSTL}	HSTL Input Logic Threshold Low, AC	-	-	V_{REF_HSTL} -0.20	V	All inputs for FPGA Interface, AC From JEDEC specification for HSTL I/O Standard
I_{IH_HSTL}	HSTL Input High Current	-	-	-6	uA	All inputs for FPGA Interface
$I_{HSTL\_OZL\_PU}$, $I_{HSTL\_IL\_PU}$	HSTL Input Low/ Output Tristate Current	-	-	25	uA	-
$I_{HSTL\_ZH\_PD}$, $I_{HSTL\_IH\_PD}$	HSTL Input High/ Output Tristate Current	-	-	25	uA	-

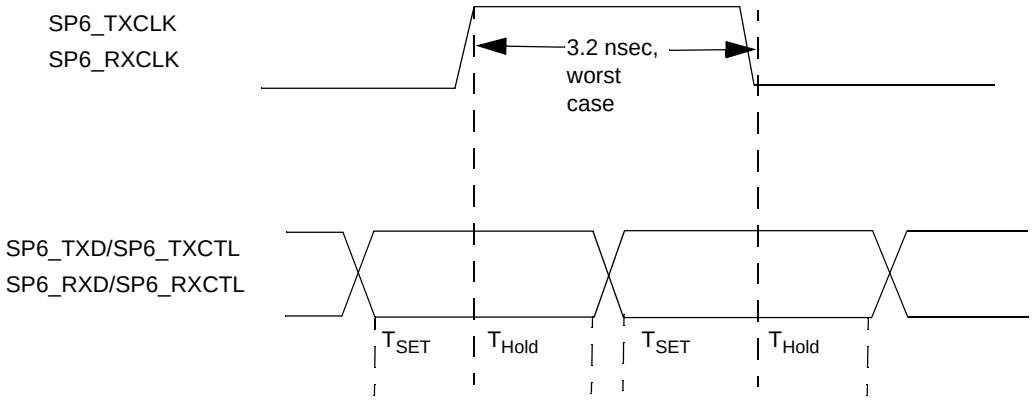
Table 129: HSTL Electrical Characteristics (Continued)

Symbol	Parameter	Min.	Typ.	Max.	Unit	Notes
F_{MAX}	Maximum operating frequency	-		156.25	MHz	Maximum verified operating frequency, lower frequencies are possible.
V_{OH_HSTL}	Output Logic High	VDD_HSTL - 0.4			V	Value for unterminated drivers.
V_{OL_HSTL}	Output Logic Low			0.4	V	Value for unterminated drivers.

Table 130: Transmit Clock and Receive Clock Timing Parameters — FPGA Interface

Symbol	Transmitter	Receiver	Units
T_{SET}	960	480	Picoseconds
T_{Hold}	960	480	Picoseconds

Figure 59: Setup and Hold Timing — FPGA Interface



20.4.8 LVTTL I/O and Open Drain Signals

Table 131 lists the electrical characteristics for the 3.3 V digital LVTTL Interface pins on the Tsi620.

Table 131: LVTTL I/O and Open Drain Electrical Characteristics

Symbol	Parameter	Min.	Typ.	Max.	Unit	Notes
V_{IL}	LVTTL Input Low Voltage	-	-	0.8	V	All inputs and I/Os of LVTTL type
V_{IH}	LVTTL Input High Voltage	2.0	-	-	V	All inputs and I/Os of LVTTL type
I_{IL}	LVTTL Input Low Current	-0.3	-	VDD_PCI+0.5	uA	All non-PU inputs and I/Os of LVTTL type
I_{IH}	LVTTL Input High Current	-	-	-10	uA	All non-PD inputs and I/Os of LVTTL type
I_{OZL_PU}, I_{IL_PU}	LVTTL Input Low/ Output Tristate Current	5	-	100	uA	All PU inputs and I/Os of LVTTL type for voltages from 0 to VDD_PCI on the pin.
I_{OZH_PD}, I_{IH_PD}	LVTTL Input High/ Output Tristate Current	-5	-	-100	uA	All PD inputs and I/Os of LVTTL type for voltages from 0 to VDD_PCI on the pin.
V_{OL}	LVTTL Output Low Voltage	-	-	0.4	V	I_{OL}=2mA for INT_b, CHIP_RST_b, BLK_RST_b, RST_IRQ_b and TDO pins I_{OL}=8mA for I2C_SCLK and I2C_SD pins
V_{OH}	LVTTL Output Low Voltage	VDD_PCI -0.5	-	-	V	I_{OH}=2mA for INT_b, CHIP_RST_b, BLK_RST_b, and TDO pins
$V_{OVERSHOOT}$	Dynamic Overshoot	-	-	0.9	V	0.9V Max with a maximum energy of 0.75 V-ns
V_{Hyst}	LVTTL Input Hysteresis Voltage	-	200	-	mV	All Hysteresis inputs and I/Os of LVTTL type
C_{Pad}	LVTTL Pad Capacitance	-	-	10	pF	All pads of LVTTL type
T_{cfgpS}	Configuration Pin Setup Time	100	-	-	ns	For all Configuration pins (except SP{n}_MODE_SEL with respect to CHIP_RST_b/BLK_RST_b rising edge
T_{cfgpH}	Configuration Pin Hold Time	100	-	-	ns	For all Configuration pins (except SP{n}_MODE_SEL) with respect to CHIP_RST_b/BLK_RST_b rising edge
$T_{sp\_modeselS}$	SP{n}_MODE_SEL Setup Time	5	-	-	ns	With respect to rising edge of P_CLK. SP{n}_MODE_SEL pins are sampled on every other rising edge of S_CLK.

Table 131: LVTTL I/O and Open Drain Electrical Characteristics (Continued)

Symbol	Parameter	Min.	Typ.	Max.	Unit	Notes
$T_{sp\_modeseH}$	SP{n}_MODE_SEL Hold Time	5	-	-	ns	With respect to rising edge of S_CLK. SP{n}_MODE_SEL pins are sampled on every other rising edge of S_CLK.
T_{ISOV1}	INT_b and CHIP_RST_b/BLK_RST_b Output Valid Delay from rising edge of S_CLK	-	-	15	ns	Measured between 50% points on both signals. Output Valid delay is guaranteed by design.
T_{ISOF1}	NT_b and CHIP_RST_b/BLK_RST_b Output Float Delay from rising edge of S_CLK	-	-	15	ns	A float condition occurs when the output current becomes less than I_{LO}, where I_{LO} is 2 x I_{OZ}. Float delay guaranteed by design.
$F_{in\_S\_CLK}$	Input Clock Frequency	100	-	100	MHz	-
F_{in_STAB}	P_CLK Input Clock Frequency Stability	-100	-	+100	ppm	-
$F_{in\_PCLK\_DC}$	P_CLK Input Clock Duty Cycle	40	50	60	%	-
J_{PCLK}	P_CLK Input Jitter	-	-	300	ps_{pp}	-
T_{R_PCLK}, T_{F_PCLK}	P_CLK Input Rise/Fall Time	-	-	2.5	ns	-
f_{MCES}	MCES pin frequency	-	-	1	MHz	Both as input and output
R pull-up	Resistor pull-up	82K	-	260K	ohms	@Vil=0.8V
R pull-down	Resistor pull-down	28K	-	54K	ohms	@Vih=2.0V

20.4.9 I²C Interface

Table 132 lists the AC specifications for the I²C Interface. This interface includes the following signals: I2C_SCLK, I2C_SD, I2C_DISABLE, I2C_MA, I2C_SEL, I2C_SA[6:0], and I2C_SEL.

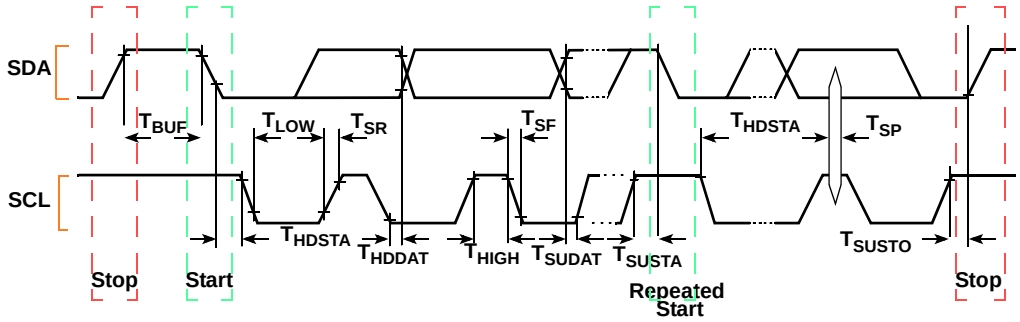
Table 132: AC Specifications for I²C Interface

Symbol	Parameter	Min.	Max.	Units	Notes
F_{SCL}	I2C_SD/I2C_SCLK Clock Frequency	0	100	kHz	-
T_{BUF}	Bus Free Time Between STOP and START Condition	4.7	-	µs	a
T_{LOW}	I2C_SD/I2C_SCLK Clock Low Time	4.7	-	µs	a
T_{HIGH}	I2C_SD/I2C_SCLK Clock High Time	4	-	µs	a
T_{HDSTA}	Hold Time (repeated) START condition	4	-	µs	a, b
T_{SUSTA}	Setup Time for a Repeated START condition	4.7	-	µs	a
T_{HDDAT}	Data Hold Time	0	3.45	µs	a
T_{SUDAT}	Data Setup Time	250	-	ns	a
T_{SR}	Rise Time for I2C_xxx (all I2C signals)	-	1000	ns	a
T_{SF}	Fall Time for I2C_xxx (all I2C signals)	-	300	ns	a
T_{SUSTOP}	Setup Time for STOP Condition	4	-	µs	a

a. For more information on I²C Interface signal timing, see Figure 60.

b. After this period, the first clock pulse is generated.

Figure 60: I²C Interface Signal Timings



20.4.10 Boundary Scan Test Interface Timing

Table 133 lists the test signal timings for Tsi620.

Table 133: Boundary Scan Test Signal Timings

Symbol	Parameter	Min.	Max.	Units	Notes
T_{BSF}	TCK Frequency	0	25	MHz	-
T_{BSCH}	TCK High Time	50	-	ns	• Measured at 1.5V • Note test
T_{BSCL}	TCK Low Time	50	-	ns	• Measured at 1.5V • Note test
T_{BSCR}	TCK Rise Time	-	25	ns	• 0.8V to 2.0V • Note test
T_{BSCF}	TCK Fall Time	-	25	ns	• 2.0V to 0.8V • Note test
T_{BSIS1}	Input Setup to TCK	10	-	ns	-
T_{BSIH1}	Input Hold from TCK	10	-	ns	-
T_{BSOV1}	TDO Output Valid Delay from falling edge of TCK.[a]	-	15	ns	-
T_{OF1}	TDO Output Float Delay from falling edge of TCK	-	15	ns	-
$T_{BSTRST1}$	TRST_b release before CHIP_RST_b/BLK_RST_b release	-	10	ns	TRST_b must become asserted while CHIP_RST_b/BLK_RST_b is asserted during device power-up
$T_{BSTRST2}$	TRST_b release before TMS or TDI activity	1	-	ns	-

a. Outputs precharged to VDD.

21. Performance Information

This chapter discusses Tsi620 performance characteristics. The chapter is divided according to the two main device functions as follows:

- "Tsi620 Switch Performance"
- "Tsi620 Bridge Performance"

21.1 Tsi620 Switch Performance

This section discusses performance information for the switching segment of the Tsi620. Performance for packet switching is characterized by two measurements: throughput and latency.

Performance is specified for error-free transmission and reception of packets for pin-to-pin transfers through the Tsi620 Switch. No performance specifications are made for the different stages of transfers through the Tsi620 Switch. Performance is specified for a single switch; however, performance for larger systems can be computed from this data.

21.1.1 Throughput

Throughput for a packet is a measurement of the amount of packet data that can be transferred in a given amount of time. It can be presented in different forms:

- Percentage of a link's bandwidth (for example, 56% of a 1x @ 3.125 Gbaud)
- Number of packets of a given size per unit time (for example, 3000 44-byte packets every second)
- Bit transfer rate (for example, 300 Mbps)

Throughput measurements include only successfully transferred packets. Measured throughput does not include control symbols, retried packets, or other non-packet data transmitted/received on a link (/K/ and /R/ characters).

21.1.2 Latency

Latency is the amount of time between packet reception and packet transmission. However, the specific time during which packet reception and transmission are considered to have started must still be defined. Throughout this section, latency is measured as the time interval between the first bit of the Start-of-Packet arriving at the Tsi620 ingress port and that same bit leaving the device.

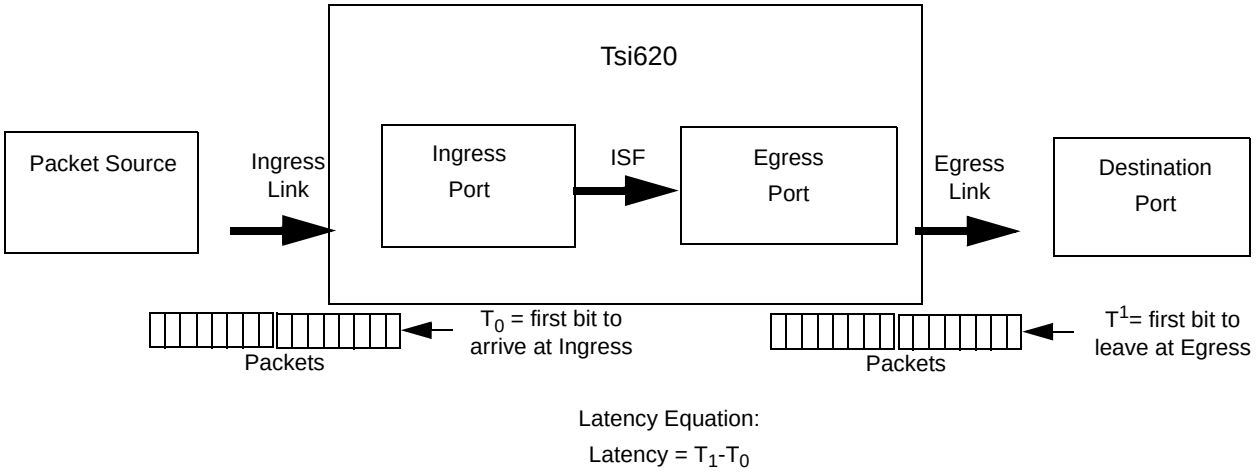
Figure 61 illustrates the path a packet follows through a Tsi620 Switch. For Tsi620 latency performance, packet reception time begins when the first bit of a packet is seen on the input pins. Packet transmission begins when the first bit of a packet is transmitted on the output pins.

As part of the resolution of resource contention, higher priority packets can be allowed to pass packets of lower priority. Latencies should therefore decrease as the priority of a packet increases.

A specific time for packet latency can only be specified when there are no conditions that create resource contention between packets. For example, if a single stream of packets passing from one ingress port to another egress port is the only traffic handled by the Tsi620 Switch, it is possible to specify the latency for the packets in this stream.

A complex traffic pattern is defined to be one that has resource contention. Complex traffic patterns make specifying the exact latency figure that each packet experiences difficult, because the amount of contention that a packet experiences can vary widely. As such, these scenarios are not covered in this document.

Figure 61: Tsi620 Switch Latency Example



In the Tsi620 Switch, packets experience packet latency variations caused by the asynchronous ability of the device. Packets can experience an extra one or two clock cycles of delay over the minimum latency when crossing from one clock domain to another clock domain. System designers should factor this into their timing budget.

21.1.3 Performance Monitoring

The main purpose of the performance monitoring functionality is to observe the data traffic on the RapidIO ports. The RapidIO traffic can come from different sources (for example, different processing endpoints) and can cause data congestion in one of the destination interfaces. This congestion can have a negative impact on overall system performance. Performance monitoring can identify and help prevent situations that negatively impact system performance.

Performance monitoring decisions can be made by system software in real-time. The system software can be programmed to routinely read the performance monitoring registers, analyze the traffic flow patterns, and re-route accordingly to avoid congestion.

Each RapidIO port in the Tsi620 Switch has a copy of the performance monitoring registers. Table 134 lists the statistic parameters that are available from the Outbound and Inbound registers, as part of each port's performance monitoring capabilities.

Table 134: Performance Monitoring Parameters

Parameters	Registers	Description
Number of 32-bit words	• "RapidIO Port x Performance Statistics Counter 0 and 1 Control Register" • "RapidIO Port x Performance Statistics Counter 0 Register"	Any of the performance statistics counter registers can be configured to count the number of 32-bit words sent or received by a RapidIO link.
Number of transactions	• "RapidIO Port x Performance Statistics Counter 0 and 1 Control Register" • "RapidIO Port x Performance Statistics Counter 0 Register"	Any of the performance statistics counter registers can be configured to count the number of packets sent or received by a RapidIO link.
Number of packets for each priority (0, 1, 2, and 3)	• "RapidIO Port x Performance Statistics Counter 0 and 1 Control Register" • "RapidIO Port x Performance Statistics Counter 0 Register"	Any of the performance statistics counter registers can be configured to count the number of packets sent or received by a RapidIO link with a specific priority.
Queue depth for inbound and outbound buffer	• "RapidIO Port x Transmitter Output Queue Depth Threshold Register" • "RapidIO Port x Transmitter Output Queue Congestion Status Register" • "RapidIO Port x Transmitter Output Queue Congestion Period Register" • "RapidIO Port x Receiver Input Queue Depth Threshold Register" • "RapidIO Port x Receiver Input Queue Congestion Status Register" • "RapidIO Port x Receiver Input Queue Congestion Period Register"	Performance statistics for monitoring the congestion situation in both Tx and Rx directions are supported.

The following sub-sections describe the use of the parameters for monitoring the performance of the Tsi620 RapidIO ports.

21.1.3.1 Traffic Efficiency

To characterize the efficiency of system traffic, the following parameters are used:

1. Packet rate (number of packets / time) – This is calculated using the number of packets computed from a counter register configured to count the number of packets.
2. Average packet size (number of 32-bit words / number of packets) – This is calculated using a counter configured to count the number of 32-bit words (call it COUNTER A), and a counter configured to count the number of packets (call it COUNTER B). The average packet size is COUNTER A divided by the value in COUNTER B.
3. Utilization ((packet rate * packet size) / maximum capacity) – This is calculated using parameter 1 and parameter 2, above.

These values are derived from the number of packets and the number of 32-bit words on each RapidIO port. The calculations of the packet rate, packet size, and utilization are completed externally.

21.1.3.2 Throughput Measurements

The count of packets per priority in each RapidIO port can be a very important parameter when debugging RapidIO systems. This information can also be valuable when used by system software to dynamically re-route traffic around congested ports. The following parameter is used to monitor the throughput on each RapidIO port:

- Number of packets for each priority level (0, 1, 2, and 3)
  - Each performance counter register, such as "RapidIO Port x Performance Statistics Counter 0 Register", can be configured to count the number of packets selected based on priority.

Retried packets are not counted.

21.1.3.3 Bottleneck Detection

Monitoring the queue depth of the ingress and egress ports can detect bottleneck traffic in the RapidIO ports. It can also determine the period of time that packets of a specific priority and below cannot be accepted. Both the inbound and outbound directions can program a queue depth watermark. The number of times that the queue depth watermark is exceeded is counted. As well, the amount of time that the queue depth watermark is exceeded is also counted to a programmable degree of accuracy. A port-write and/or an interrupt can be asserted if the queue depth watermark value exceeds a programmable number.

The registers in the outbound direction that contain the values and counters described above include:

- "RapidIO Port x Transmitter Output Queue Depth Threshold Register"
- "RapidIO Port x Transmitter Output Queue Congestion Status Register"
- "RapidIO Port x Transmitter Output Queue Congestion Period Register"

The registers in the inbound direction include:

- "RapidIO Port x Receiver Input Queue Depth Threshold Register"
- "RapidIO Port x Receiver Input Queue Congestion Status Register"
- "RapidIO Port x Receiver Input Queue Congestion Period Register"

21.1.3.4 Congestion Detection

A packet is reordered when it cannot make forward progress through the Switch ISF. Packet reordering can be a sign of congestion in a RapidIO port. A count of the number of times packets are reordered in each port is stored in the "RapidIO Port x Reordering Counter Register". After the value in a programmable threshold is reached, an interrupt is triggered.

For example, if the traffic is time-critical control data, a very low threshold is programmed so that it is not congested for long before the interrupt handler is invoked. The system host can then take action to help ease the congestion.

21.1.3.5 Resetting Performance Registers

The Inbound and Outbound performance registers are both read and writable. These registers are cleared after every read and saturate at the maximum counter values.

21.1.4 Configuring the Tsi620 Switch for Performance Measurements

Performance measurements for complex traffic patterns through the Tsi620 Switch are specified for two different configurations of performance settings. The first configuration is for lightly loaded systems where the likelihood of resource contention is low. This is known as the *fair share* performance configuration.

The second configuration is for congested systems that optimize the throughput, latency, and latency variation of the highest priority packets at the expense of lower priority packets. This is known as the *high priority* performance configuration.

It is expected that configurations different from the two described will have performance figures between the two values specified.

There are many controls in the Tsi620 Switch that allow system designers to optimize their system interconnect performance. These controls consist of the following:

- Clock speeds
- Switch ISF arbitration
- Packet scheduling
- Buffer watermark management

21.1.4.1 Clock Speed Settings

Port speeds directly affect throughput, latency and latency variation. Generally, the slower the port, the lower the throughput, the higher the average latency and the greater the spread between minimum and maximum latency.

- For ports configured in 1x mode, performance measurements are specified for operation at 3.125 Gbaud.
- For ports configured in 4x mode, performance measurements are specified for operation at 3.125 Gbaud per lane.
- All performance measurements assume that the Switch ISF is operating at its maximum frequency of 156.25 MHz.

Performance changes linearly with port and Switch ISF speed.

21.1.4.2 Switch ISF Arbitration Settings

The Switch ISF has three possible settings for its egress arbitration: First Come, First Served, Strict Priority 1, and Strict Priority 2:

- First Come, First Served algorithm – This is used in the fair share performance configuration.
- Strict Priority 1 and 2 – This is used for high priority configuration systems that require the absolute lowest possible latency and latency variation for the highest priority packets, and are willing to tolerate the additional latency and latency variation induced on the lower priority packets. ISF Strict Priority 1 can be used in verification to ensure that ISF Strict Priority 2 delivers optimal performance, but no performance targets are specified against it.

21.1.4.3 Packet Scheduling Settings

The First Come, First Served packet scheduling algorithm is used in fair share systems. In this algorithm, the oldest packet is transmitted. If this packet is retried, then the oldest, highest priority packet is transmitted. The oldest packet is then transmitted again. This leads to increased latency and decreased throughput for higher priority packets since their forward progress depends on the speed with which a lower priority packet can be retried.

21.1.4.4 Buffer Watermark Management Settings

Buffer watermarks restrict the transmission of lower priority packets to the advantage of higher priority packets. Watermark settings directly affect throughput, and indirectly latency and latency variation. For more information on watermarks, see “Egress Watermark”.

The default watermark settings are used for the fair share configuration for both RapidIO ingress and egress buffer management. For high priority configurations, watermark settings are used to deliver maximum throughput for the highest priority packets. For ingress and egress ports, a maximum of six priority 2 packets can be accepted, a maximum of four priority 1 packets can be accepted, and a maximum of two priority 0 packets are accepted.

21.1.5 Port-to-Port Performance Characteristics

The latency and throughput performance measurements of the Tsi620 use port-to-port traffic to characterize the maximum throughput and minimum latency performance. All traffic is of the same size and priority. Only the traffic described in each scenario is present in theTsi620; no other packets are accounted for. Due to the simple traffic pattern, the throughput and latency performance numbers do not change with the priority of the packets or with any of the performance affecting controls of the Tsi620. The latency variation characteristics of port-to-port transfers with these packet models are ignored because they are effectively 0.

21.1.5.1 Port-to-Port Packet Latency

Table 135 shows the 4x and 1x mode latency numbers in cut-through mode under no congestion. The numbers are based on the same ingress and egress port widths and baud rates. The minimum latency is the minimum time for the first bit of an ingress packet to appear at the egress. Due to the multi-clock domain system, the device operates in, the minimum latency can vary by one 312.5 MHz clock period and one reference clock (S_CLK) period.

Table 135: 4x/1x Latency Numbers Under No Congestion

Reference Clock	Ingress and Egress Port Width	Ingress and Egress Baud Rate (Gbaud)	Minimum Latency (ns)[a]
156.25 MHz	4x	3.125	112
		2.5	128.8
		1.25	212.8
	1x	3.125	131.2
		2.5	152.8
		1.25	260.8
125 MHz	4x	2.5	140
		1.25	224
	1x	2.5	172
		1.25	288

a. Due to the asynchronous ability of the clock frequencies within the device, the latency numbers can vary by as much by one 312.5 MHz clock period and one reference clock (S_CLK) period.

The Tsi620 is designed to allow high priority traffic to bypass low priority traffic in periods of contention, as allowed in the *RapidIO Interconnect Specification (Revision 1.3)*.

21.1.6 Packet Throughput Performance

Packet throughput varies from the packet type (for example, NWRITE packets do not require a logical layer response), availability of resources within the device, ability for source and destination of traffic to generate or receive packets, retries of packets, and actual data rates.

A *bubble* is a control symbol inserted by an egress port to maintain the baud rate of the port. The appearance of a bubble indicates that the egress port is under-utilized.

A bubble packet is not the Idle Sequence inserted to maintain link synchronization, as required by the *RapidIO Interconnect Specification (Revision 1.3)*.

21.1.6.1 One Port to One Port Throughput Performance

Under a non-congested port-to-port packet traffic situation, when the ingress and egress have the same line rate (1.25, 2.5, or 3.125 Gbaud), the ingress and egress always maintain the line rates. This means that packet retries do not occur at the ingress ports, and bubbles do not appear in the egress packet stream except for the idle sequence insertion every 5000 code-groups as required by the *RapidIO Interconnect Specification (Revision 1.3)*. This is true for any payload size or packet priority.

When the ingress line rate exceeds that of the egress port, a retry occurs at the ingress port when the buffer is filled to the capacity permitted by the priority of the packets. The egress port still maintains its maximum packet rate with no bubble. This is true for any payload size or packet priority.

21.1.6.2 Many Ports to One Port Throughput Performance

Under a non-congested, many ports to one port packet traffic scenario, when all of the total ingress line rates are the same as the egress line rate (for example, four 1x mode, 3.125 Gbaud ingress ports all going to one 4x mode, 3.125 Gbaud egress port), the ingress port and egress port will always maintain line rates. This means there will be no retry of packets at the ingress and no bubble occurring in the egress packet streams except for the idle sequence insertion every 5000 code-groups required by the *RapidIO Interconnect Specification (Revision 1.3)*. This is true for any payload size and different priorities. The arbitration scheme within the device allocates sufficient bandwidth for each ingress port.

When the total of the ingress line rates exceed that of egress port, retries occur at one or more of the ingress ports if the packet density exceeds the capacity of the egress port. The egress port still maintains its maximum packet rate with no bubble. This is true for any payload size or packet priority.

21.1.6.3 One Port to Many Port Throughput Performance

Under a non-congested one port to many ports packet traffic scenario, when the ingress line rate is the same as the total egress line rates (for example one 4x mode, 3.125 Gbaud ingress port splitting to four 1x mode, 3.125 Gbaud egress port), the ingress and egress ports will always maintain line rates. This means there is no retry of packets in ingress and no bubble-packet in the egress packet streams except for the idle sequence insertion every 5000 code-groups required by the *RapidIO Interconnect Specification (Revision 1.3)*. This is true for any payload size and different priorities. The arbitration scheme within the device divides the traffic according to the egress port bandwidths.

When the ingress line rate exceeds that of the total of the egress ports, retries occur at the ingress port when the packet density exceeds the buffer capacity. The egress ports still maintain their maximum packet rates with no bubble. This is true for any payload size or packet priority.

21.1.7 Multicast Performance

21.1.7.1 Multicast Latency

Since multicast involves more than one egress port and each egress port can have independent traffic conditions, a multicast packet can appear at the destination egress ports at different times. A minimum multicast latency is defined as the shortest time from the arrival of the first bit of a packet at an ingress port that will be multicast, to the appearance of the first bit of the multicast packet at an egress port under no resource contention.

Table 136: 4x/1x Multicast Latency Numbers Under No Congestion

Reference Clock	Ingress and Egress Port Width	Ingress and Egress Baud Rate (Gbaud)	Minimum Latency (ns)[a]
156.25 MHz	4x	3.125	163.2
		2.5	178.4
		1.25	254.4
	1x	3.125	188.8
		2.5	210.4
		1.25	318.4
125 MHz	4x	2.5	204
		1.25	280
	1x	2.5	236
		1.25	344

a. Due to the asynchronous ability of the clock frequencies within the device, the latency numbers can vary by as much as 6.4 ns.

21.1.7.2 Multicast Throughput

The maximum input payload bandwidth of the multicast engine is 10 Gbps. This corresponds to a line rate of 4x mode, 3.125 Gbaud at the ingress port. The maximum input bandwidth of the Multicast Engine can be sourced from one ingress port or multiple ingress ports.

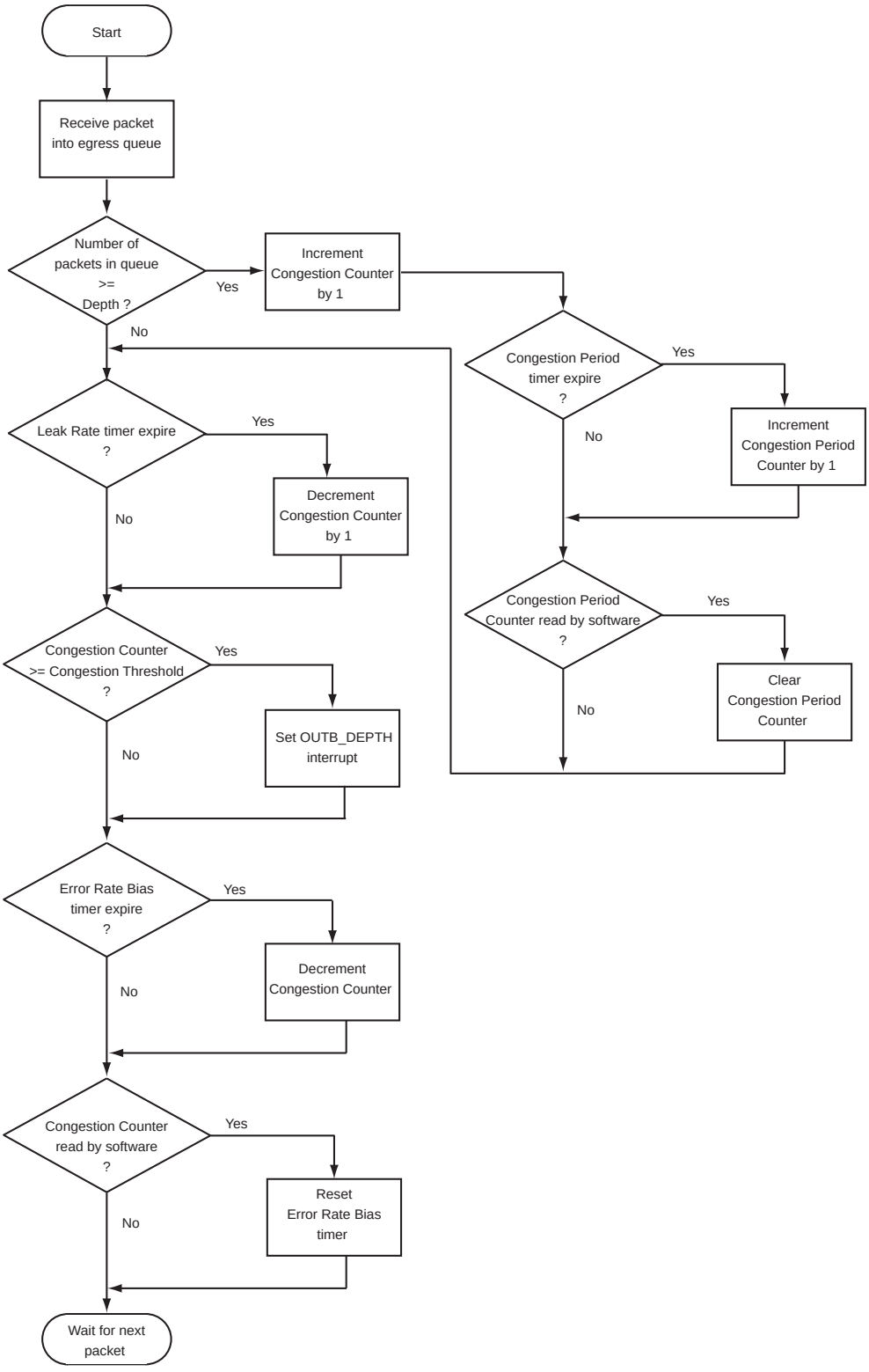
When there is no congestion, and when all destination egress ports have a line rate of 4x mode, 3.125 Gbaud, the egress port always maintains the line rate. There is no bubble-packet in the egress packet streams except for the idle sequence insertion every 5000 code-groups required by the *RapidIO Interconnect Specification (Revision 1.3)*. There is also no retry at the ingress port because the ingress aggregation is handled by the multicast arbitration. This is true for any payload size or packet priority.

When any of the egress ports has a line rate lower than the input bandwidth of the Multicast Engine, retries occur at the ingress port. In this situation, the egress port maintains its line rate. For example, when an egress port is set to 4x mode, 2.5 Gbaud while the multicast engine is receiving a single or aggregated input data at maximum 10 Gbps, retries happen at the ingress port(s). However, the egress port still maintains its line rate with no bubble inserted in that packet stream.

21.1.8 Congestion Detection and Management

The congestion detection and management functionality enables the system management host to monitor the system through a series of registers. The system host can monitor the ingress and egress queue levels and the frequency at which the queues are above the threshold defined by the DEPTH parameter. The behavior and effects of the various tick timers and counters is described in the flow chart shown in Figure 62.

Figure 62: Congestion and Detection Flowchart



21.1.8.1 Congestion Registers

The Tsi620 contains the following port-based registers that can detect and monitor ingress and egress queue levels:

- "RapidIO Port x Transmitter Output Queue Depth Threshold Register": This register contains three fields:
  - Congestion Period (CONG_PERIOD): This value sets the tick interval for the Congestion Period Timer. At the timer expiry, the Congestion Counter is incremented by 1 if the counter value is greater than 0. The Congestion Period Counter indicates the number of tick intervals that the number of packets in the egress queue has exceeded the preset value in the DEPTH field of the register between the current and previous register reads of the Congestion Period Counter.
  - DEPTH: When the number of packets in the egress buffer exceeds the DEPTH threshold, the congestion counter will be incremented.
  - Leak Rate (LEAK_RT): This field is the count for the Leak Rate tick timer. At every tick of the Leak Rate tick timer, the Congestion Counter is decremented.
- "RapidIO Port x Transmitter Output Queue Congestion Period Register": This register contains one field called the Congestion Period Counter, or CONG_PERIOD_CTR. This counter is incremented at every tick whose period is set by the CONG_PERIOD field when the Congestion Counter is greater than 0. This register keeps a running count and is cleared only with a register read.
- "RapidIO Port x Transmitter Output Queue Congestion Status Register": This register contains two fields:
  - Congestion Counter (CONG_CTR): This field keeps a running count of the number of times that the number of packets in the egress queue exceed the DEPTH field setting. The test to increment this counter is performed when a packet has arrived in the egress buffer in its entirety. The test is not synchronous to any clocks.

    CONG_CTR is decremented on Leak Rate timer ticks and on Error Rate Bias timer ticks, and is cleared by writing 1 to OUTB_DEPTH in the "RapidIO Port x Interrupt Status Register".
  - Congestion Threshold (CONG_THRESH): This is the threshold value that if exceeded by the Congestion Counter, sets the OUTB_DEPTH interrupt bit in the "RapidIO Port x Interrupt Status Register".
- "RapidIO Port x Reordering Counter Register": This register contains two fields that track when a packet in an ingress buffer is unable to make forward progress to an egress buffer because the egress buffer cannot accept any more packets of the priority of the received packet. Packet reordering takes place when a packet of higher priority is blocked by a packet of lower priority.
  - Counter (CTR): This is a counter that is incremented each time the Switch ISF selects a packet in the ingress queue that is not at the head of the queue for transmission to an egress buffer.
  - Threshold (THRESH): This field sets the threshold of how many times the Switch ISF can re-order packets before INB_RDR is set in the "RapidIO Port x Interrupt Status Register".

The receiver versions of the registers contain the same fields as the registers related to the transmitters, however the receiver registers pertain to the ingress buffer queue status:

- "RapidIO Port x Receiver Input Queue Depth Threshold Register"
- "RapidIO Port x Receiver Input Queue Congestion Period Register"
- "RapidIO Port x Receiver Input Queue Congestion Status Register"

Interrupts

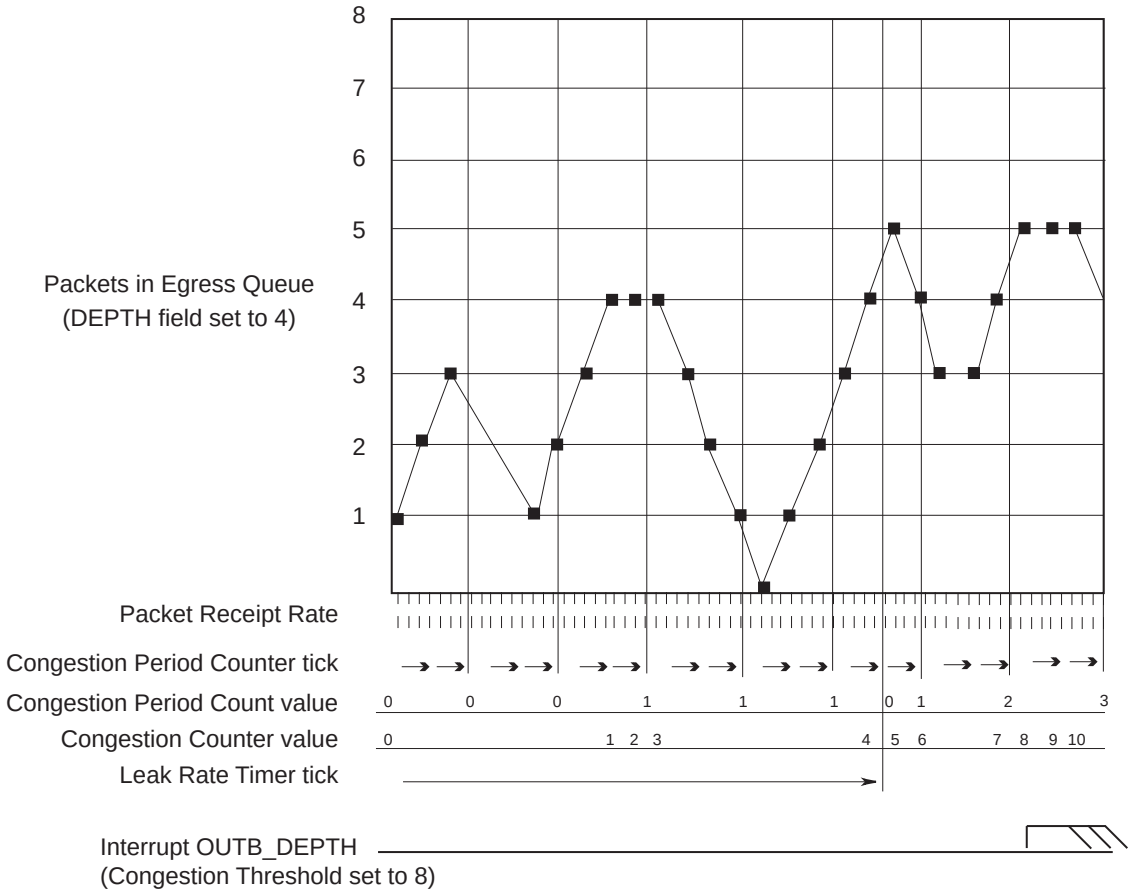
Each port's congestion management register set has two status interrupts, one for the ingress queue depth status (INB_DEPTH) and one for the egress queue status (OUTB_DEPTH). Both status interrupts are located in the "RapidIO Port x Interrupt Status Register".

The Congestion Counter and Congestion Period Counter fields must be polled to determine the current congestion trend if waiting for the interrupts to occur is insufficient warning that the buffers in the Tsi620 Switch have become congested. Also, by tracking the Congestion Counter and Congestion Period Counter values, the traffic trend for the buffer can be determined.

Example of Congestion Register Behavior

Figure 63 is an example of what the congestion registers may contain at various times during port operation. In this example, the number of packets in the egress queue are monitored. The bursting ability of the queue is shown with the frequent increase and decrease of packets. When the queue appears flat it indicates either condition where the number of packets entering the queue is the same as the number of packets leaving the queue appear in this manner, or that queue was stalled during that period.

Figure 63: Congestion Example



The Packet Receipt Rate in the chart indicates how quickly packets can enter and leave the queue. Essentially, this represents the packet line rate. A system with smaller packets increases this receipt rate while a system with predominantly large packages decreases this rate.

The chart also shows the Congestion Period Counter tick and the Leak Rate Timer tick. These ticks occur based on their programmed values in the "RapidIO Port x Transmitter Output Queue Depth Threshold Register". In this example, the DEPTH bit is programmed to four in the "RapidIO Port x Transmitter Output Queue Depth Threshold Register".

Also shown is the Congestion Period Count value (see "RapidIO Port x Transmitter Output Queue Depth Threshold Register") and the Congestion Counter value (see "RapidIO Port x Transmitter Output Queue Congestion Status Register").

In this example, the value of the Congestion Threshold is set to eight in the "RapidIO Port x Transmitter Output Queue Congestion Status Register". When the Congestion Counter equals the value in the Congestion Threshold, the OUTB_DEPTH interrupt is asserted in the "RapidIO Port x Interrupt Status Register". When the OUTB_DEPTH interrupt is asserted, a port-write packet can be generated which causes an in-band notification of the condition that can be routed to any host in the system.

21.2 Tsi620 Bridge Performance

This section discusses performance information for the bridging portion of the Tsi620, both PCI to RapidIO and RapidIO to PCI.

The Tsi620 is designed to maximize throughput and minimize latency between all interfaces. The key performance parameter for the Tsi620 is the clock speed of the SREP module and the Switch ISF. Performance is quoted for the 156.25 MHz reference frequency, which gives the maximum throughput and minimum latency (for more information, see "Clocks").

Throughput and latency figures are provided for read requests, read responses, and writes between each pair of interfaces, as follows:

- RapidIO and PCI
- PCI and RapidIO
- RapidIO to RapidIO, non-transparently bridged

21.2.1 Register Settings

The PCI block does not have any register settings to control performance. The clock speed of the PCI Interface (33 MHz, 66 MHz) does affect throughput and latency.

For purposes of throughput and latency measurements, data path ECC checking and generation has been disabled in the Bridge ISF and the SREP module. In addition, the RapidIO ports operate in cut-through mode.

The reference clock frequency for RapidIO affects throughput of the Bridge ISF and Switch ISF as well as the speed of register request processing. For purposes of throughput and latency measurements, a 156.25 MHz reference clock frequency has been used.

RapidIO flow control makes use of three concepts: priority based reordering, watermarks, and buffer release management.

For more information about Tsi620 flow control, see the *Tsi620 Flow Control Application Note*.

The RapidIO physical layer protocol allows higher priority pakets to be sent/received ahead of lower priority packets: this process is called priority-based reordering. The reordering of higher priority packets ahead of lower priority packets allows for forward progress in the system even when congestion occurs.

The RapidIO protocol requires a buffer management scheme that ensures higher priority packets are accepted ahead of lower priority packets when there is resource contention. Watermarks are control values which determine when packets of a given priority can no longer be accepted. Since priority 3 packets are always accepted when there is a free buffer, watermarks control when priority 0, 1, and 2 packets are not accepted.

Reordering and watermarks can lead to the starvation of lower priority packets under congested scenarios, as only high priority packets can make forward progress. Buffer release management is a method for modulating the congestion in a way that allows an opportunity for lower priority packets to make progress. Buffer release management delays acceptance of more packets until the congestion in a queue/buffer has decreased to a point where lower priority packets can be accepted.

Similar registers control watermarks and buffer release management throughout the SREP module (see “SREP Registers”).

The Europa’s RapidIO ports and SREP module have performance monitoring features (for more information, see “Performance Monitoring”).

21.2.2 Throughput

Two figures are relevant to throughput. The first is the maximum amount of data that can be exchanged; this is expressed as megabytes per second (MBps). The second is the maximum latency for read responses for which maximum throughput can be delivered; this is expressed in nanoseconds (ns).

The throughput data are provided under the following conditions:

- No contention on the Switch ISF
- The only traffic in the system are the transactions given
- PCI target response time is 150 nsec
- RapidIO ports operate in cut-through mode
- RapidIO port width is 4x, lane rate is 3.125 Gbaud

Throughput is provided for a 156.25-MHz reference clock frequency, and is provided for read and write transactions. The throughput figures for read transactions are based on the throughput of the read responses from the target bus. For example, PCI-to-RapidIO read throughput is quoted based on the maximum throughput of the RapidIO port for read responses, and the maximum latency which the RapidIO port can support to maintain that maximum throughput.

21.2.2.1 RapidIO-PCI Bridging Throughput

Table 137: Throughput Between PCI and RapidIO

Test Condition			Target Spec		Actual
Configuration	**Request Direction**	**Payload Size (Bytes)**	**Maximum Throughput (MBps)**	**Maximum Response Latency (ns)**	**Maximum Throughput (MBps)**
PCI: • 66-MHz bus speed • 4 different PCI masters originating PCI read transactions RapidIO: • 4x port, 3.125 GBaud/lane • 34-bit addresses • 8-bit destination IDs • SWRITE packets for writes. Switch ISF • 156.25-MHz reference clock	*RapidIO to PCI* Read	4	26.5	150	78
		256	198.9	150	243
	PCI to RapidIO Read	4	26.5	220	26.3
		256	198.9	2840	133
	RapidIO to PCI Write	4	83.9	N/A	105
		256	237.0	N/A	246
	PCI to RapidIO Write	4	83.9	N/A	69.7
		256	237.0	N/A	234
PCI: • 66-MHz bus speed • 4 different PCI masters originating PCI read transactions RapidIO: • 4x port, 3.125 GBaud/lane • 34-bit addresses • 8-bit destination IDs • SWRITE packets for writes. Switch ISF • 125-MHz reference clock	*RapidIO to PCI* Read	4	26.5	150	74.6
		256	198.9	150	243
	PCI to RapidIO Read	4	26.5	218	26.3
		256	198.9	278	133
	RapidIO to PCI Write	4	83.9	N/A	102
		256	237.0	N/A	246
	PCI to RapidIO Write	4	83.9	N/A	68
		256	237.0	N/A	234

21.2.2.2 RapidIO-to-RapidIO Non-transparent Bridging Throughput

Table 138: Throughput Between RapidIO and RapidIO

Test Condition			Target Spec		Actual
Configuration	**Request Direction**	**Payload Size (Bytes)**	**Maximum Throughput (MBps)**	**Maximum Response Latency (ns)**	**Maximum Throughput (MBps)**
RapidIO: • 4x port, 3.125 GBaud/lane • 34-bit addresses • 8-bit destination IDs • SWRITE packets for writes. Switch ISF • 156.25-MHz reference clock	RapidIO Read	8	397.4	267.6	255
		256	1122.0	4794.2	1092
	RapidIO Write	8	397.4	N/A	260
		256	1122.0	N/A	1085
RapidIO: • 4x port, 3.125 GBaud/lane • 34-bit addresses • 8-bit destination IDs • SWRITE packets for writes. Switch ISF • 125-MHz reference clock	RapidIO Read	8	381.5	266.0	193
		256	897.6	4743.0	881
	RapidIO Write	8	381.5	N/A	196
		256	897.6	N/A	878

21.2.3 Latency

Latency is defined as the time from which the first bit is received on the Tsi620 pins to the time of transfer of the first bit on the Tsi620 pins.

The latency data assume the following:

- RapidIO Interface operates in 4x mode
- 156.25-MHz reference clock for the Switch Fabric, SREP, and the RapidIO MACs
- ECC checking and generation is disabled within the SREP and Bridge ISF
- PCI target will respond with the full 256 bytes in a single transaction, with no retry required

21.2.3.1 Latency for RapidIO-PCI Bridging

Table 139: Latency Between RapidIO and PCI

Test Condition			Target Spec	Simulation
Configuration	**Request Direction**	**Payload Size (Bytes)**	**Latency (ns)**	**Latency (ns)**
RapidIO: • 4x port, 3.125 GBaud/lane • 34-bit addresses • 8-bit destination IDs • SWRITE packets for writes. PCI: • 66-MHz bus speed Switch ISF • 156.25-MHz reference clock Time between first bits of RapidIO packet and PCI FRAME assertion, and vice versa.	*PCI to RapidIO* Read	4	254.1	267
		256	254.1	267
	RapidIO to PCI Read Response	4	278.1	244
		256	479.7	455
	RapidIO to PCI Read	4	254.1	269
		256	254.1	269
	PCI to RapidIO Read Response	4	329.9	331
		256	1269.8	1252
	PCI to RapidIO Write	4	329.9	324
		256	1269.8	1247
	RapidIO to PCI Write	4	260.5	275
		256	458.9	467

21.2.3.2 Latency for RapidIO-to-RapidIO Non-Transparent Bridging

Table 140: Latency for RapidIO Non-Transparent Bridging

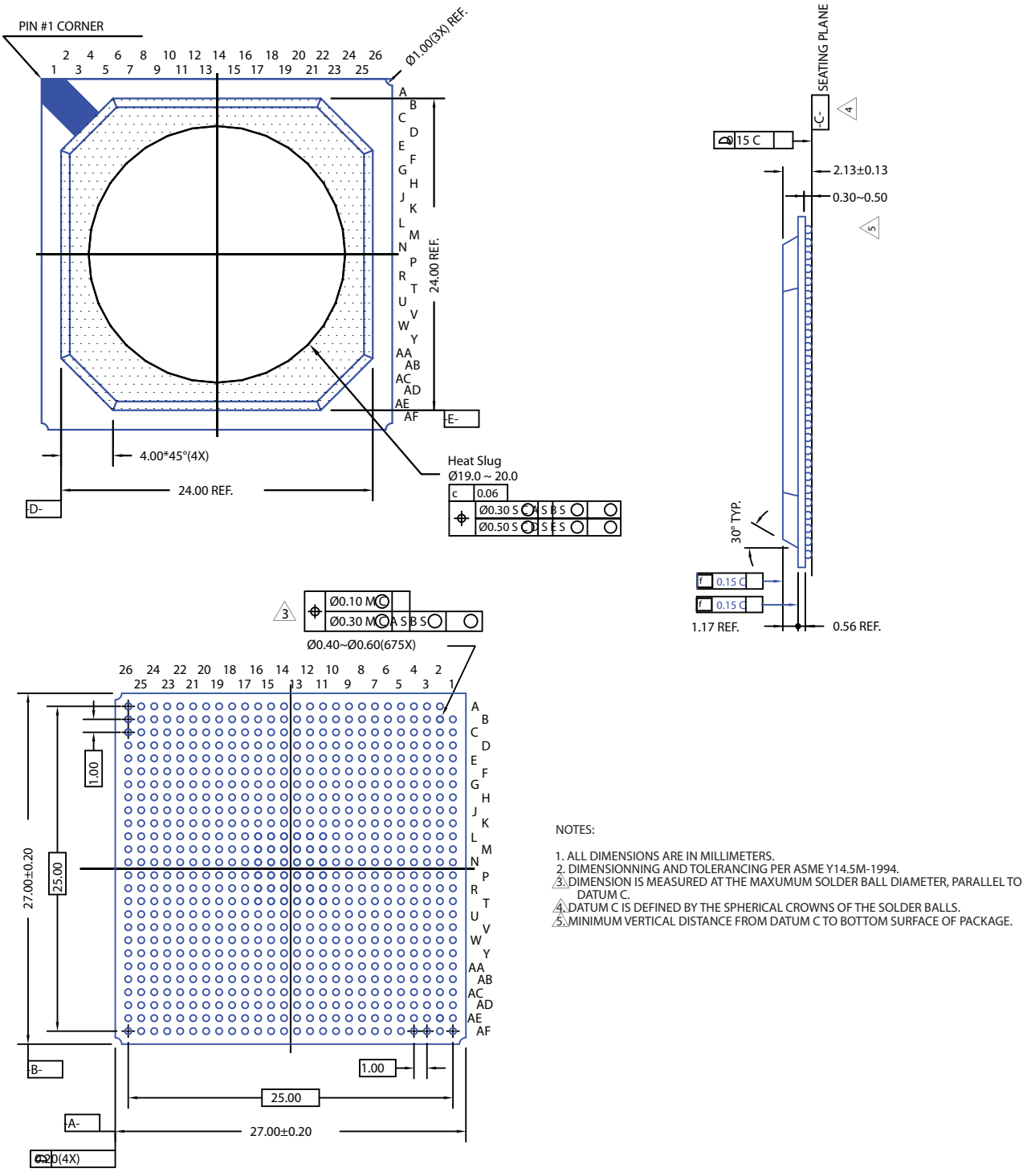
Test Condition			Target Spec	Actual
Configuration	**Request Direction**	**Payload Size (Bytes)**	**Latency (ns)**	**Latency (ns)**
RapidIO: • 4x port, 3.125 GBaud/lane • 34-bit addresses • 8-bit destination IDs • SWRITE packets for writes. Switch ISF • 156.25-MHz reference clock Time between first bits of RapidIO packet in and out.	RapidIO Read	8	246.4	324
		256	246.4	324
	RapidIO Read Response	8	252.8	336
		256	451.2	537
	RapidIO Write	8	252.8	335
		256	451.2	536

22. Packaging

Topics discussed include the following:

22.1 Mechanical Diagrams

Figure 64: Mechanical Diagrams



NOTES:

1. ALL DIMENSIONS ARE IN MILLIMETERS.
2. DIMENSIONNING AND TOLERANCING PER ASME Y14.5M-1994.
3. DIMENSION IS MEASURED AT THE MAXUMUM SOLDER BALL DIAMETER, PARALLEL TO DATUM C.
4. DATUM C IS DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
5. MINIMUM VERTICAL DISTANCE FROM DATUM C TO BOTTOM SURFACE OF PACKAGE.

22.2 Thermal Characteristics

Heat generated by the packaged silicon must be removed from the package to ensure the silicon is maintained within its functional and maximum design temperature limits. If heat buildup becomes excessive, the silicon temperature may exceed the temperature limits. A consequence of this is that the silicon may fail to meet the performance specifications and the reliability objectives may be affected.

Failure mechanisms and failure rate of a device has an exponential dependence on the silicon operating temperatures. Therefore, the control of the package, and by extension the Junction temperature, is essential to ensure product reliability. The Tsi620 is specified safe for operation when the Junction temperature is within the recommended limits as shown in Table 116.

Table 141 shows the simulated thermal characteristic ($Theta_{JB}$ and $Theta_{JC}$) of the Tsi620 package.

Table 141: Thermal Characteristics

Interface	Results
$Theta_{JB}$	7.79°C/W
$Theta_{JC}$	4.43°C/W

Table 142 shows the simulated Junction to Ambient ($Theta_{JA}$) characteristics of the Tsi620. The thermal resistance $Theta_{JA}$ characteristics of a package depends on multiple variables other than just the package. In a typical application, designers must consider various system-level and environmental characteristics, such as:

- Package mounting (vertical/horizontal)
- System airflow conditions (laminar/turbulent)
- Heat sink design and thermal characteristics
- Heat sink attachment method
- PWB size, layer count, and conductor thickness
- Influence of the heat dissipating components assembled on the PWB (neighboring effects)

The results in Table 142 are based on a JEDEC Thermal Test Board configuration (JESD51-9), and does not factor in the system-level characteristics described above. As such, these values are for reference only.

Table 142: Junction to Ambient Characteristics

Device	$Theta_{JA}$ at Specified Airflow (No heat sink)		
	0 m/s	1 m/s	2 m/s
Tsi620	10.43°C/W	9.63°C/W	9.12°C/W

Example of Thermal Data Usage

Based on above $Theta_{JA}$ data and specified conditions, the Junction temperature of the Tsi620 with a 0m/s airflow can be determined using the following formula:

$$T_J = \theta_{JA} * P + T_{AMB}$$

Where:

- T_J is the Junction temperature
- P is the Power consumption
- T_{AMB} is the Ambient temperature

Assuming a power consumption of 4.7W and an ambient temperature of 85°C (industrial), the resultant junction temperature would be 134°C. Assuming a power consumption of 4.7 W and an ambient temperature of 70°C (commercial), the resultant junction temperature for the Tsi620 package would be 119°C.

Results and Recommendation

The results indicate that a Tsi620 package mounted onto a standard test board and submitted to natural and forced convection would exceed the $Theta_{JA}$ specification for industrial applications (8.5°C/W). For commercial applications (10.8°C/W), $Theta_{JA}$ without heat sink would operate under natural and forced convection.

To reduce the thermal resistance value for industrial applications to within the acceptable limit, a heat dissipating device is required.

22.3 Moisture Sensitivity

The moisture sensitivity level (MSL) for the Tsi620 is 3.

23. Ordering Information

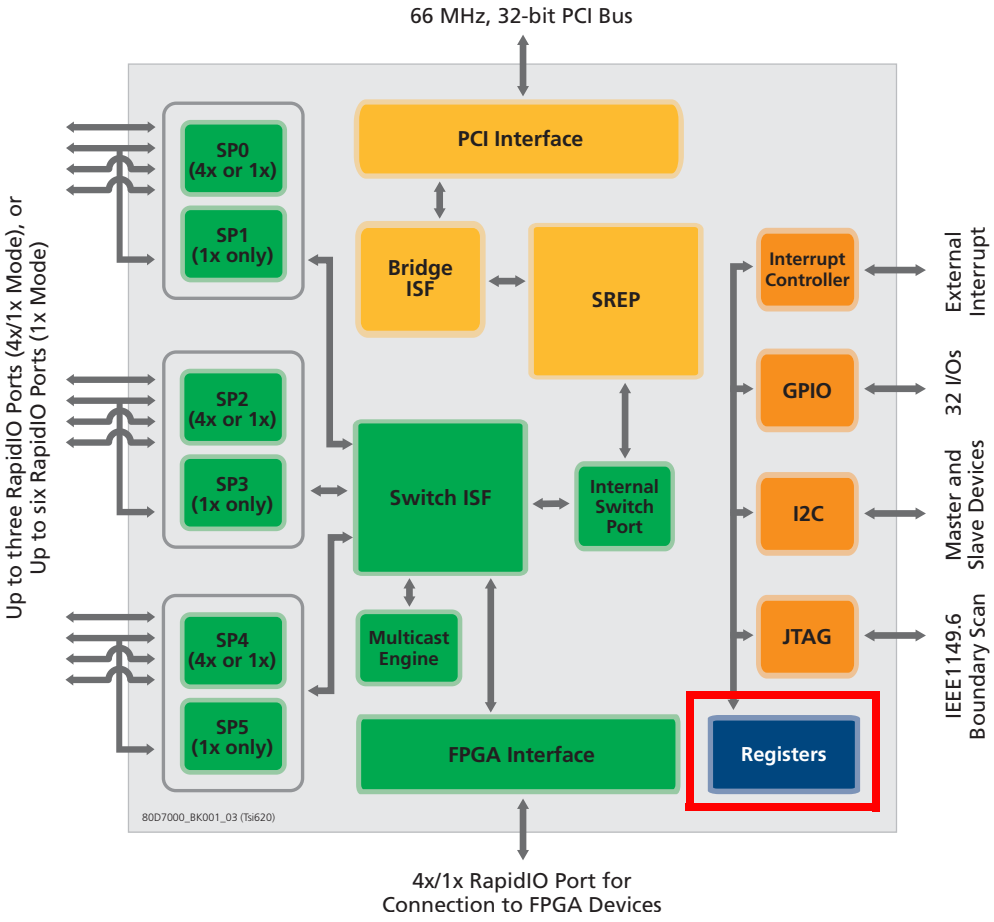
23.1 Part Numbers

Table 143: Part Numbers

Part Number	Frequency/Data Transfer Rate	Temperature	Package	Pin Count
Tsi620-10GCL	10 Gbps	Commercial	Eutectic	675
Tsi620-10GIL	10 Gbps	Industrial	Eutectic	675
Tsi620-10GCLV	10 Gbps	Commercial	RoHS/Green	675
Tsi620-10GILV	10 Gbps	Industrial	RoHS/Green	675

PART 6: REGISTERS

This part of the document describes the Tsi620's device registers. No distinction is made regarding registers that are specific to the Tsi620 Switch, the Tsi620 Bridge, or the Tsi620's secondary functions.



24. Register Access

Topics discussed include the following:

24.1 Overview of Device Register Map

Each Tsi620 interface (PCI, RapidIO) supports two methods of accessing registers: memory-mapped transactions, and register-specific transactions. Registers specific transactions are different depending on the interface:

- Configuration cycles for the PCI Interface
- Maintenance read and write packets for the RapidIO ports

Memory transactions allow access to all Tsi620 registers. The PCI and SREP each allow the configuration of address ranges that can access all Tsi620 registers using memory transactions.

- On PCI, a 256-KB window controlled by the "PFAB_BAR0 Register" and "PFAB_BAR0_UPPER Register" access the Tsi620 registers using PCI memory-mapped transactions. These registers must be initialized using PCI configuration cycles from the PCI side, or RapidIO NREAD/NWRITE/NWRITE_R transactions from the RapidIO side.
- On RapidIO, a 256-KB window controlled by the "SREP Local Configuration Space Base Address CSR" access the Tsi620 registers using RapidIO NREAD/NWRITE/NWRITE_R transactions. This register must be initialized using RapidIO Maintenance transactions from the RapidIO side, or PCI memory-mapped transactions from the PCI side.

The Tsi620 register map is described in Table 145. The addresses of registers are described as offsets from the base address configured in the "PFAB_BAR0 Register" and the "PFAB_BAR0_UPPER Register" and/or the "SREP Local Configuration Space Base Address CSR".

Table 144: Tsi620 Registers Accessed Using Memory Transactions

Interface/Block	Starting Offset	Ending Offset	See
Switch RapidIO Registers	0x0_0000	0x1_A9FC	“RapidIO Registers”
Switch ISF	0x1_AA00	0x1_ABFC	“Switch ISF Registers”
Switch Utility Block	0x1_AC00	0x1_AFFC	“Switch Utility Registers”
Switch ISF Arbitration	0x1_B000	0x1_CFFC	“Multicast Registers”
Reset Control Registers	0x1_AC80	0x1ACBC	“Reset Control Registers”
Event Management Registers	0x1_ACC0	0x1_ACDC	“Event Management Registers”
GPIO Registers	0x1_ACE0	0x1ACFC	“GPIO Registers”
Clock Generator Registers	0x1_AD00	0x1AD7C	“Clock Generator Registers”
I^2C Interface	0x1_D000	0x1_DFFC	“I2C Registers”
Switch SerDes Registers	0x1_E000	0x1_FFFC	“SerDes Per Lane Register”
SREP Registers	0x2_0000	0x2_0FFC	“SREP Registers”
Bridge ISF Registers	0x2_1000	0x2_1FFC	“Bridge ISF Registers”
PCI Interface Registers	0x2_2000	0x2_2FFC	“PCI Registers”
Reserved	0x2_3000	0x3_FFFC	N/A

Register specific transactions can only access registers within the interface. For example, PCI Configuration cycles can only access the configuration space of the PCI block. It is not possible to access RapidIO registers using PCI Configuration cycles. The Tsi620 register map is divided into four groups of registers when accessed by register specific transactions (see Table 145). It is not possible to access all PCI Interface registers using PCI Configuration cycles.

Table 145: Tsi620 Registers Accessed using Register Transactions

Register Group	Interface/Block	Starting Address	Ending Address
Switch (RapidIO Maintenance Request)	Switch RapidIO Registers	0x0_0000	0x1_A9FC
	Switch ISF	0x1_AA00	0x1_ABFC
	Switch Utility Block	0x1_AC00	0x1_AFFC
	Switch ISF Arbitration	0x1_B000	0x1_CFFC
	Reset Control Registers	0x1_AC80	0x1ACBC
	Event Management Registers	0x1_ACC0	0x1_ACDC
	GPIO Registers	0x1_ACE0	0x1ACFC
	Clock Generator Registers	0x1_AD00	0x1AD7C
	I^2C Interface	0x1_D000	0x1_DFFC
	Switch SerDes Registers	0x1_E000	0x1_FFFC
SREP (RapidIO Maintenance Request)	SREP RapidIO Standard Registers	0x0_0000	0x0_02FC
	Reserved	0x0_0300	0x1_02FC
	SREP Implementation- specific Registers	0x1_0300	0x1_0FFC
PCI (PCI Configuration Cycles)	PCI Configuration Space Registers	0x0_0000	0x000FC

Note that the SREP registers are packed into a contiguous 4-KB address space when accessed using memory transactions, but are spread out over 68 KB when accessed using Maintenance packets. The lower 12 bits of a registers address are the same when accessed using memory transactions or maintenance packets. The upper address bits change, depending on the access type used.

All Tsi620 registers can receive initial values during power on initialization through power-up configuration signals, the I^2C Interface and external serial EEPROM; all undefined registers read 0 and a write is ignored.

The Tsi620 registers use direct addressing of 32-bit registers.

An arbiter determines which register bus master is the next allowed on the register bus based on a round-robin scheme. Each master is given equivalent priority on the register bus.

24.1.1 Register Access Sizes

All Tsi620 registers support 4-byte accesses. PCI registers can be accessed from the PCI bus using sub-4-byte accesses. However, PCI accesses to any other registers must be 4 bytes in size.

24.1.2 Reserved Register Addresses and Fields

Reserved register addresses should not be read or written. Reads to reserved register addresses return unspecified data. Writes to reserved register addresses can lead to unpredictable results.

Reserved fields within registers return undefined data when read. Reserved fields should be written as 0 unless noted otherwise.

Table 146 shows the defined register access types.

Table 146: Register Access Types

Abbreviation	Description
R	Read Only Note: R registers are not read/write during the I^2C boot unless it specified in the register description
RS	Read Only, Sticky (only cleared on Power-up Reset) Note: R registers are not read/write during the I^2C boot unless it specified in the register description
R/W	Read or Write Note: All R/W registers are read/write during the I^2C boot.
R/WS	Read or Write, Sticky (only cleared on Powerup Reset) Note: All R/W registers are read/write during the I^2C boot.
R/W1C	Readable. Write 1 to Clear
R/W1CS	Readable. Write 1 to Clear. Sticky (only cleared on Powerup Reset)
R/W0C	Readable. Write 0 to Clear.
R/W1S	Readable. Write 1 to Set (Writing a 1 triggers an event)
RC	Read, then automatically Clear These fields are writable for test purposes.

24.2 Conventions

Register fields are named using “dotted” notation: <registerName>.<fieldName>. For example, the MSB field of the DESTID register is DESTID.MSB.

Often, there are multiple instances of a register (for example, one instance per RapidIO port). Two notations refer to such registers:

- In the first notation, a lower-case letter such as “x” is used as a wildcard character. For example, Sx_DESTID refers to S0_DESTID, S1_DESTID, S2_DESTID, and so on.
- In the second notation, the names of the instances are listed. For example, S{BC,0..2}_DESTID refers to registers SBC_DESTID, S0_DESTID, S1_DESTID, and S2_DESTID.

24.2.1 Endianness

PCI is a little-endian protocol, so the PCI registers are defined in little-endian format. The Bridge ISF registers are defined in little-endian format since they are closely associated with PCI.

RapidIO is a big-endian protocol, so the RapidIO registers are defined in big-endian format. The registers for JTAG, I2C, GPIO, and the Switch ISF are also defined in big-endian format.

RapidIO accesses to PCI registers are byte-swapped. Similarly, PCI accesses to RapidIO registers are byte-swapped.

25. RapidIO Registers

Topics discussed include the following:

25.1 Overview

The application defined Tsi620 registers receive initial values during power on initialization through the I^2C Interface and external serial EEPROM; all undefined registers read 0 and a write is ignored.

The Tsi620 registers use direct addressing of 32-bit registers. The *RapidIO Interconnect Specification (Revision 1.2)* uses 64-bit addressing of registers. Table 147 shows the rules that associate the register offsets in both specifications.

Table 147: Address Rules

Tsi620 Address — Register Offset	RapidIO Specification Address — Register Offset
0x*XXXX*0	0x*XXXX*0, Word 0
0x*XXXX*4	0x*XXXX*0, Word 1

Table 147: Address Rules (Continued)

Tsi620 Address — Register Offset	RapidIO Specification Address — Register Offset
0x*XXXX*8	0x*XXXX*8, Word 0
0x*XXXX*C	0x*XXXX*8, Word 1

25.1.1 Reserved Register Addresses and Fields

Reserved register addresses should not be read or written. Reads to reserved register addresses return unspecified data. Writes to reserved register addresses can lead to unpredictable results.

Reserved fields within registers return undefined data when read. Reserved fields should be written as 0 unless noted otherwise.

Table 148 shows the defined register access types.

Table 148: Register Access Types

Abbreviation	Description
R	Read Only Note: R registers are not read/write during the I^2C boot unless it specified in the register description
R/W	Read or Write Note: All R/W registers are read/write during the I^2C boot.
R/W1C	Readable. Write 1 to Clear
R/W0C	Readable. Write 0 to Clear.
R/W1S	Readable. Write 1 to Set.Writing a 1 triggers an event, bit reads as 0.
RC	Read, then automatically Clear These fields are writable for test purposes.

25.1.2 Ports

The ports are numbered as shown in Table 149.

The odd ports are unavailable when the even port is in 4x mode.

Table 149: RapidIO Port Numbering

Port Number	Description
0	1x/4x Serial port
1	1x Serial port
2	1x/4x Serial port
3	1x Serial port
4	1x/4x Serial port
5	1x Serial port
6	FPGA Interface, 1x/4x Serial port
8	Internal Switch Port, 1x/4x Serial port This port mimics the operation of a RapidIO link to/from the SREP.

25.2 Conventions

Register fields are named using “dotted” notation: <registerName>.<fieldName>. For example, the MSB field of the DESTID register is DESTID.MSB.

Often, there are multiple instances of a register, for example, one instance per RapidIO port. Two notations refer to such registers:

- In the first notation, a lower-case letter such as “x” is used as a wildcard character. For example, Sx_DESTID refers to S0_DESTID, S1_DESTID, S2_DESTID, and so on.
- In the second notation, the names of the instances are listed. For example, S{BC,0..2}_DESTID refers to registers SBC_DESTID, S0_DESTID, S1_DESTID, and S2_DESTID.

The instance number refers to a RapidIO port number. The special instance “BC” (broadcast) refers to a register that when written simultaneously affects all powered-up ports, and that when read returns a value from port number 0.

Port 0 must never be powered-down.

25.3 Register Map

Table 150 gives an overview of the Tsi620 register map.

Table 150: Overview of RapidIO Register Map

Register Group	Start Address	End Address
RapidIO Logical Layer and Transport Layer Registers (Parts I and III of the RapidIO Specification)	0x00000	0x000FC
RapidIO Physical Layer Registers (Parts IV and VI of the RapidIO Specification)	0x00100	0x00FFC
RapidIO Error Management Extension Registers (Part VIII of the RapidIO Specification)	0x01000	0x0FFFF
IDT-Specific RapidIO Registers, and Serial Port Electrical Layer Registers	0x10000	0x15FFC
Per-port copies of RapidIO Global Registers	0x16000	0x1A9FC
Switch ISF Interrupt Registers	0x1AA00	0x1ABFC
Switch Utility registers	0x1AC00	0x1AFFC
Fabric Arbitration Registers	0x1B000	0x1CFFC
Tsi620 "I2C Registers"	0x1D000	0x1DFFC
Tsi620 SerDes Registers	0x1E000	0x1FFFC

Table 151 shows the Tsi620 register map.

Table 151: RapidIO Register Map

Offset	Register Name	See
RapidIO Logical Layer and Transport Layer Registers		
00000	RIO_DEV_ID	"RapidIO Device Identity CAR"
00004	RIO_DEV_INFO	"RapidIO Device Information CAR"
00008	RIO_ASBLY_ID	"RapidIO Assembly Identity CAR"
0000C	RIO_ASBLY_INFO	"RapidIO Assembly Information CAR"
00010	RIO_PE_FEAT	"RapidIO Processing Element Features CAR"
00014	RIO_SW_PORT	"RapidIO Switch Port Information CAR"
00018	RIO_SRC_OP	"RapidIO Source Operation CAR"
0001C - 0002C	Reserved	

Table 151: RapidIO Register Map (Continued)

Offset	Register Name	See
00030	RIO_PE_MC_FEAT	“RapidIO Switch Multicast Support CAR”
00034	RIO_LUT_SIZE	“RapidIO Route LUT Size CAR”
00038	RIO_SW_MC_INFO	“RapidIO Switch Multicast Information CAR”
0003C - 00064	Reserved	
00068	RIO_HOST_BASE_ID_LOCK	“RapidIO Host Base Device ID Lock CSR”
0006C	RIO_COMP_TAG	“RapidIO Component Tag CSR”
00070	RIO_ROUTE_CFG_DESTID	“RapidIO Route Configuration DestID CSR”
00074	RIO_ROUTE_CFG_PORT	“RapidIO Route Configuration Output Port CSR”
00078	RIO_LUT_ATTR	“RapidIO Route LUT Attributes (Default Port) CSR”
0007C	Reserved	
00080	RIO_MC_MASK_CFG	“RapidIO Multicast Mask Configuration Register”
00084	RIO_MC_DESTID_CFG	“RapidIO Multicast DestID Configuration Register”
00088	RIO_MC_DESTID_ASSOC	“RapidIO Multicast DestID Association Register”
0008C - 000FC	Reserved	
RapidIO Physical Layer Registers (using extended features block ID = 0x0009)		
General Physical Layer Registers		
00100	RIO_SP_MB_HEAD	“RapidIO 1x or 4x Switch Port Maintenance Block Header”
00104 - 0011C	Reserved	
00120	RIO_SW_LT_CTL	“RapidIO Switch Port Link Timeout Control CSR”
00124 - 00138	Reserved	
0013C	RIO_SW_GEN_CTL	“RapidIO Switch Port General Control CSR”
Serial Port 0 Registers (Offset 0x140 - 0x15C)		
00140	SP0_LM_REQ	“RapidIO Serial Port x Link Maintenance Request CSR”
00144	SP0_LM_RESP	“RapidIO Serial Port x Link Maintenance Response CSR”
00148	SP0_ACKID_STAT	“RapidIO Serial Port x Local ackID Status CSR”
0014C - 00154	Reserved	
00158	SP0_ERR_STATUS	“RapidIO Port x Error and Status CSR”

Table 151: RapidIO Register Map (Continued)

Offset	Register Name	See
0015C	SP0_CTL	“RapidIO Serial Port x Control CSR”
160 - 17C	Serial Port 1	Same set of registers as Serial Port 0, offset 140 - 15C.
180 - 19C	Serial Port 2	
1A0 - 1BC	Serial Port 3	
1C0 - 1DC	Serial Port 4	
1E0 - 1FC	Serial Port 5	
200 - 21C	Serial Port 6	
220 - 23C	Serial Port 7 (Not Functional), indicates Port Uninitialized	
240 - 25C	Serial Port 8	Same set of registers as Serial Port 0, offset 140 - 15C.
260 - FFC	Tsi620 Reserved	
RapidIO Error Management Extensions (Part VIII of the RapidIO Specification)		
General Error Management Registers		
01000	RIO_ERR_RPT_BH	“RapidIO Error Reporting Block Header Register”
01004	Reserved	
01008	RIO_LOG_ERR_DET	“RapidIO Logical and Transport Layer Error Detect CSR”
0100C	RIO_LOG_ERR_DET_EN	“RapidIO Logical and Transport Layer Error Enable CSR”
01014	RIO_LOG_ERR_ADDR	“RapidIO Logical and Transport Layer Address Capture CSR”
01018	RIO_LOG_ERR_DEVID	“RapidIO Logical and Transport Layer Control Capture CSR”
0101C	RIO_LOG_ERR_CTRL_INFO	“RapidIO Logical and Transport Layer Device ID Capture CSR”
01020 - 01024	Reserved	
01028	RIO_PW_DESTID	“RapidIO Port Write Target Device ID CSR”
0102C - 0103C	Reserved	
Per Port Error Management Registers		
01040	SP0_ERR_DET	“RapidIO Port x Error Detect CSR”
01044	SP0_RATE_EN	“RapidIO Port x Error Rate Enable CSR”
01048	SP0_ERR_ATTR_CAPT_DBG0	“RapidIO Port x Error Capture Attributes CSR and Debug 0 Register”
0104C	SP0_ERR_ATTR_CAPT_0_DBG1	“RapidIO Port x Packet and Control Symbol Error Capture CSR 0 and Debug 1 Register”

Table 151: RapidIO Register Map (Continued)

Offset	Register Name	See
01050	SP0_ERR_ATTR_CAPT_1_DBG2	"RapidIO Port x Packet Error Capture CSR 1 and Debug 2 Register"
01054	SP0_ERR_ATTR_CAPT_2_DBG3	"RapidIO Port x Packet Error Capture CSR 2 and Debug 3 Register"
01058	SP0_ERR_ATTR_CAPT_3_DBG4	"RapidIO Port x Packet Error Capture CSR 3 and Debug 4 Register"
0105C - 1064	Reserved	
01068	SP0_ERR_RATE	"RapidIO Port x Error Rate CSR"
0106C	SP0_ERR_THRESH	"RapidIO Port x Error Rate Threshold CSR"
01070 - 0107C	Reserved	
01080 - 010BC	Serial Port 1 (1x mode)	Same set of registers as for SP0, offsets 0x01040 - 0x0107C.
010C0 - 010FC	Serial Port 2 (1x/4x mode)	
01100 - 0113C	Serial Port 3 (1x mode)	
01140 - 0117C	Serial Port 4 (1x/4x mode)	
01180 - 011BC	Serial Port 5 (1x mode)	
011C0 - 011FC	Serial Port 6 (1x/4x mode)	
01200 - 0123C	Tsi620 Reserved	
01240 - 0127C	Serial Port 8 (1x/4x mode)	Same set of registers as for SP0, offsets 0x01040 - 0x0107C.
01280 - 0FFFC	Tsi620 Reserved	
IDT-Specific RapidIO Registers		
Broadcast Registers (Offset 10000 - 10FFC) - Writing these registers affects all ports. Read data comes from port SP0.		
10000	SPBC_DISCOVERY_ TIMER	"RapidIO Port x Discovery Timer Register"
10004	SPBC_MODE	"RapidIO Port x Mode CSR"
10008	SPBC_CS_INT_STATUS	"RapidIO Port x Multicast-Event Control Symbol and Reset Control Symbol Interrupt CSR"
1000C	SPBC_RIO_WM	"RapidIO Port x RapidIO Watermarks Register"
10010 - 1006C	Reserved	
10070	SPBC_ROUTE_CFG_DESTID	"RapidIO Port x Route Configuration DestID CSR"
10074	SPBC_ROUTE_CFG_ PORT	"RapidIO Port x Route Configuration Output Port CSR"
10078	SPBC_ROUTE_BASE	"RapidIO Port x Local Routing LUT Base CSR"

Table 151: RapidIO Register Map (Continued)

Offset	Register Name	See
1007C - 102FC	Reserved	
10300, 10304, 10308, 1030C, 10310, 10314, 10318, 1031C	RIO_MC_ID{0..7}	"RapidIO Multicast Write ID x Register"
10320, 10324, 10328, 1032C, 10330, 10334, 10338, 1033C	RIO_MC_MASK{0..7}	"RapidIO Multicast Write Mask x Register"
Per Port Instances of the Broadcast Registers (Offset 11000 - 110FC) Writing/reading these registers is a port specific operation.		
11000 - 110FC	Serial Port 0 (1x/4x mode)	Same set of registers as Broadcast Registers, offset 10000 - 100FC.
11100 - 111FC	Serial Port 1 (1x mode)	
11200 - 112FC	Serial Port 2 (1x/4x mode)	
11300 - 113FC	Serial Port 3 (1x mode)	
11400 - 114FC	Serial Port 4 (1x/4x mode)	
11500 - 115FC	Serial Port 5 (1x mode)	
11600 - 116FC	Serial Port 6 (1x/4x mode)	
11700 - 117FC	Tsi620 Reserved	
11800-118FC	Serial Port 8 (1x/4x mode)	Same set of registers as Broadcast Registers, offset 10000 - 100FC.
11900 - 12FFC	Tsi620 Reserved	
Non-Broadcast Per Port Registers		
13000	Reserved	
13004	SP0_CTL_INDEP	"RapidIO Port x Control Independent Register"
13008	Reserved	
1300C	SP0_SEND_MCS	"RapidIO Port x Send Multicast-Event Control Symbol Register"
13010	SP0_LUT_PAR_ERR_INFO	"RapidIO Port x LUT Parity Error Info CSR"
13014	SP0_CS_TX	"RapidIO Port x Control Symbol Transmit Register"
13018	SP0_INT_STATUS	"RapidIO Port x Interrupt Status Register"
1301C	SP0_INT_GEN	"RapidIO Port x Interrupt Generate Register"

Table 151: RapidIO Register Map (Continued)

Offset	Register Name	See
13020	SP0_PSC0n1_CTRL	"RapidIO Port x Performance Statistics Counter 0 and 1 Control Register"
13024	SP0_PSC2n3_CTRL	"RapidIO Port x Performance Statistics Counter 2 and 3 Control Register"
13028	SP0_PSC4n5_CTRL	"RapidIO Port x Performance Statistics Counter 4 and 5 Control Register"
1302C - 1303C	Reserved	
13040	SP0_PSC0	"RapidIO Port x Performance Statistics Counter 0 Register"
13044	SP0_PSC1	"RapidIO Port x Performance Statistics Counter 1 Register"
13048	SP0_PSC2	"RapidIO Port x Performance Statistics Counter 2 Register"
1304C	SP0_PSC3	"RapidIO Port x Performance Statistics Counter 3 Register"
13050	SP0_PSC4	"RapidIO Port x Performance Statistics Counter 4 Register"
13054	SP0_PSC5	"RapidIO Port x Performance Statistics Counter 5 Register"
13058 - 1307C	Reserved	
13080	SP0_TX_Q_D_THRESH	"RapidIO Port x Transmitter Output Queue Depth Threshold Register"
13084	SP0_TX_Q_STATUS	"RapidIO Port x Transmitter Output Queue Congestion Status Register"
13088	SP0_TX_Q_PERIOD	"RapidIO Port x Transmitter Output Queue Congestion Period Register"
1308C	Reserved	
13090	SP0_RX_Q_D_THRESH	"RapidIO Port x Receiver Input Queue Depth Threshold Register"
13094	SP0_RX_Q_STATUS	"RapidIO Port x Receiver Input Queue Congestion Status Register"
13098	SP0_RX_Q_PERIOD	"RapidIO Port x Receiver Input Queue Congestion Period Register"
1309C	Reserved	
130A0	SP0_REORDER_CTR	"RapidIO Port x Reordering Counter Register"
130A4-130AC	Reserved	
130B0	SMAC0_CFG_CH0	"RapidIO SMAC x SerDes Configuration Channel 0 Register"
130B4	SMAC0_CFG_CH1	"RapidIO SMAC x SerDes Configuration Channel 1 Register"
130B8	SMAC0_CFG_CH2	"RapidIO SMAC x SerDes Configuration Channel 2 Register"

Table 151: RapidIO Register Map (Continued)

Offset	Register Name	See
130BC	SMAC0_CFG_CH3	“RapidIO SMAC x SerDes Configuration Channel 3 Register”
130C0	SMAC0_CFG_GBL	“RapidIO SMAC x SerDes Configuration Global Register”
130C8	SMAC0_DLOOP_CLK_SEL	“RapidIO SMAC x Digital Loopback and Clock Selection Register”
130CC	SMAC0_SERDES_OUTPUT_PIN	“RapidIO SMAC SerDes Output Pins Register”
130D0	MCES_PIN_CTRL	“Switch MCES Pin Control Register”
130D4-130FC	Reserved	
13100 - 131AC	Serial Port 1 (1x Mode)	Same set of registers as for SP0, offsets 0x13000 - 0x130AC. The registers at offsets 0x130B0 - 0x130FC are excluded.
13200 - 132FC	Serial Port 2 (1x/4x mode)	All registers as for SP0, offsets 0x13000 - 0x130FC.
13300 - 133AC	Serial Port 3 (1x Mode)	Same set of registers as for SP0, offsets 0x13000 - 0x130AC. The registers at offsets 0x130B0 - 0x130FC are excluded.
13400 - 134FC	Serial Port 4 (1x/4x Mode)	All registers as for SP0, offsets 0x13000 - 0x130FC.
13500 - 135AC	Serial Port 5 (1x Mode)	Same set of registers as for SP0, offsets 0x13000 - 0x130AC. The registers at offsets 0x130B0 - 0x130FC are excluded.
13600 - 136FC	Serial Port 6 (1x/4x Mode)	All registers as for SP0, offsets 0x13000 - 0x130D0. The registers from 136D4 through 136FC are reserved.
13700 - 137C4	Reserved	
137C8	SMAC6_PRBS_CTRL	“RapidIO SMAC 6 PRBS Control Register”
137CC	SMAC6_CH0_PRBS_CTR0	“RapidIO SMAC 6 PRBS Channel 0 Counter 0 Register”
137D0	SMAC6_CH0_PRBS_CTR1	“RapidIO SMAC 6 PRBS Channel 0 Counter 1 Register”
137D4	SMAC6_CH1_PRBS_CTR0	“RapidIO SMAC 6 PRBS Channel 1 Counter 0 Register”
137D8	SMAC6_CH1_PRBS_CTR1	“RapidIO SMAC 6 PRBS Channel 1 Counter 1 Register”
137DC	SMAC6_CH2_PRBS_CTR0	“RapidIO SMAC 6 PRBS Channel 2 Counter 0 Register”
137E0	SMAC6_CH2_PRBS_CTR1	“RapidIO SMAC 6 PRBS Channel 2 Counter 1 Register”
137E4	SMAC6_CH3_PRBS_CTR0	“RapidIO SMAC 6 PRBS Channel 3 Counter 0 Register”
137E8	SMAC6_CH3_PRBS_CTR1	“RapidIO SMAC 6 PRBS Channel 3 Counter 1 Register”
137EC	SMAC6_CH0_BERT_DATA	“RapidIO SMAC 6 BERT Data Register for Channel 0 Register”
137F0	SMAC6_CH1_BERT_DATA	“RapidIO SMAC 6 BERT Data Register for Channel 1 Register”
137F4	SMAC6_CH2_BERT_DATA	“RapidIO SMAC 6 BERT Data Register for Channel 2 Register”

Table 151: RapidIO Register Map (Continued)

Offset	Register Name	See
137F8	SMAC6_CH3_BERT_DATA	"RapidIO SMAC 6 BERT Data Register for Channel 3 Register"
13800 - 138FC	Serial Port 8 (1x/4x Mode)	All registers as for as for SP0, offsets 0x13000 0 - 0x130AC. The registers at offsets 0x130B0 - 0X130FC are excluded, with the exception of offset 130C8.
Per Port Copies of RapidIO Global Registers		
Serial Port 0 RapidIO Global Registers (Offset 0x16000-160FC)		
16000	SP0_RIO_MC_ID0	"RapidIO Serial Port x Multicast Write ID 0 Register"
16004	SP0_RIO_MC_ID1	"RapidIO Serial Port x Multicast Write ID 1 Register"
16008	SP0_RIO_MC_ID2	"RapidIO Serial Port x Multicast Write ID 2 Register"
1600C	SP0_RIO_MC_ID3	"RapidIO Serial Port x Multicast Write ID 3 Register"
16010	SP0_RIO_MC_ID4	"RapidIO Serial Port x Multicast Write ID 4 Register"
16014	SP0_RIO_MC_ID5	"RapidIO Serial Port x Multicast Write ID 5 Register"
16018	SP0_RIO_MC_ID6	"RapidIO Serial Port x Multicast Write ID 6 Register"
1601C	SP0_RIO_MC_ID7	"RapidIO Serial Port x Multicast Write ID 7 Register"
16020	SP0_RIO_RIO_SW_LT_CTL	"RapidIO Serial Port x Switch Port Link Timeout Control CSR"
16028	SP0_RIO_PW_DESTID	"RapidIO Serial Port x Port Write Target Device ID CSR"
1602C	Reserved	
1603C	SP0_RIO_SW_GEN_CTL	"RapidIO Serial Port x Switch Port General Control CSR"
1606C	SP0_RIO_COMP_TAG	"RapidIO Serial Port x Component Tag CSR"
16078	SP0_RIO_LUT_ATTR	"RapidIO Serial Port x Route LUT Attributes (Default Port) CSR"
16100 - 161FC	Serial Port 1 (1x mode)	Same set of registers as Port 0, 16000-160FF.
16200 - 162FC	Serial Port 2 (1x/4x mode)	
16300 - 163FC	Serial Port 3 (1x mode)	
16400 - 164FC	Serial Port 4 (1x/4x mode)	
16500 - 165FC	Serial Port 5 (1x mode)	
16600 - 166FC	Serial Port 6 (1x/4x mode)	
16700 - 167FC	Reserved	
16800 - 168FC	Serial Port 8 (1x/4x mode)	Same set of registers as Port 0, 16000-160FF.

Table 151: RapidIO Register Map (Continued)

Offset	Register Name	See
Fabric Global Interrupt Registers		
1AA00	FAB_CTL	"Switch ISF Control Register"
1AA04	FAB_INT_STAT	"Switch ISF Interrupt Status Register"
1AA08	RIO_MC_LAT_ERR	"Switch ISF Broadcast Buffer Maximum Latency Expired Error Register"
1AA0C	RIO_MC_LAT_ERR_SET	"Switch ISF Broadcast Buffer Maximum Latency Expired Override Register"
1AA10 - 1ABFC	Reserved	
Switch Utility Registers		
1AC00	GLOB_INT_STATUS	"Switch Interrupt Status Register"
1AC04	GLOB_INT_ENABLE	"Switch Interrupt Enable Register"
1AC08 - 1AC10	Reserved	
1AC14	RIO_PW_TIMEOUT	"Switch Port Write Timeout Control Register"
1AC18	RIO_PW_OREQ_STATUS	"Switch Port Write Outstanding Request Register"
1AFFC	Reserved	
Multicast Registers		
1B000	RIO0_MC_REG_VER	"RapidIO Multicast Register Version CSR"
1B004	RIO0_MC_LAT_LIMIT	"RapidIO Multicast Maximum Latency Counter CSR"
1B008	SP0_ISF_WM	"RapidIO Port x Switch ISF Watermarks Register"
1B00C	Reserved	
1B010	SP0_WRR_0	"RapidIO Port x Prefer Unicast and Multicast Packet Priority 0 Register"
1B014	SP0_WRR_1	"RapidIO Port x Prefer Unicast and Multicast Packet Priority 1 Register"
1B018	SP0_WRR_2	"RapidIO Port x Prefer Unicast and Multicast Packet Priority 2 Register"
1B01C	SP0_WRR_3	"RapidIO Port x Prefer Unicast and Multicast Packet Priority 3 Register"
1B020 - 1B0FC	Reserved	

Table 151: RapidIO Register Map (Continued)

Offset	Register Name	See
1B100 - 1B1FC	Serial Port 1 (1x mode)	Same set of registers as Serial Port 0, offset 1B000 - 1B0FC
1B200 - 1B2FC	Serial Port 2 (1x/4x mode)	
1B300 - 1B3FC	Serial Port 3 (1x mode)	
1B400 - 1B4FC	Serial Port 4 (1x/4x mode)	
1B500 - 1B5FC	Serial Port 5 (1x mode)	
1B600 - 1B6FC	Serial Port 6 (1x/4x mode)	
1B700 - 1B7FC	Reserved	
1B800 - 1B8FC	Serial Port 8 (1x/4x mode)	Same set of registers as Serial Port 0, offset 1B000 - 1B0FC
1B900 - 1CFFC	Reserved	
SerDes Per Lane Register		
1E000-1E01C	Reserved	
1E020	SMAC{0,2,4}_PG_CTL_0	“SerDes N Lane 0 Pattern Generator Control Register”
1E024-1E02C	Reserved	
1E030	SMAC{0,2,4}_PM_CTL_0	“SerDes N Lane 0 Pattern Matcher Control Register”
1E034	SMAC{0,2,4}_FP_VAL_0	“SerDes N Lane 0 Frequency and Phase Value Register”
1E038-1E03C	Reserved	
1E040-1E05C	Reserved	
1E060	SMAC{0,2,4}_PG_CTL_1	“SerDes N Lane 1 Pattern Generator Control Register”
1E064-1E06C	Reserved	
1E070	SMAC{0,2,4}_PM_CTL_1	“SerDes N Lane 1 Pattern Matcher Control Register”
1E074	SMAC{0,2,4}_FP_VAL_1	“SerDes N Lane 1 Frequency and Phase Value Register”
1E078-1E07C	Reserved	
1E080-1E09C	Reserved	
1E0A0	SMAC{0,2,4}_PG_CTL_2	“SerDes N Lane 2 Pattern Generator Control Register”
1E0A4-1E0AC	Reserved	
1E0B0	SMAC{0,2,4}_PM_CTL_2	“SerDes N Lane 2 Pattern Matcher Control Register”
1E0B4	SMAC{0,2,4}_FP_VAL_2	“SerDes N Lane 2 Frequency and Phase Value Register”

Table 151: RapidIO Register Map (Continued)

Offset	Register Name	See
1E0B8-1E0BC	Reserved	
1E0C0-1E0DC	Reserved	
1E0E0	SMAC{0,2,4}_PG_CTL_3	“SerDes N Lane 3 Pattern Generator Control Register”
1E0E4-1E0EC	Reserved	
1E0E0	SMAC{0,2,4}_PM_CTL_3	“SerDes N Lane 3 Pattern Matcher Control Register”
1E0E4	SMAC{0,2,4}_FP_VAL_3	“SerDes N Lane 3 Frequency and Phase Value Register”
1E0E8-1E0EC	Reserved	

25.4 RapidIO Logical Layer and Transport Layer Registers

Every processing element contains a set of capability registers (CARs) that allows another processing element to determine its capabilities through maintenance read operations. All registers are 32 bits wide and are organized and accessed in 32-bit quantities. CARs are read-only. CARs are big-endian — bit 0 is the most significant bit.

A processing element contains a set of command and status registers (CSRs) that allows another processing element to control and determine the status of its internal hardware. All registers are organized and accessed in the same way as the CARs.

All of the registers in this section are defined in the *RapidIO Interconnect Specification (Revision 1.3)*.

When an individual port is powered down, the RapidIO Logical Layer and Transport Layer Registers for that port are read only and return 0.

These registers are reset by the CHIP_RST_b reset input signal, as well as when the Tsi620 performs a self-reset. The registers within a port are also reset by a port reset, performed by powering up and down the port. For more information on Tsi620's reset design and behavior, see "Clock, Reset, Power-up, and Initialization Options". It is possible to override reset values of writable fields, and some read-only fields, using the I^2C register loading capability on boot. For more information on the use of I^2C register loading capability, see "I2C Interface".

The I^2C register loading capability may be used during a block or chip reset ("Resets").

Reads to reserved register addresses return 0, while writes to reserved register addresses complete without error and do not affect the operation of the Tsi620.

25.4.1 RapidIO Device Identity CAR

This register identifies the device and vendor information for the Tsi620.

Register name: RIO_DEV_ID **Reset value: 0x**0575**_000D**	**Register offset: 00000**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	DEV_ID							
07:15	DEV_ID							
16:23	DEV_VEN_ID							
24:31	DEV_VEN_ID							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:15	DEV_ID	Device Identifier This field contains the IDT-assigned part number of the device.	R	0x0575
16:31	DEV_VEN_ID	Device Vendor Identifier This field identifies vendor identification number. IDT as the vendor that manufactured the device. This value is assigned by the RapidIO Trade Association.	R	0x000D

25.4.2 RapidIO Device Information CAR

The SILICON_REV and METAL_REV fields in this register identify the stepping of the device.

Register name: RIO_DEV_INFO Reset value: Undefined	Register offset: 00004

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							
24:31	SILICON_REV				METAL_REV			

Bits	Name	Description	Type	Reset Value
0:23	Reserved	N/A	R	0
24:27	SILICON_REV	Indicates the version of silicon used in the device. This value may change with different silicon revisions of the device. 0 = A revision 1 = B revision Note: This field has the same value as the RapidIO Device Information CAR registers in the SREP, and the PCI Revision ID register.	R	Undefined
28:31	METAL_REV	Indicates the version of the metal layers for the current silicon version. This value may change with different metal revisions of the device. Note: This field has the same value as the RapidIO Device Information CAR registers in the SREP, the PCI Revision ID, and the I2C Device ID register.	R	Undefined

25.4.3 RapidIO Assembly Identity CAR

This register contains assembly identification information about the Tsi620.

Register name: RIO_ASBLY_ID Reset value: 0x0001_000D	Register offset: 00008

Bits	0	1	2	3	4	5	6	7
00:07	ASBLY_ID							
08:15	ASBLY_ID							
16:23	ASBLY_VEN_ID							
24:31	ASBLY_VEN_ID							

Bits	Name	Description	Type	Reset Value
0:15	ASBLY_ID	Assembly ID. Identifies the type of assembly from the vendor specified by the ASBLY_VEN_ID field. I^2C load from EEPROM	R	0x0001
16:31	ASBLY_VEN_ID	Assembly Vendor ID Identifies the vendor that manufactured the assembly or subsystem that contains the device. I^2C load from EEPROM	R	0x000D

25.4.4 RapidIO Assembly Information CAR

This register contains additional information about the assembly.

Register name: RIO_ASBLY_INFO Reset value: 0x0000_0100	Register offset: 0000C

Bits	0	1	2	3	4	5	6	7
00:07	ASBLY_REV							
08:15	ASBLY_REV							
16:23	EXT_FEAT_PTR							
24:31	EXT_FEAT_PTR							

Bits	Name	Description	Type	Reset Value
0:15	ASBLY_REV	Assembly Revision Level I^2C load from EEPROM.	R	0x0000
16:31	EXT_FEAT_PTR	Extended Features Pointer This is the pointer to the first entry in the extended features list. In the Tsi620 it points to the Serial Physical Layer (see “RapidIO Physical Layer Registers”).	R	0x0100

25.4.5 RapidIO Processing Element Features CAR

This register identifies the major functionality provided by the processing element.

Register name: RIO_PE_FEAT Reset value: 0x1000_051F	Register offset: 00010

Bits	0	1	2	3	4	5	6	7
00:07	BRDG	MEM	PROC	SW	Reserved			
08:15	Reserved							
16:23	Reserved					MC	Reserved	SBR
24:31	Reserved			CTLS	EXT_FEA	EXT_AS		

Bits	Name	Description	Type	Reset Value
0	BRDG	Bridge 0 = Processing element is not a bridge 1 = Processing element can bridge to another interface	R	0
1	MEM	Endpoint 0 = Not a RapidIO endpoint addressable for reads and writes 1 = The processing element has physically addressable local address space and can be accessed as an endpoint through non-maintenance (that is, NREAD and NWRITE) transactions.	R	0
2	PROC	Processor 0 = Not a processor 1 = Physically contains a local processor or similar device that executes code. A device that bridges to an interface that connects to a processor does not count (see bit 0).	R	0
3	SW	Switching Capabilities The processing element can bridge to another external RapidIO port. For example, a device with two RapidIO ports and a local endpoint is a two port switch, not a three port switch, regardless of its architecture. 0 = Processing element is not a switch 1 = Processing element is a switch. Ftype 8 packets with hop count equal to 0 are routed to the register bus. Note: While this setting is true for SMAC ports and the FPGA Interface, the Internal Switch Port does not support reception of Maintenance packets with hop count of 0.	R	1
4:20	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
21	MC	Multicast 0 = Does not support the multicast extensions 1 = Supports the multicast extensions	R	1
22	Reserved	N/A	R	0
23	SBR	System bringup register extension 0 = System bringup register extension is not supported 1 = System bringup register extension is supported	R	1
24:26	Reserved	N/A	R	0
27	CTLS	For the Tsi620, packets are forwarded according to the configuration of the ingress port's lookup table. This bit is not used in the control of any functionality in the Tsi620. 0 = Device supports 8-bit destination IDs only 1 = Device supports 8- and 16-bit destination IDs	R	1
28	EXT_FEA	Extended Features Pointer is valid Pointer to the first entry in the extended features list.In the Tsi620 this pointer points to the "RapidIO Physical Layer Registers".	R	1
29:31	EXT_AS	Extended Addressing Support. 001 = 34-bit addresses 011 = 50- and 34-bit addresses 101 = 66- and 34-bit addresses 111 = 66-, 50-, and 34-bit addresses	R	0b111

25.4.6 RapidIO Switch Port Information CAR

This register defines the switching capabilities of a processing element.

Register name: RIO_SW_PORT Reset value: Undefined	Register offset: 00014

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved							
16:23	PORT_TOTAL							
24:31	PORT_NUM							

Bits	Name	Description	Type	Reset Value
0:15	Reserved	N/A	R	0x0000
16:23	PORT_TOTAL	Port Total The total number of RapidIO ports on the device. Note that the number of ports reported in this register assumes that all RapidIO ports are in 1x mode. For example, when a port is configured for 4x mode it consumes two ports from this reported number.	R	0x09
24:31	PORT_NUM	Port Number The port number that received the maintenance read packet that caused this register to be read. This value is undefined if the register is read through JTAG and I^2C. When using memory-mapped transactions from the SREP and PCI, this value is 8. For more information on the Tsi620 RapidIO port mapping, see “Ports”.	R	Undefined

25.4.7 RapidIO Source Operation CAR

This register defines the set of RapidIO I/O logical operations that can be issued by the Tsi620. The device can generate I/O logical maintenance read and write requests if it is required to access CARs and CSRs in other processing elements. The Tsi620 can route any packet.

Register name: RIO_SRC_OP **Reset value: 0x0000_0004**	**Register offset: 00018**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved						IMPLEMENT_DEF	
16:23	READ	WRITE	STRM_WR	WR_RES	D_MSG	DBELL	Reserved	A_TSWAP
24:31	A_INC	A_DEC	A_SET	A_CLEAR	Reserved	PORT_WR	Reserved	

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:13	Reserved	N/A	R	0
14:15	IMPLEMENT_ DEF	Implementation defined	R	0
16	READ	Read operation supported	R	0
17	WRITE	Write operation supported	R	0
18	STRM_WR	Streaming write operation supported	R	0
19	WR_RES	Write-with-response operation supported	R	0
20	D_MSG	Data messaging	R	0
21	DBELL	Doorbell	R	0
22	Reserved	N/A	R	0
23	A_TSWAP	Atomic (test-and-swap) operation supported	R	0
24	A_INC	Atomic (increment) operation supported	R	0
25	A_DEC	Atomic (decrement) operation supported	R	0
26	A_SET	Atomic (set) operation supported	R	0
27	A_CLEAR	Atomic (clear) operation supported	R	0
28	Reserved	N/A	R	0
29	PORT_WR	Port-Write operation The RapidIO ports support port-write generation to report errors.	R	1
30:31	Reserved	Implementation defined	R	0

25.4.8 RapidIO Switch Multicast Support CAR

This register identifies the multicast programming model supported by a RapidIO switch. The Tsi620 does not support the simple programming model (see the "RapidIO Multicast Mask Configuration Register").

Register name: RIO_PE_MC_FEAT **Reset value: 0x0000_0000**	**Register offset: 00030**

Bits	0	1	2	3	4	5	6	7
00:07	SIMP	Reserved						
08:15	Reserved							
16:23	Reserved							
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0	SIMP	0 = Does not support simple block programming model	R	0
1:31	Reserved	N/A	R	0

25.4.9 RapidIO Route LUT Size CAR

This register tells host software that the Tsi620 supports 512 destination IDs in its Destination ID LUT. When the "RapidIO Port x Mode CSR" is set to 0, the corresponding RapidIO port supports 64-KB destination IDs with limited capabilities

Register name: RIO_LUT_SIZE **Reset value: 0x0000_01FF**	**Register offset: 0034**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved							
16:23	LUT_SIZE[0:7]							
24:31	LUT_SIZE[8:15]							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:15	Reserved	Reserved	R	0
16:31	LUT_SIZE	Lookup Table Size Identifies the destination IDs that can route packets through the Tsi620 Switch. Destination IDs 0x0000 to LUT_SIZE are valid. Note that when LUT_512 in SPx_MODE is set to 0, Destination IDs 0x0000 to 0xFFFF are valid on the corresponding port, regardless of the LUT_SIZE value.	R	0x01FF

25.4.10 RapidIO Switch Multicast Information CAR

This RapidIO standard register gives information about the multicast programming model, the number of multicast destination IDs supported, and the number of multicast masks supported.

Register name: RIO_SW_MC_INFO Reset value: 0x0000_0008	Register offset: 00038

Bits	0	1	2	3	4	5	6	7
00:07	ASSOC_ MODE	ASSOC_ SCOPE	MAX_DESTID_ASSOC					
08:15	MAX_DESTID_ASSOC							
16:23	MAX_MASKS							
24:31	MAX_MASKS							

Bits	Name	Description	Type	Reset Value
0	ASSOC_MODE	Defines the capabilities of a RapidIO switch to associate destination IDs with multicast masks. 0 = Single associate. One “associate” write is required per association between a DestID and a multicast mask.	R	0
1	ASSOC_SCOPE	Defines the capabilities of a RapidIO switch to associate a destination ID with a multicast mask on a per-inbound-port basis. 0 = A destination ID, when associated with a multicast mask, associates with the mask regardless of which switch inbound port received the packet.	R	0
2:15	MAX_DESTID_ ASSOC	This is the maximum number of Destination IDs that can be associated with any multicast mask. The values are one less than the maximum number of associations. For example, 0x0000 - Maximum of 1 destination ID per multicast mask.	R	0
16:31	MAX_MASKS	Defines the number of multicast masks available. Multicast masks are sequentially numbered, with the first multicast mask number being 0x0000. Additional multicast masks are sequentially numbered. Eight multicast masks available on the Tsi620.	R	8

25.4.11 RapidIO Host Base Device ID Lock CSR

The host base device ID lock CSR contains the base device ID value for the processing element in the system that initializes this processing element.

The HOST_BASE_ID field is a write-once/reset field. Once the HOST_BASE_ID field is written, all subsequent writes to the field are ignored, except when the value written matches the value in the field. In this case, the register is re-initialized to 0xFFFF.

After writing the HOST_BASE_ID field, a processing element must read the Host Base Device ID Lock CSR to verify that it owns the lock before attempting to initialize this processing element.

Register name: RIO_HOST_BASE_ID_LOCK **Reset value: 0x0000_FFFF**	**Register offset: 00068**

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved							
16:23	HOST_BASE_ID[16:23]							
24:31	HOST_BASE_ID[24:31]							

Bits	Name	Description	Type	Reset Value
0:15	Reserved	Reserved	R	0x0000
16:31	HOST_BASE_ID	Base Device ID for the processing element that is initializing this processing element. Note: Setting this field does not enforce exclusive access to the device. It coordinates device identification during initialization and discovery.	R/W	0xFFFF

The HOST_BASE_ID set in this register does not enforce exclusive access to the device. It coordinates device identification during initialization and discovery.

Writing a value of 0x0000_FFFF to this register will lock the HOST_BASE_ID field. As a result, the register is not compliant with the *RapidIO Interconnect Specification (Revision 1.2)*.

25.4.12 RapidIO Component Tag CSR

This register is written by software. It is used for labeling and identifying the port-write transactions to the host.

Note that there are per port copies of this register, described in “RapidIO Serial Port x Component Tag CSR”

Register name: RIO_COMP_TAG **Reset value: 0x0000_0000**	**Register offset: 0006C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	CTAG[0:7]							
08:15	CTAG[8:15]							
16:23	CTAG[16:23]							
24:31	CTAG[24:31]							

Bits	**Name**	**Description**	**Type**	**Reset Value**
00:31	CTAG	Component Tag	R/W	0

25.4.13 RapidIO Route Configuration DestID CSR

This register and "RapidIO Route Configuration Output Port CSR" operate together to provide indirect read and write access to the destination ID lookup tables (LUTs).

Writes to the LUTs through these registers affect the LUTs of all ports on the device. Reads from these registers return the data from Port 0.

This register set is identical to "RapidIO Port x Route Configuration DestID CSR" and "RapidIO Port x Route Configuration Output Port CSR", except that these register are per-port configuration and they include an auto-increment bit to increment the contents of "RapidIO Port x Route Configuration DestID CSR" after a read or write operation.

For information on how to configure the LUTs using this register, see "Lookup Tables".

Register name: RIO_ROUTE_CFG_DESTID **Reset value: 0x0000_0000**	**Register offset: 0070**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved							
16:23	LRG_CFG_DEST_ID							
24:31	CFG_DEST_ID							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:15	Reserved	Reserved	R	0
16:23	LRG_CFG_DEST_ID	Large Configuration Destination ID This field specifies the most significant byte of the destination ID used to select an entry in the LUT, when the "RapidIO Route Configuration Output Port CSR" register is read or written.	R/W	0x00
24:31	CFG_DEST_ID	Configuration Destination ID This field specifies the destination ID used to select an entry in the LUT when the "RapidIO Route Configuration Output Port CSR" register is read or written.	R/W	0x00

25.4.14 RapidIO Route Configuration Output Port CSR

This register and "RapidIO Route Configuration DestID CSR" operate together to provide indirect read and write access to the LUTs.

Writes to the LUTs through these registers affect the LUTs of all ports on the device. Reads from these registers return the data from Port 0.

This register set is identical to "RapidIO Port x Route Configuration DestID CSR" and "RapidIO Port x Route Configuration DestID CSR" except that "RapidIO Port x Route Configuration DestID CSR" are per-port configuration registers and they include an auto-increment bit to increment the contents of "RapidIO Port x Route Configuration DestID CSR" after a read or write operation.

For information on how to configure the LUTs using this register, see "Lookup Tables".

Register name: RIO_ROUTE_CFG_PORT **Reset value: Undefined**	**Register offset: 0074**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							
24:31	PORT							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:23	Reserved	Reserved	R	0
24:31	PORT	Port This is the RapidIO output port through which all reads and writes for "RapidIO Route Configuration DestID CSR".CONFIG_DESTID are sent. Writing a value greater than RIO_SW_PORT.PORT_TOTAL to this field sets the LUT entry to an unmapped state. For compatibility with future IDT devices, write the value 0xFF to indicate an unmapped DestID. When reading an unmapped value from the LUT, this field is set to 0xFF.	R/W	Undefined

25.4.15 RapidIO Route LUT Attributes (Default Port) CSR

This register provides a default route for packets that do not match a valid entry in the destination ID lookup table (LUT). By default the default route is unmapped, and packets that attempt to use the default route are discarded..

Note that there are per port copies of this register, described in “RapidIO Serial Port x Route LUT Attributes (Default Port) CSR”.

Register name: RIO_LUT_ATTR **Reset value: 0x0000_00FF**	**Register offset: 0078**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							
24:31	DEFAULT_PORT							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:23	Reserved	Reserved	R	0
24:31	DEFAULT_ PORT	Default Output Port All transactions with destination IDs not defined in the LUT are routed to this predefined default output port. DEFAULT_PORT can be set to “unmapped” with a value greater than RIO_SW_PORT[PORT_TOTAL]. For compatibility with future IDT devices, it is recommended that the value 0xFF be used to indicate “unmapped”. If a packet needs to consult the default route and the default route is “unmapped”, the packet is discarded. For a mapping of port numbers to physical ports, see “RapidIO Port Numbering”.	R/W	0xFF

25.4.16 RapidIO Multicast Mask Configuration Register

This register adds and removes egress port numbers to multicast masks. This can be completed either before or after a mask is bound to a DestID and placed in the multicast group table. This register can also retrieve the current configuration of a multimask mask.

Register name: RIO_MC_MASK_CFG **Reset value: 0x0000_0000**	**Register offset: 00080**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved					MC_MASK_NUM		
16:23	Reserved				EG_PORT_NUM			
24:31	Reserved	MASK_CMD			Reserved			PORT_ PRESENT

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:12	Reserved	N/A	R	0
13:15	MC_MASK_NUM	Specifies the multicast mask that is to be modified when this register is written with the MASK_CMD field set to “Add” or “Delete“. Specifies the Multicast mask that is to be queried for the presence of a port (by a subsequent read of this register) when this register is written with a “Write_to_Verify“command. When the register is read, this field returns the contents that were previously written.	R/W	0
16:19	Reserved	N/A	R	0
20:23	EG_PORT_NUM	When this register is written, specifies the port number to be added or deleted from the Multicast_Mask. This field is ignored when the MASK_CMD field indicates “Delete All Ports” or “Add All Ports”. When the register is read, this field returns the contents that were previously written.	R/W	0
24	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
25:27	MASK_CMD	Specifies the mask action on a write. Contains the last value written for read operations. • 000 = Write_to_Verify. This write is only to set up a Multicast_Mask and Egress_Port_Number for a subsequent read of this register to check the Port_Present bit. No bits are changed in any multicast mask. • 001 = Add the given Egress_Port_Number to the specified Multicast Mask. • 010 = Delete the given Egress_Port_Number from the specified Multicast Mask. • 011 = Reserved • 100 = Delete all egress ports from the specified Multicast Mask. • 101 = Add all egress ports to the specified Multicast Mask. • 110 = Reserved • 111 = Reserved When the register is read, this field returns the contents that were previously written.	R/W	0
28:30	Reserved	N/A	R	0
31	PORT_ PRESENT	0 = Port is not enabled as an outbound port in the specified multicast mask. 1 = Port is enabled as an outbound port in the specified multicast mask. The Multicast_Mask and Egress_Port_Number were specified in a prior write to this register using the “Write_to_Verify” command.	R	0

25.4.17 RapidIO Multicast DestID Configuration Register

This register configures the multicast group table. It contains the association between a destination ID and a multicast mask number. The association is formed or removed only when the RIO_MC_DESTID_ASSOC register is written.

This register associates only one DestID to one mask. Associating ranges of DestIDs to ranges of masks is not supported.

Register name: RIO_MC_DESTID_CFG **Reset value: 0x0000_0000**	**Register offset: 00084**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	DESTID_BASE_LT							
08:15	DESTID_BASE							
16:23	Reserved							
24:31	Reserved					MASK_NUM_BASE		

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:7	DESTID_BASE_LT	The most significant 8-bit8-bits of a 16-bit DestID to be associated with the mask MASK_NUM_BASE. This field is ignored when an 8-bit DestID is being associated with a multicast mask.	R/W	0
8:15	DESTID_BASE	The 8-bit destination ID or the least significant 8 bits of a 16-bit destination ID to be associated with the mask MASK_NUM_BASE.	R/W	0
16:28	Reserved	N/A	R	0
29:31	MASK_NUM_BASE	The multicast mask number to be associated with the destination ID configured above.	R/W	0

25.4.18 RapidIO Multicast DestID Association Register

This register populates and depopulates the multicast group table. When this register is written, the device consults the value in the RIO_MC_DESTID_CFG register to determine which destination ID is associated with which multicast mask (or which association must be removed).

Register name: RIO_MC_DESTID_ASSOC **Reset value: 0x0000_0000**	**Register offset: 00088**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	ASSOC_BLK_SIZE							
08:15	ASSOC_BLK_SIZE							
16:23	INGRESS_PORT							
24:31	LARGE	CMD		RESERVED				ASSOC_ PRESENT

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:15	ASSOC_BLK_ SIZE	This field is ignored. For compatibility with future IDT devices, set this field to 0 when writing the register.	R	0
16:23	INGRESS_ PORT	This field is ignored.	R	0
24	LARGE	0 = The association is for a Small Transport Destination IDs 1 = The association is for a Large Transport Destination IDs This field returns the previously written value when this register is read.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
25:26	CMD	• 00 = Verify Association This read checks the association status, for a given transport type, between a given Destination ID and a multicast mask number. The transport type is specified by the LARGE field. The Multicast Mask number and Destination ID are specified by a prior write to the "RapidIO Multicast DestID Configuration Register". • 01 = Reserved • 10 = Remove Associations This register write removes the associations between a destination ID and multicast mask number. The Multicast Mask number and Destination ID are specified by a prior write to the "RapidIO Multicast DestID Configuration Register". • 11 = Add Associations This register write adds associations between a destination ID and multicast mask number.The Multicast Mask number and Destination ID are specified by a prior write to the "RapidIO Multicast DestID Configuration Register". • This field returns the previously written value when this register is read.	R/W	0
27:30	Reserved	Reserved	R	0000
31	ASSOC_ PRESENT	When read, this bit indicates the association status between a destination ID, given a transport type, and a multicast mask number. The destination ID and multicast mask number are specified by a previous write to the "RapidIO Multicast DestID Configuration Register". The transport type is specified by a previous write to this register with a command of "Write_To_Verify" with the LARGE field set accordingly. 0 = No association present 1 = Association present This bit is reserved on a write to this register.	R	0

25.5 RapidIO Physical Layer Registers

This section specifies the Command and Status Register (CSR) set. All registers in the set are 32-bits long and aligned to a 32-bit boundary. These registers allow an external processing element to determine the capabilities, configuration, and status of a processing element using the LP-Serial physical layer. The registers can be accessed using the maintenance operations defined in the *RapidIO Interconnect Specification (Revision 1.3)*.

When an individual port is powered down, the RapidIO Physical Layer Registers for that port are read only and return undetermined values.

These registers are reset by several sources of Tsi620 reset (see "Resets"). The registers within a port are also reset by a port reset, performed by powering up and down the port. For more information on Tsi620's reset design, see "Clock, Reset, Power-up, and Initialization Options". It is possible to override reset values of writable fields, and some read-only fields, using the I^2C register loading capability on boot. For more information on the use of I^2C Interface register loading capability, see "I2C Interface".

The I^2C register loading capability may be used on a block or chip reset (see "Resets").

Reads to reserved register addresses return 0, while writes to reserved register addresses complete without error and do not affect the operation of the Tsi620

Table 152: Physical Interface Register Offsets

RapidIO Port x Registers, Physical interface		
Port	**Offset**	**Description**
All	0x0100	These registers affect the operation of all ports.
0	0x0140	1x/4x Serial port
1	0x0160	1x Serial port
2	0x0180	1x/4x Serial port
3	0x01A0	1x Serial port
4	0x01C0	1x/4x Serial port
5	0x01E0	1x Serial port
6	0x0200	1x/4x Serial port
7	0x0220	1x Serial port
8	0x0240	1x/4x Serial port

25.5.1 RapidIO 1x or 4x Switch Port Maintenance Block Header

This register contains the block header information.

Register name: RIO_SW_MB_HEAD Reset value: 0x1000_0009	Register offset: 100

Bits	0	1	2	3	4	5	6	7
00:07	EF_PTR							
08:15	EF_PTR							
16:23	EF_ID							
24:31	EF_ID							

Bits	Name	Description	Type	Reset Value
0:15	EF_PTR	Extended Features Pointer Hard-wired pointer to the next block in the features data structure.	R	0x1000
16:31	EF_ID	Hard-wired extended features ID 0x0009 = Switch with software recovery capability	R	0x0009

25.5.2 RapidIO Switch Port Link Timeout Control CSR

This register contains the timeout timer value for all ports on a device. This timeout is for link events such as sending a packet and receiving the corresponding acknowledge, or sending a link-request and receiving the corresponding link-response. The reset value is the maximum timeout interval, and represents between three and six seconds. When Link Timeout is expired the port enters the Output-Error state, as discussed in the *RapidIO Interconnect Specification (Revision 1.3)*. Note that this register controls no functionality for packet transfer between the Tsi620 Switch and the Tsi620 Bridge (Internal Switch Port, Port 8). It is also used in port 0 of the Tsi620 Switch.

Note that there are per port copies of this register, described in "RapidIO Serial Port x Switch Port Link Timeout Control CSR"

Register name: RIO_SW_LT_CTL **Reset value: 0xFFFF_FF00**	**Register offset: 120**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	TVAL							
08:15	TVAL							
16:23	TVAL							
24:31	Reserved							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:23	TVAL	Timeout Interval Value Timeout = (32/F) * TVAL, where F is the register bus frequency. For a 156.25 MHz reference clock frequency, the register bus frequency is 78.125 MHz and the default value of this register gives a timeout of 6.9 seconds. For a 125 MHz reference clock frequency, the register bus frequency is 62.5 MHz and the default value of this register gives a timeout of 8.6 seconds. When TVAL is 0, the timer is disabled.	R/W	0xFFFFFF
24:31	Reserved	N/A	R	0

25.5.3 RapidIO Switch Port General Control CSR

This register applies to all ports on the device. A device has only one copy of the bits in this register. These bits are also accessible through the Port General Control CSR of any other physical layer used on a device.

Note that there are per port copies of this register, described in “RapidIO Serial Port x Switch Port General Control CSR”

Register name: RIO_SW_GEN_CTL **Reset value: 0x0000_0000**	**Register offset: 13C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved		DISC	Reserved				
08:15	Reserved							
16:23	Reserved							
24:31	Reserved							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:1	Reserved	N/A	R	0
2	DISC	Discovered The Tsi620 is located by the processing element that configures the system. 1 = Device discovered by system host 0 = Device not discovered	R/W	0
3:31	Reserved	N/A	R	0

25.5.4 RapidIO Serial Port x Link Maintenance Request CSR

According to the *RapidIO Interconnect Specification (Revision 1.3)* only one link maintenance request can be outstanding at a time. However, this can produce four consecutive link maintenance requests to quickly re-establish a link.

Multiple link maintenance request symbols are generated by the CMD field in this register. An external device can write to this register and generate a link-request control symbol on the corresponding RapidIO port. A read to this register returns the last value written.

If the sends its own link maintenance request, if that Request is outstanding and the CMD field is written to then the register write is ignored. If this register is written twice in rapid succession, it could cause a protocol violation.

If this register does not indicate that the link-request is complete, software must ensure that a period of time equal to the Port Link Timeout period, controlled by the "RapidIO Switch Port Link Timeout Control CSR", has passed before attempting another link maintenance request to avoid protocol violations.

This register can send a reset or a link-request/input status. Both functions are mimiced by the Internal Switch Port and SREP (see "Link Maintenance Functions").

Register name: SP{0,1,2,3,4,5,6,8}_LM_REQ **Reset value: 0x0000_0000**	**Register offset: 140, 160, 180, 1A0, 1C0, 1E0, 200, 240**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							
24:31	Reserved					CMD		

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:28	Reserved	N/A	R	0
29:31	CMD	Command Command to be sent in the link-request control symbol. If read, this field returns the last written value. 011 = Reset. Writing this value causes the device to send four consecutive reset control symbols. 100 = Input-status Other values are reserved, but the device sends a control symbol with the requested value. Note: The Internal Switch Port discards requests to transmit reserved command values.	R/W	0

Writing this register on a port in normal operation affects traffic on that port. This register should only be used on ports in an error state.

25.5.5 RapidIO Serial Port x Link Maintenance Response CSR

This register is accessed by an external RapidIO device. A read of this register returns the status from the last link-request response issued using the "RapidIO Serial Port x Link Maintenance Request CSR". This register is used when mimicing the operation of a link-request/input status (see "Link Maintenance Functions").

Register name: SP{0,1,2,3,4,5,6,8}_LM_RESP **Reset value: 0x0000_0000**	**Register offset: 144, 164, 184, 1A4, 1C4, 1E4, 204, 244**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	RESP_VLD	Reserved						
08:15	Reserved							
16:23	Reserved						ACK_ID_STAT	
24:31	ACK_ID_STAT			LINK_STAT				

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	RESP_VLD	Response Valid 0 = No link-response control symbol received or no link-request/reset transmitted 1 = If the link-request was a link-request/input-status, this bit indicates that the link-response control symbol is received. The LINK_STAT field contains information pertaining to that link response. If link-request was a link-request/reset, this bit indicates that the link-request/reset is transmitted. Note: For link-response control symbols, this bit certifies the availability of data, it does not certify the correctness of the data.	RC	0
1:21	Reserved	N/A	R	0
22:26	ACK_ID_STAT	ackID Status ackID status field from the link-response control symbol. The value of the next ackID expected by the receiver.	R	0
27:31	LINK_STAT	Link Status Link status field from the link-response control symbol. Other values are reserved. 0b00010 = Error 0b00100 = Retry-stopped 0b00101 = Error-stopped 0b10000 = OK	R	0

25.5.6 RapidIO Serial Port x Local ackID Status CSR

A read to this register returns the local ackID for both the inbound and outbound port of the device. This register has no functionality in the Internal Switch Port (Port 8)(see “Link Maintenance Functions”).

Register name: SP{0,1,2,3,4,5,6,8}_ACKID_STAT **Reset value: 0x0000_0000**	**Register offset: 148, 168, 188, 1A8, 1C8, 1E8, 208, 248**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	CLR_PKTS	Reserved		INBOUND				
08:15	Reserved							
16:23	Reserved			OUTSTANDING				
24:31	Reserved			OUTBOUND				

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	CLR_PKTS	Reserved. Setting this bit will prevent any further packets from being accepted by the port.	R/W1S	0
1:2	Reserved	N/A	R	0
3:7	INBOUND	Inbound Acknowledge ID Next expected ackID value for the receive side of the port. Software can write this field to force re-transmission of outstanding unacknowledged packets, in order to manually implement error recovery. Note: The INBOUND value can be initialized through the I^2C Interface. Initializing the INBOUND value from I^2C is required for test purposes only. Unless the INBOUND value is initialized to 0, the device state is not consistent with the state required by the *RapidIO Specification*. It is not possible to exchange packets after a reset if the INBOUND value is other than 0.Note: This field controls no functionality for Port 8.	R/W	0
8:18	Reserved	N/A	R	0
19:23	OUTSTANDING	Outstanding Acknowledge IDs The first unacknowledged ackID. Note: This field controls no functionality for Port 8.	R	0
24:26	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
27:31	OUTBOUND	Outbound Acknowledge ID Next ackID to be transmitted by the port. Software can write this field to force re-transmission of outstanding unacknowledged packets, in order to manually implement error recovery. If an error or retry occurs on the inbound while the write to this register is being processed, the OUTBOUND value is not used for the next packet transmitted. The new OUTBOUND value is used when the port is operating normally, and when the port is in an error state. Note: This field controls no functionality for Port 8.	R/W	0

Changing the OUTBOUND or INBOUND field when there are packets being exchanged with a link partner causes non-deterministic ackID values to be used for all ports except port 8. It is likely that a fatal error due to ackID mismatch will result.

The OUTBOUND and INBOUND fields can only be written when there are no packets outstanding in the transmit queue, and no packets are being exchanged with a link partner.

25.5.7 RapidIO Port x Error and Status CSR

This register contains the port error and status information. This register returns 0x0000001 if it is read when the port is powered down. Only some bits in this register are operational in the Internal Switch Port (see "Link Maintenance Functions").

The Port 7 version of this register returns 0x0000_0001. Only some bits in this register are operational in Port 8.

Register name: SP{0,1,2,3,4,5,6,7,8}_ERR_STATUS **Reset value: 0x0000_0001, 0x0000_0001, 0x0000_0001, 0x0000_0001, 0x0000_0001, 0x0000_0001, 0x0000_0001, 0x0000_0001,0x0000_0002**	**Register offset: 158, 178, 198, 1B8, 1D8, 1F8, 218, 238, 258**

Bits	0	1	2	3	4	5	6	7
00:07	Reserved					OUTPUT_DROP	OUTPUT_FAIL	OUTPUT_DEG
08:15	Reserved			OUTPUT_RE	OUTPUT_R	OUTPUT_RS	OUTPUT_ERR	OUTPUT_ERR_STOP
16:23	Reserved					INPUT_RS	INPUT_ERR	INPUT_ERR_STOP
24:31	Reserved			PORT_W_PEND	Reserved	PORT_ERR	PORT_OK	PORT_UNINIT

Bits	Name	Description	Type	Reset Value
0:4	Reserved	N/A	R	0
5	OUTPUT_DROP	Output port discarded a packet. The packet is dropped when the Error Rate Threshold is reached or when a TEA error has occurred.	R/W1C	0
6	OUTPUT_FAIL	Output Failed Encountered Output port encountered a failed condition, meaning that the failed port error threshold is reached in the "RapidIO Port x Error Rate Threshold CSR".	R/W1C	0
7	OUTPUT_DEG	OUTPUT_DEG Output Degraded Encountered Output port encountered a degraded condition, meaning that the degraded port error threshold is reached in "RapidIO Port x Error Rate CSR".	R/W1C	0
8:10	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
11	OUTPUT_RE	Output Retry-encountered Outbound port encountered a retry condition. Set when bit 13, Output Retry-stopped, is set. Note: This bit is never set in the Internal Switch Port (Port 8).	R/W1C	0
12	OUTPUT_R	Output Retried Outbound port received a packet-retry control symbol and cannot make forward progress. This bit is set when bit 13, Output Retry-stopped, is set, and cleared after receiving a packet-accepted or packet-not-accepted control symbol. Note: This bit is never set in the Internal Switch Port (Port 8).	R	0
13	OUTPUT_RS	Output Retry-stopped Outbound port received a packet-retry control symbol and is in the output retry-stopped state. Note: This bit is never set in the Internal Switch Port (Port 8).	R	0
14	OUTPUT_ERR	Output Error-encountered Outbound port encountered (and possibly recovered from) a transmission error. This bit is set when OUTPUT_ERR_STOP is set.	R/W1C	0
15	OUTPUT_ERR_STOP	Output Error-stopped Outbound port is in the output error-stopped state. “Link Maintenance Functions”	R	0
16:20	Reserved	N/A	R	0
21	INPUT_RS	Input Retry-stopped Inbound port is in the input retry-stopped state. Note: This bit is never set in the Internal Switch Port (Port 8).	R	0
22	INPUT_ERR	Input Error-encountered Inbound port has encountered (and possibly recovered from) a transmission error. This bit is set when INPUT_ERR_STOP is set.	R/W1C	0
23	INPUT_ERR_STOP	Input Error-stopped Inbound port is in the input error-stopped state. “Link Maintenance Functions”	R	0
24:26	Reserved	N/A	R	0
27	PORT_W_PEND	Port-Write Pending Port has encountered a condition that required it to issue an I/O logical port-write maintenance request. Note: This bit is never set for the Internal Switch Port (Port 8).	R/W1C	0
28	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
29	PORT_ERR	Port Error Inbound or Outbound port has encountered an error from which the hardware was unable to recover (fatal error). The following fatal errors are included: • Four link-request tries with link-response, but no outstanding ackID • Four link-request tries with timeout error for link-response • Dead Link Timer expires This bit is never set in the Internal Switch Port (Port 8).	R/W1C	0
30	PORT_OK	Port OK Inbound and Outbound ports are initialized and can communicate with the adjacent device. This bit and bit 31, Port Un-initialized, are mutually exclusive. The operation of this bit depends on different conditions in the FPGA Interface. A port is considered to be OK if the FPGA Interface is receiving a clock signal from the link partner, and the port can initialize. Data alignment and data corruption issues do not affect PORT_OK in the FPGA Interface; the FGPA Interface can only detect corrupted K characters..	R	0, 0, 0, 0, 0, 0, 0, 0, 1
31	PORT_UNINIT	Port Un-initialized Inbound and Outbound ports are not initialized. This bit and bit 30, Port OK, are mutually exclusive. This bit is set to a 1 after reset on all ports except the Internal Switch Port (Port 8). The operation of this bit depends on different conditions in the FPGA Interface (Port 6). A port is considered to be OK if the FPGA Interface is receiving a clock signal from the link partner, and the port can initialize. Data alignment and data corruption issues do not affect PORT_OK in the FPGA Interface. “RapidIO Serial Port x Control CSR”, and “SREP Port Control CSR”.	R	1, 1, 1, 1, 1, 1, 1, 1, 0

25.5.8 RapidIO Serial Port x Control CSR

This register returns a default value when read in power-down mode. This register returns 0x0000001 if it is read when ports 0 through 6 are powered down.

The Port 7 register always returns 0x0000_0001. The Port 8 register has restricted functionality (see "Link Maintenance Functions").

Register name: SP{0,1,2,3,4,5,6,7,8}_CTL **Reset value: Undefined**	**Register offset: 15C, 17C, 19C, 1BC, 1DC, 1FC, 21C, 23C, 25C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	PORT_WIDTH		INIT_PWIDTH			OVER_PWIDTH		
08:15	PORT_DIS	OUTPUT_EN	INPUT_EN	ERR_DIS	MCS_EN	Reserved	ENUM_B	Reserved
16:23	Reserved							
24:31	Reserved				STOP_FAIL_EN	DROP_EN	PORT_LOCKOUT	PORT_TYPE

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:1	PORT_WIDTH	Port Width This field determines the port mode after reset. • 00 = Single-lane port - Port is 1x mode only. • 01 = Four-lane port - Port has 1x/4x mode and can operate in 1x or 4x mode. PORT_WIDTH is defined by the SPx_MODE_SEL pin (where x = 0, 2, 4, 6) as follows: • If the SPx_MODE_SEL pin is high, PORT_WIDTH for SP*x* is 0 and PORT_WIDTH for SP*x+1* is 0. • If the SPx_MODE_SEL pin is low, PORT_WIDTH for SPx is 1 and PORT_WIDTH for SP*x+1* is 0. However, that port SP*x+1* cannot be used. • Note that for the FPGA Interface (port 6), port 7 is not affected by the SP6_MODE_SEL pin. • The Internal Switch Port operates only in 4x mode.	R	Undefined

(Continued)

Bits	Name	Description	Type	Reset Value
2:4	INIT_PWIDTH	Initialization Port Width Current operating mode of the port. This bit is set by hardware when the initialization process is complete, and when the operating width of the port changes (that is, a 4x port degrades to a 1x port). • 000 = 1x port, lane 0 • 001 = 1x port, lane 2 • 010 = 4x port Note: For the FPGA Interface, on reset this field should be the same as what is configured by the SPx_MODE_SEL pin, regardless of what the link partner supports. After a reset, this field should indicate the OVER_PWIDTH control. The current operating mode of the port mimics the operation of OVER_PWIDTH. It is computed by 0b010 XOR OVER_PWIDTH.	R	0
5:7	OVER_PWIDTH	Override Port Width Software port configuration that overrides the hardware size. This field is valid only if the PORT_WIDTH field is set to 01 and the port is operating in 4x mode. It can be configured as a power-up option (I^2C) or forced during normal mode of operation (forced re-initialization). • 000 = No override (stay in current operation mode, either 1x or 4x) • 001 = Reserved • 010 = Force single lane, lane 0 • 011 = Force single lane, lane 2 (Reserved for Port 6) Other values are reserved. Re-initialization occurs on the port if the value of this register is changed. Note: Re-initialization can be forced by the FORCE_REINIT field of the "RapidIO Port x Control Independent Register". Note: Initial port width of the port is set by SPx_MODE_SEL pins at power-up. After that, the SPx_MODE_SEL is ignored and can be controlled by this register. Note: This bit does not affect the transfer of packets between the Switch and the Bridge. Note: For Port 6 (FPGA interface), the value 0b011 is reserved. Programming this field to 0b011 is considered a programming error.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
8	PORT_DIS	Port Disable 0 = Port receivers/drivers are enabled. 1 = Port receivers/drivers are disabled and are unable to receive/transmit to any packets or control symbols. When the port is disabled, there is no data flow to the output drivers. Transmit drivers of a disabled port transmits all zeros. Any data sent to this port sits in the Output Queue. Due to the different electrical interface, the FPGA Interface asserts the 'PHY_DISABLE' pin when PORT_DIS is set, and ignores what is received from the link partner. The clock and data signals continue to be driven. To disable clock and data signals, power down the port as described in "Port Power Down". On Port 8 this bit affects the transfer of packets between the Switch and the Bridge (see "Link Maintenance Functions").	R/W	0
9	OUTPUT_EN	Output Port Transmit Enable 0 = Port is stopped. It is not able to issue any packets. It can only route and respond to maintenance packets. 1 = Port is enabled to issue any packets. Note: In Port 8 (Internal Switch Port), this bit affects the transfer of packets between the Switch and the Bridge (see "Link Maintenance Functions").	R/W	1
10	INPUT_EN	Inbound Port Enable 0 = Inbound port is stopped and only routes or responds to maintenance requests. Other packets generate packet-not-accepted control symbols to force an error condition on the sending device. 1 = Inbound port responds to any packet. Note: In Port 8 (Internal Switch Port), this bit affects the transfer of packets between the Switch and the Bridge (see "Link Maintenance Functions").	R/W	1
11	ERR_DIS	Error Checking Disable. (Physical layer CRC error only) 0 = Enable error checking and recovery 1 = Disable error checking and recovery When this bit is set, retransmission is suppressed for all packets. Note: If error checking is disabled, then corrupt maintenance packets may be accepted by the Tsi620. Even when error checking is disabled, a corrupt maintenance write request is ignored by the registers. If error checking is enabled, corrupt maintenance packets are not accepted. Note: This bit has no effect on the operation of the Internal Switch Port (Port 8).	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
12	MCS_EN	Multicast-Event Participant 0 = Do not forward incoming Multicast-Event control symbols out this port. 1 = Forward incoming Multicast-Event control symbols out this port Note: In Port 8 (Internal Switch Port), this bit affects whether or not MECS received by the Switch are forwarded to the Bridge (see "Link Maintenance Functions").	R/W	0
13	Reserved	N/A	R	0
14	ENUM_B	Enumeration boundary bit, used in system discovery algorithms. This bit does not control any functionality within the Tsi620. The reset value of this bit is 1 for Port 2. For all other ports, the reset value of this bit is 0. Note: This bit does not control any functionality in the Internal Switch Port (Port 8) (see "Link Maintenance Functions").	R/W	Undefined
15:27	Reserved	N/A	R	0
28	STOP_FAIL_EN	Stop on Port Failed Encountered Enable This bit is used with bit 29, DROP_EN, to force the port to stop attempting to send packets to the connected device when the Error Rate Failed Threshold is met or exceeded.	R/W	0
29	DROP_EN	Drop Packet Enable This bit is used with bit 28, STOP_FAIL_EN, to force the output port to drop packets that are acknowledged with a packet-not-accepted control symbol when the Error Rate Failed Threshold is met or exceeded.	R/W	0
30	PORT_LOCKOUT	When cleared, the packets that may be received and issued are controlled by the state of the Output Port Enable and Input Port Enable bits. When set, this port is stopped and is not enabled to issue or receive any packets. The input port can still send and respond to link-requests. According to the *RapidIO Interconnect Specification (Revision 1.3)*, all received packets return packet-not-accepted control symbols to force the sending device to signal an error condition. This is not possible with Port 8, so we set input/output error stopped conditions instead (see "Link Maintenance Functions"). Note: This bit affects the transfer of packets between the Switch and the Bridge (see "Link Maintenance Functions").	R/W	0
31	PORT_TYPE	Port Type 1 = RapidIO port Note that for the FPGA Interface (port 6), even though the physical interface is not a RapidIO standard, because FPGA implements the RapidIO protocol, a value of 1 is still appropriate in this field.	R	1

25.6 RapidIO Error Management Extension Registers

This section describes the registers in the Extended Features block (EF_ID = 0x0007), which is defined in Part VIII of the RapidIO specification. This block allows an external processing element to manage the error status and reporting for a processing element.

These registers are reset by several sources of Tsi620 reset (see "Resets"). The registers within a port are also reset by a port reset, performed by powering up and down the port. It is possible to override reset values of writable fields using the I^2C register loading capability on boot. For more information on Tsi620's reset design and behavior, see "Clock, Reset, Power-up, and Initialization Options". For more information on the use of I^2C Interface register loading capability, see "I2C Interface".

The Logical/Transport Error detect registers are not required for a RapidIO switch. However, a switch's register bus access errors and transport errors are reported per port in bit 0 of the "RapidIO Port x Error Detect CSR". The port's capture registers contain error information.

When a individual port is powered down, the RapidIO Error Management Extension Registers are read only and return 0.

Reads to reserved register addresses return 0, while writes to reserved register addresses complete without error and do not affect the operation of the Tsi620.

Not all Error Management Extension registers are supported in the Tsi620.

Software must not write to reserved addresses and reserved bits in registers should be written with the read value.

All registers are 32-bits and aligned to a 32-bit boundary.

Table 153: Error Management Registers

Port	Offset	Description
All	0x	General Error Management capability registers
SP0	0x1040	1x/4x Serial port
SP1	0x1080	1x Serial port
SP2	0x10C0	1x/4x Serial port
SP3	0x1100	1x Serial port
SP4	0x1140	1x/4x Serial port
SP5	0x1180	1x Serial port
SP6	0x11C0	1x/4x Serial port
SP7	0x1200	1x Serial port
SP8	0x1240	1x/4x Serial port

25.6.1 Port Behavior When Error Rate Failed Threshold is Reached

When the Error Rate Counter (ERR_RATE_CNT field) reaches the enabled Error Rate Failed Threshold (ERR_RFT field), the behavior of the port depends upon the value of the STOP_FAIL_EN bit and the DROP_EN bit. The required behavior is defined in Table 154.

Table 154: RapidIO Serial Port x Control CSR — Bit Settings for Bits 28 and 29

Bit Setting		Port Behavior
Bit 28 — STOP_FAIL_EN **Stop on Port Failed Encountered Enable**	**Bit 29 — DROP_EN** **Drop Packet Enable**	
0	0	The port continues to attempt to transmit packets to the connected device if the Output Failed Encountered bit is set and/or if the Error Rate Failed threshold is met or exceeded.
0	1	The port discards packets that receive a Packet-not-accepted control symbol when the Error Rate Failed Threshold is met or exceeded. Upon discarding a packet, the port sets the Output Packet-dropped bit in the "RapidIO Port x Error and Status CSR". If the output port "heals", the Error Rate Counter falls below the Error Rate Failed Threshold and the output port continues to forward all packets. Note that for Port 8, this behavior is the same as for the '0 0' case.
1	0	The port stops attempting to send packets to the connected device when the Output Failed Encountered bit is set. The output port becomes congested.
1	1	The port discards all output packets without attempting to send when the port's Output Failed Encountered bit is set. Upon discarding a packet, the port sets the Output Packet-dropped bit in the "RapidIO Port x Error and Status CSR".

25.6.2 RapidIO Error Reporting Block Header Register

The error reporting block header indicates the start of the Error Management Extensions registers in the Tsi620.

Register name: RIO_ERR_RPT_BH **Reset value: 0x0000_0007**	**Register offset: 1000**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	EF_PTR							
08:15	EF_PTR							
16:23	EF_ID							
24:31	EF_ID							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:15	EF_PTR	Extended Features Pointer Hard wired pointer to the next block in the data structure. 0000 = Last extended feature block	R	0x0000
16:31	EF_ID	Hard-wired Extended Features ID 0x0007 = EF ID for error management capability	R	0x0007

25.6.3 RapidIO Logical and Transport Layer Error Detect CSR

This register indicates the error that was detected by the Logical or Transport logic layer. Multiple bits may get set in the register if simultaneous errors are detected during the same clock cycle that the errors are logged. Note that for RapidIO switches, the errors detected are limited to maintenance packets (maintenance requests, maintenance responses, and port writes) with a hop count of 0. No other packets reach the logical layer of a switch.

Register name: RIO_LOG_ERR_DET **Reset value: 0x0000_0000**	**Register offset:1008**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved				L_ILL_TRANS	Reserved		
08:15	L_ILL_RESP	L_UNSUP_TRANS	Reserved					
16:23	Reserved							
24:31	Reserved							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:3	Reserved	N/A	R	0
4	L_ILL_TRANS	Illegal Transaction Bit is set when a terminating maintenance (Type 8, hopcount = 0) request transaction was received with one or more of the following conditions: • TTYPE = 0b0101 - 0b1111 • TT code = 0b00/0b01	R/W0C	0
5:7	Reserved	N/A	R	0
8	L_ILL_RESP	Illegal Response A maintenance response was received with a hop count of 0.	R/W0C	0
9	L_UNSUP_TRANS	Unsupported Transaction A port-write transaction was received with a hop count of 0.	R/W0C	0
10:31	Reserved	N/A	R	0

25.6.4 RapidIO Logical and Transport Layer Error Enable CSR

This register contains the bits that control if an error condition locks the Logical/Transport Layer Error Detect and Capture registers, and is reported to the system host using an interrupt and/or a port write. Note that for RapidIO switches, the errors detected are limited to maintenance packets (maintenance requests, maintenance responses, and port writes) with a hop count of 0. This enables detection of these errors. Once enabled, port writes and interrupts can be generated for these sources. No other packets reach the logical layer of a switch.

Register name: RIO_LOG_ERR_DET_EN **Reset value: 0x0000_0000**	**Register offset:100C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved				ILL_TRANS_EN	Reserved		
08:15	ILL_RESP_EN	UNSUP_TRANS_EN	Reserved					
16:23	Reserved							
24:31	Reserved							

Bits	**Name**	**Description**[a]	**Type**	**Reset Value**
0:3	Reserved	N/A	R	0
4	ILL_TRANS_EN	Illegal Transaction Decode Enable 0 = disable L_ILL_TRANS 1 = enable L_ILL_TRANS	R/W	0
5:7	Reserved	N/A	R	0
8	ILL_RESP_EN	Illegal Response Enable 0 = disable L_ILL_RESP 1 = enable L_ILL_RESP	R/W	0
9	UNSUP_TRANS_EN	Unsupported Transaction Enable 0 = disable L_UNSUP_TRANS 1 = enable L_UNSUP_TRANS	R/W	0
10:31	Reserved	N/A	R	0

a. All bits in this register enable bits in "RapidIO Logical and Transport Layer Error Detect CSR".

25.6.5 RapidIO Logical and Transport Layer Address Capture CSR

This register contains error information. It is locked when a Logical/Transport error is detected and the corresponding enable bit is set. Note that for RapidIO switches, the errors detected are limited to maintenance packets (maintenance requests, maintenance responses, and port writes) with a hop count of 0. No other packets reach the logical layer of a switch. Therefore, the only time this register contains correct information is for maintenance requests that have incorrect field values. Only invalid data can be captured in this register for erroneous port-writes and maintenance responses, as these transactions reserve the address field. If the TT code for an erroneous maintenance request is invalid, this register captures the address of the faulty packet. This register is not updated when a correctly formatted maintenance request fails.

For the sourceID where the error originated, see “RapidIO Logical and Transport Layer Device ID Capture CSR”.

Register name: RIO_LOG_ERR_ADDR **Reset value: 0x0000_0000**	**Register offset:1014**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	ADDRESS							
16:23	ADDRESS							
24:31	ADDRESS					Reserved	WDPTR	Reserved

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:7	Reserved	N/A	R	0
8:28	ADDRESS	Address of the illegal maintenance request received.	R/W	0
29	Reserved	N/A	R	0
30	WDPTR	Word pointer from the illegal maintenance request received	R/W	0
31	Reserved	N/A	R	0

25.6.6 RapidIO Logical and Transport Layer Device ID Capture CSR

This register contains error information, specifically the device ID field values for failed transactions. It is locked when a Logical/Transport error is detected and the corresponding enable bit is set. When the TT field of the erroneous message is not a defined value, the contents of this register are bytes 3 and 4 of the packet received. Note that for RapidIO switches, the errors detected are limited to maintenance packets (maintenance requests, maintenance responses, and port writes) with a hop count of 0. No other packets can reach the logical layer of a switch. None of these maintenance packets have a destination ID, only a source ID. This register is not updated when a correctly formatted maintenance request fails.

Register name: RIO_LOG_ERR_DEVID **Reset value: 0x0000_0000**	**Register offset: 1018**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved							
16:23	SRCID_MSB							
24:31	SRCID							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:15	Reserved	N/A	R	0
16:23	SRCID_MSB	Source ID Most Significant Byte Most significant byte of the source ID associated with the error (large transport systems only)	R/W	0
24:31	SRCID	Source ID The sourceID associated with the error.	R/W	0

25.6.7 RapidIO Logical and Transport Layer Control Capture CSR

This register contains error information, specifically the message type and subtype field values for failed transactions. It is locked when a Logical/Transport error is detected and the corresponding enable bit is set. The Ftype value should be 8 for a maintenance packet. The TTYPE value indicates the 4 bits following the FTYPE field in the packet. Note that for RapidIO switches, the errors detected are limited to maintenance packets (maintenance requests, maintenance responses, and port writes) with a hop count of 0. No other packets reach the logical layer of a switch. This register is not updated when a correctly formatted maintenance request fails.

Note: To set FTYPE and TTYPE to expected values for purposes of software testing, the values must be written to the least significant byte of this register.

Register name: RIO_LOG_ERR_CTRL_INFO **Reset value: 0x0000_0000**	**Register offset: 101C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	FTYPE				TTYPE			
08:15	Reserved							
16:23	Reserved							
24:31	Reserved							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:3	FTYPE	Format type associated with the error	R/W	0
4:7	TTYPE	Transaction type associated with the error	R/W	0
8:31	Reserved	N/A	R	0

A value of 0x00000000 must be written to this register to clear it.

25.6.8 RapidIO Port Write Target Device ID CSR

This register contains the target device ID to be used when a device generates a Maintenance Port-write operation to report errors to a system host. Port-Write packets are routed to the output port defined by the routing LUT of the Tsi620 Switch.

This register has no functionality in the Internal Switch Port (Port 8) (see "Link Maintenance Functions"). Note that there are per port copies of this register, described in "RapidIO Serial Port x Port Write Target Device ID CSR"

Register name: RIO_PW_DESTID **Reset value: 0x0000_0000**	**Register offset: 1028**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	DESTID_MSB							
08:15	DESTID_LSB							
16:23	LARGE_ DESTID	Reserved						
24:31	Reserved							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:7	DESTID_MSB	Most Significant Byte of Port-Write Target Device ID. Used only when LARGE_DESTID is 1.	R/W	0
8:15	DESTID_LSB	If LARGE_DESTID is 0, the DESTID_LSB field is the 8-bit DESTID used in locally-generated Port-Write requests. If LARGE_DESTID is 1, the DESTID_LSB field forms the least significant bits of a 16-bit DestID used in locally-generated Port-Write requests.	R/W	0
16	LARGE_DESTID	0 = Port-write transactions are generated with an 8-bit destination ID. 1 = Port-write transactions are generated with a 16-bit destination ID.	R/W	0
17:31	Reserved	N/A	R	0

25.6.9 RapidIO Port x Error Detect CSR

This register indicates transmission errors that are detected by the hardware.

Each write of a non-zero value to the "RapidIO Port x Error Detect CSR" causes the Error Rate Counter to increment, if the corresponding error bit is enabled in the "RapidIO Port x Error Rate Enable CSR". When the threshold is reached, hardware informs the system software of the error using its standard error reporting function. After the error is reported, the system software can read and clear registers as necessary to complete its error handling protocol testing.

Register name: SP{0,1,2,3,4,5,6,8}_ERR_DET **Reset value: 0x0000_0000**	**Register offset: 1040, 1080, 10C0, 1100, 1140, 1180, 11C0, 1240**

Bits	0	1	2	3	4	5	6	7
00:07	IMP_ SPEC_ ERR	Reserved						
08:15	Reserved	CS_CRC_E RR	CS_ILL_ID	CS_NOT_A CC	PKT_ILL_A CKID	PKT_CRC_ ERR	PKT_ILL_ SIZE	Reserved
16:23	Reserved							
24:31	Reserved		LR_ACKID _ILL	PROT_ ERR	Reserved	DELIN_ ERR	CS_ACK_ ILL	LINK_TO

Bits	Name	Description	Type	Reset Value
0	IMP_SPEC_ERR	Implementation Specific Error Detected Logical/Transport error per port. This bit indicates one or more of the following illegal field errors: • Reserved transport type (TT) detected (TT field = 10 or 11 for all but maintenance packets with hop count = 0) • Maximum retry threshold exceeded • Unmapped DestID Error • Parity Error in Lookup Table • Switch ISF TEA Error • Multicast TEA Error • Port Fatal Error Note: To clear this bit and the associated interrupt status bits, the entire register must be written to 0. Note: Only the first TEA or MC_TEA error that changes the "RapidIO Port x Error and Status CSR".OUTPUT_DROP bit from 0 to 1 will be counted towards an Error Rate Degraded Threshold Reached or Error Rate Failed Threshold Reached event. Subsequent TEA or MC_TEA errors which occur while the OUTPUT_DROP bit is 1 do not cause the error rate counter in the "RapidIO Port x Error Rate CSR" to increment. Once the OUTPUT_DROP bit is cleared, the next TEA or MC_TEA error will cause the error rate counter to increment. Note: Clearing this bit also clears the following interrupt status bits: • MAX_RETRY, TEA, MC_TEA, LUT_PAR_ERR, ILL_TRANS_ERR in "RapidIO Port x Interrupt Status Register"	R/W0C	0
1:8	Reserved	N/A	R	0
9	CS_CRC_ERR	Received a control symbol with a CRC error.	R/W0C	0
10	CS_ILL_ID	Received an acknowledge control symbol with an unexpected ackID (packet-accepted or packet_retry). The Capture register does not have valid information during this error detection.[a]	R/W0C	0
11	CS_NOT_ACC	Received packet-not-accepted control symbol.[a]	R/W0C	0
12	PKT_ILL_ACKID	Received packet with unexpected ackID.[a]	R/W0C	0
13	PKT_CRC_ERR	Received a packet with a CRC error.[a]	R/W0C	0
14	PKT_ILL_SIZE	Received packet exceeds 276 bytes.[a]	R/W0C	0
15:25	Reserved	N/A	R	0
26	LR_ACKID_ILL	Link response received with an ackID that is not outstanding. The Capture register does not have valid information during this error detection.[a]	R/W0C	0

(Continued)

Bits	Name	Description	Type	Reset Value
27	PROT_ERR	Protocol Error Received control symbol is unexpected.[a]	R/W0C	0
28	Reserved	N/A	R	0
29	DELIN_ERR	Delineation Error Received unaligned /SC/ or /PD/, or undefined code-group. The Capture register does not capture information for this error. [a]	R/W0C	0
30	CS_ACK_ILL	Received an unexpected acknowledge control symbol[a]	R/W0C	0
31	LINK_TO	An acknowledge or Link-response is not received within the specified timeout interval (see "RapidIO Switch Port Link Timeout Control CSR" register). The Capture register does not capture information for this error.[a]	R/W0C	0

a. This bit is never set by the Tsi620 Internal Switch Port (Port 8).

25.6.10 RapidIO Port x Error Rate Enable CSR

This register contains the bits that control when an error condition is allowed to increment the error rate counter. and be captured in the error capture register.

Each write of a non-zero value to the "RapidIO Port x Error Detect CSR" causes the Error Rate Counter to increment, if the corresponding error bit is enabled in this register. When the threshold is reached, hardware informs the system software of the error using its standard error reporting function. After the error is reported, the system software can read and clear registers as necessary to complete its error handling protocol testing.

Register name: SP{0,1,2,3,4,5,6,8}_RATE_EN **Reset value: 0x0000_0000**	**Register offset: 1044, 1084, 10C4, 1104, 1144, 1184, 11C4, 1244**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	IMP_SPEC_ERR	Reserved						
08:15	Reserved	CS_CRC_ERR_EN	CS_ILL_ID_EN	CS_NOT_ACC_EN	PKT_ILL_ACKID_EN	PKT_CRC_ERR_EN	PKT_ILL_SIZE_EN	Reserved
16:23	Reserved							
24:31	Reserved		LR_ACKID_ILL_EN	PROT_ERR_EN	Reserved	DELIN_ERR_EN	CS_ACK_ILL_EN	LINK_TO_EN

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	IMP_SPEC_ERR	Logical/Transport Error Enable For information on the errors applicable to this field, see "RapidIO Port x Error Detect CSR".	R/W	0
1:8	Reserved	N/A	R	0
9	CS_CRC_ERR_EN	Enable error rate counting. Received Control Symbol with a CRC error.[a]	R/W	0
10	CS_ILL_ID_EN	Enable error rate counting. Received an acknowledge control symbol with an unexpected ackID (packet-accepted or packet_retry).[a]	R/W	0
11	CS_NOT_ACC_EN	Enable error rate counting. Received packet-not-accepted control symbol.[a]	R/W	0
12	PKT_ILL_ACKID_EN	Enable error rate counting. Received packet with not unexpected ackID.[a]	R/W	0
13	PKT_CRC_ERR_EN	Enable error rate counting. Received packet with a CRC error.[a]	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
14	PKT_ILL_SIZE_EN	Enable error rate counting. Received packet exceeds 276 bytes.[a]	R/W	0
15:25	Reserved	N/A	R	0
26	LR_ACKID_ILL_EN	Enable error rate counting. A received Link Response control symbol contains an ackID that is not outstanding.[a]	R/W	0
27	PROT_ERR_EN	Enable error rate counting. Protocol Error. Received Control Symbol is unexpected.[a]	R/W	0
28	Reserved	N/A	R	0
29	DELIN_ERR_EN	Enable error rate counting Delineation Error Received unaligned /SC/or/PD/ or undefined code-group.[a]	R/W	0
30	CS_ACK_ILL_EN	Enable error rate counting An unexpected acknowledge control symbol was received.[a]	R/W	0
31	LINK_TO_EN	Enable error rate counting An acknowledge or Link-response is not received within the specified timeout interval.[a]	R/W	0

a. This error is not detected by the Internal Switch Port (Port 8).

25.6.11 RapidIO Port x Error Capture Attributes CSR and Debug 0 Register

This register indicates the type of information contained in the following registers:

- "RapidIO Port x Packet and Control Symbol Error Capture CSR 0 and Debug 1 Register"
- "RapidIO Port x Packet Error Capture CSR 1 and Debug 2 Register"
- "RapidIO Port x Packet Error Capture CSR 2 and Debug 3 Register"
- "RapidIO Port x Packet Error Capture CSR 3 and Debug 4 Register"

In the case of multiple detected errors during the same clock cycle, only one of the errors must be indicated in the ERR_TYPE field. In debug mode this register is unlocked, and its fields are used for writing the content of the debug packet.

Register name: SP{0,1,2,3,4,5,6,8}_ERR_ATTR_CAPT_DBG0 **Reset value: 0x0000_0000**	**Register offset: 1048, 1088, 10C8, 1108, 1148, 1188, 11C8, 1248**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	INFO_TYPE		Reserved	ERR_TYPE				
08:15	Reserved							
16:23	Reserved							
24:31	Reserved							VAL_CAPT

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:1	INFO_TYPE	Type of information logged. • 00 = Packet • 01 = Control Symbol and unaligned /SC/or/PD/ or undefined code-group • 10 = Implementation specific (capture register contents are implementation-specific to report implementation-specific errors) • 11 = Reserved for RapidIO ports	R/W	0
2	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
3:7	ERR_TYPE	Encoded 5-bit value of captured error bit in the "RapidIO Port x Error Detect CSR". • 00000 = bit 0 (IMP_SPEC_ERR) • 00001 = bit 1 (reserved) • ... • 00111 = bit 8 (reserved) • 01000 = bit 9 (CS_CRC_ERR) • 01001 = bit 10 (CS_ILL_ID) • 01010 = bit 12 (CS_NOT_ACC) •	R/W	0
8:30	Reserved	N/A	R	0
31	VAL_CAPT	Capture Valid Information This bit is set by hardware to indicate that the packet/control symbol capture registers contain valid information. For control symbols, only capture register 0 contains meaningful information. Software writes "0" to clear this bit and unlock all capture registers of port *x*. The capture registers are: • "RapidIO Port x Packet and Control Symbol Error Capture CSR 0 and Debug 1 Register" • "RapidIO Port x Packet Error Capture CSR 1 and Debug 2 Register" • "RapidIO Port x Packet Error Capture CSR 2 and Debug 3 Register" • "RapidIO Port x Packet Error Capture CSR 3 and Debug 4 Register"	R/W0C	0

25.6.12 RapidIO Port x Packet and Control Symbol Error Capture CSR 0 and Debug 1 Register

In debug mode this register is unlocked. It contains bytes 4 to 7 of the debug packet being composed. During normal operation, this register captures bytes 0 to 3 of the packet, or the entire control symbol, that was detected to be in error.

To assist in software testing and debug of the system error recovery and threshold function, the "RapidIO Port x Error Detect CSR" and the Port x Error Capture registers (0x14C, 0x1050, 0x1054, and 0x1058) are also writable. Software must set the Capture Valid Info bit in the Port x Attributes Error Capture CSR, then write the packet/control symbol information to the other capture registers.

Register name: SP{0,1,2,3,4,5,6,8}_ERR_CAPT_0_DBG1 **Reset value: 0x0000_0000**	**Register offset: 104C, 108C, 10CC, 110C, 114C, 118C, 11CC, 124C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	CAPT_0[0:7]							
8:15	CAPT_0[8:15]							
16:23	CAPT_0[16:23]							
24:31	CAPT_0[24:31]							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:31	CAPT_0	Character and control symbol or bytes 0 to 3 of packet header. Unmapped DestID is the only error for which information is latched in Port 8. This register is only valid when VAL_CAPT in "RapidIO Port x Error Capture Attributes CSR and Debug 0 Register" is 1. Contents vary with the error type.	R/W	0

25.6.13 RapidIO Port x Packet Error Capture CSR 1 and Debug 2 Register

In debug mode this register is unlocked. It contains bytes 8 to 11 of the debug packet being composed. During normal operation, this register captures bytes 4 to 7 of the packet that was detected to be in error.

To assist in software testing and debug of the system error recovery and threshold function, the "RapidIO Port x Error Detect CSR" and the Port x Error Capture registers (0x14C, 0x1050, 0x1054, and 0x1058) are also writable. Software must set the Capture Valid Info bit in the Port x Attributes Error Capture CSR, then write the packet/control symbol information to the other capture registers.

Register name: SP{0,1,2,3,4,5,6,8}_ERR_CAPT_1_DBG2 **Reset value: 0x0000_0000**	**Register offset: 1050, 1090, 10D0, 1110, 1150, 1190, 11D0, 1250**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	CAPT_1[0:7]							
8:15	CAPT_1[8:15]							
16:23	CAPT_1[16:23]							
24:31	CAPT_1[24:31]							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:31	CAPT_1	Bytes 4 to 7 of the packet This register is only valid when VAL_CAPT in "RapidIO Port x Error Capture Attributes CSR and Debug 0 Register" is 1. Contents vary with the error type.	R/W	0

25.6.14 RapidIO Port x Packet Error Capture CSR 2 and Debug 3 Register

In debug mode this register is unlocked. It contains bytes 12 to 15 of the debug packet being composed. During normal operation, this register captures bytes 8 to 11 of the packet that was detected to be in error.

To assist in software testing and debug of the system error recovery and threshold function, the "RapidIO Port x Error Detect CSR" and the Port x Error Capture registers (0x14C, 0x1050, 0x1054, and 0x1058) are also writable. Software must set the Capture Valid Info bit in the Port x Attributes Error Capture CSR, then write the packet/control symbol information to the other capture registers..

Register name: SP{0,1,2,3,4,5,6,8}_ERR_CAPT_2_DBG3 **Reset value: 0x0000_0000**	**Register offset: 1054, 1094, 10D4, 1114, 1154, 1194, 11D4, 1254**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0:7	CAPT_2[0:7]							
8:15	CAPT_2[8:15]							
16:23	CAPT_2[16:23]							
24:31	CAPT_2[24:31]							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:31	CAPT_2	Byte 8 to 11 of the packet. This register is only valid when VAL_CAPT in "RapidIO Port x Error Capture Attributes CSR and Debug 0 Register" is 1. Contents vary with the error type.	R/W	0

25.6.15 RapidIO Port x Packet Error Capture CSR 3 and Debug 4 Register

In debug mode this register is unlocked. It contains bytes 16 to 19 of the debug packet being composed. During normal operation, this register captures bytes 12 to 15 of the packet that was detected to be in error.

To assist in software testing and debug of the system error recovery and threshold function, the "RapidIO Port x Error Detect CSR" and the Port x Error Capture registers are also writable. Software must set the Capture Valid Info bit in the Port x Attributes Error Capture CSR, then write the packet/control symbol information to the other capture registers..

Register name: SP{0,1,2,3,4,5,6,8}_ERR_CAPT_3_DBG4 **Reset value: 0x0000_0000**	**Register offset: 1058, 1098, 10D8, 1118, 1158, 1198, 11D8, 1258**

Bits	0	1	2	3	4	5	6	7
0:7	CAPT_3[0:7]							
8:15	CAPT_3[8:15]							
16:23	CAPT_3[16:23]							
24:31	CAPT_3[24:31]							

Bits	Name	Description	Type	Reset Value
0:31	CAPT_3	Byte 12 to 15 of the packet. This register is only valid when VAL_CAPT in "RapidIO Port x Error Capture Attributes CSR and Debug 0 Register" is 1. Contents vary with the error type.	R/W	0

25.6.16 RapidIO Port x Error Rate CSR

This register and the "RapidIO Port x Error Rate Threshold CSR" monitor and control the reporting of transmission errors.

Register name: SP{0,1,2,3,4,5,6,8}_ERR_RATE Reset value: 0x8000_0000	Register offset: 1068, 10A8, 10E8, 1128, 1168, 11A8, 11E8, 1268

Bits	0	1	2	3	4	5	6	7
00:07	ERR_RB							
08:15	Reserved						ERR_RR	
16:23	PEAK							
24:31	ERR_RATE_CNT							

Bits	Name	Description	Type	Reset Value
0:7	ERR_RB	The Error Rate Bias value. Reference clock frequency - 156.25 MHz • 00 = Do not decrement error rate counter • 01 = Decrement every 1.68ms • 02 = Decrement every 13.41ms • 04 = Decrement every 107.37ms • 08 = Decrement every 1.72 s • 10 = Decrement every 13.75 s • 20 = Decrement every 110 s • 40 = Decrement every 17589s • 80 = Decrement every 14074 s • FF - Decrement every 1.6384 us (Debug only) Other values are reserved. Reference clock frequency - 125 MHz • 00 = Do not decrement error rate counter • 01 = Decrement every 2.1 ms • 02 = Decrement every 16.77 ms • 04 = Decrement every 134.21 ms • 08 = Decrement every 2.147 s • 10 = Decrement every 17.18 s • 20 = Decrement every 137.44 s • 40 = Decrement every 2198 s • 80 = Decrement every 17592 s • FF - Decrement every 2.048 us (Debug only) Other values are reserved.	R/W	0x80

(Continued)

Bits	Name	Description	Type	Reset Value
8:13	Reserved	N/A	R	0
14:15	ERR_RR	Error Rate Recovery This field defines how far above the Error Rate Failed Threshold Trigger the Error Rate Counter is allowed to count. • 00 = 2 errors above • 01 = 4 errors above • 10 = 16 errors above • 11 = No limit	R/W	0
16:23	PEAK	The maximum value attained by the error rate counter. This value increments with ERR_RATE_CNT, but does not decrement except through a host controlled register write.	R/W	0
24:31	ERR_ RATE_ CNT	Error Rate Counter These bits maintain a count of the number of transmission errors that are detected by the port. This number is decremented by the Error Rate Bias function. The counter cannot over or underflow and continue to increment or decrement as defined, even if thresholds are met. Software can reset this counter. If the value of the counter equals the error rate threshold trigger register, an error is reported. For more information see the *RapidIO Interconnect Specification (Revision 1.3), Part 8: Error Management Extensions Specification*. Note: Only the first TEA or MC_TEA error which changes the "RapidIO Port x Error and Status CSR".OUTPUT_DROP bit from 0 to 1 will be counted towards an Error Rate Degraded Threshold Reached or Error Rate Failed Threshold Reached event. Subsequent TEA or MC_TEA errors which occur while the OUTPUT_DROP bit is 1 do not cause the error rate counter in the "RapidIO Port x Error Rate CSR" to increment. Once the OUTPUT_DROP bit is cleared, the next TEA or MC_TEA error will cause the error rate counter to increment.	R/W	0

25.6.17 RapidIO Port x Error Rate Threshold CSR

This register and the "RapidIO Port x Error Rate CSR" monitor and control the reporting of transmission errors.

Register name: SP{0,1,2,3,4,5,6,8}_ERR_THRESH Reset value: 0xFFFF_0000	Register offset: 106C, 10AC, 10EC, 112C, 116C, 11AC, 11EC, 126C

Bits	0	1	2	3	4	5	6	7
00:07	ERR_RFT							
08:15	ERR_RDT							
16:23	Reserved							
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0:7	ERR_RFT	Error Rate Failed Threshold These bits provide the threshold value for reporting an error condition due to a possibly broken link. • 00 = Disable the error rate failed register • 01 = Set the error reporting threshold to 1 • 02 = Set the error reporting threshold to 2 • ... • FF - Set the error reporting threshold to 255	R/W	0xFF
8:15	ERR_RDT	Error Rate Degraded Threshold These bits provide the threshold value for reporting an error condition due to a degrading link. • 00 = Disable the error rate degraded register • 01 = Set the error reporting threshold to 1 • 02 = Set the error reporting threshold to 2 • ... • FF - Set the error reporting threshold to 255	R/W	0xFF
16:31	Reserved	N/A	R	0

25.7 IDT-Specific RapidIO Registers

Table 155 shows IDT-specific RapidIO registers that are not defined in the *RapidIO Interconnect Specification (Revision 1.3)*.

When a individual port is powered down, the IDT-Specific RapidIO Registers are read only and return 0 with the exception of the "RapidIO Port x Error and Status CSR" and "RapidIO Serial Port x Control CSR", both of which return 0x00000001 when read.

These registers are reset by several sources of Tsi620 reset (see "Resets"). The registers within a port are also reset by a port reset, performed by powering up and down the port. For more information on Tsi620 reset design, see "Clock, Reset, Power-up, and Initialization Options". It is possible to override reset values of writable fields, and some read-only fields, using the I^2C register loading capability on boot. For more information on the use of I^2C Interface register loading capability, see "I2C Interface"

Table 155: IDT-Specific "Broadcast" RapidIO Registers

Port	Register Offset	Description
BC	10000	Broadcast addresses, which affect register copies in all the ports.
SP0	11000	1x/4x Serial port
SP1	11100	1x Serial port
SP2	11200	1x/4x Serial port
SP3	11300	1x Serial port
SP4	11400	1x/4x Serial port
SP5	11500	1x Serial port
SP6	11600	1x/4x Serial port
SP7	11700	1x Serial port
SP8	11800	Internal Switch Port

Non-Broadcast Per-Port Registers

Table 156 shows the IDT-specific per-port registers not defined by the *RapidIO Interconnect Specification (Revision 1.3)*. It is not possible to broadcast to these registers.

Table 156: IDT-Specific Per-Port Performance Registers

Port	Register Offset	Description
SP0	13000	1x/4x Serial port
SP1	13100	1x Serial port

Table 156: IDT-Specific Per-Port Performance Registers (Continued)

Port	Register Offset	Description
SP2	13200	1x/4x Serial port
SP3	13300	1x Serial port
SP4	13400	1x/4x Serial port
SP5	13500	1x Serial port
SP6	13600	1x/4x Serial port
SP7	13700	1x Serial port
SP8	13800	1x/4x Serial port

25.7.1 RapidIO Port x Discovery Timer Register

Where *x* refers to a broadcast register (BC) and ports 0-8. This register defines discovery-timer value for the RapidIO ports in 4x mode:

- Offset 10000: (BC) Broadcast to all RapidIO ports
- Offset 11000–11800: Port 0-6,8 specific

Note that this register has no functionality for Port 8 in the Tsi620.

Register name: SP{BC,0,1,2,3,4,5,6,8}_DISCOVERY_TIMER **Reset value: 0x90C0_0000**	**Register offset: 10000, 11000, 11100, 11200, 11300, 11400, 11500, 11600, 11800**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	DISCOVERY_TIMER				Reserved			
08:15	PW_PRIORITY		Reserved					
16:23	Reserved							
24:31	Reserved							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:3	DISCOVERY_ TIMER	Discovery Timer This field is used by RapidIO ports configured to operate in 4x mode. The discovery-timer allows time for the link partner to enter its discovery state, and if the link partner supports 4x mode, for all four lanes to be aligned. The period of the discovery timer is described by the following relationship: • 0: 32 cycles of sys_clk (debug only) {1:15}: DISCOVERY_TIMER*clk_period < actual time < (DISCOVERY_TIMER+1)*clk_period When the reference clock frequency is 156.25 MHz, the clk_period value is 1.677 msec, so the default period is 15.09 msec and the maximum period is 25.15 msec. When the reference clock frequency is 125 MHz, the clk_period value is 2.096 msec, so the default period is 18.86 msec and the maximum period is 31.44 msec. Note: This field controls no functionality for port 8.	R/W	9
4:7	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
8:9	PW_PRIORITY	Port-Write packet priority This field sets the priority of a Port-Write packet. The priority can be set from 0 to 3, where 00 = priority 0 01 = priority 1 10 = priority 2 11 = priority 3	R/W	11
10:31	Reserved	N/A	R	0

25.7.2 RapidIO Port x Mode CSR

Where *x* refers to a broadcast register (BC) and ports 0-8. This register defines the mode of operation for the ports, and contains the interrupt enables for the Multicast-Event control symbol and Reset control symbol.

- Offset 10004: (BC) Broadcast to all ports
- Offset 11004–11804: Port 0-8 specific

Register name: SP{BC,0,1,2,3,4,5,6,8}_MODE **Reset value: 0x0300_0000**	**Register offset: 10004, 11004, 11104, 11204, 11304, 11404, 11504, 11604, 11804**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved		IDLE_ERR_DIS	Reserved	PW_DIS	Reserved	SELF_RST	LUT_512
08:15	Reserved							
16:23	Reserved							
24:31	Reserved						MCS_INT_EN	RCS_INT_EN

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:1	Reserved	Reserved	R	0
2	IDLE_ERR_DIS	Idle Error Checking Disable 0 = Error checking enabled (default), only |K|, |A| and |R| characters are available. If any other characters are received in the idle sequence, enter the Input-Error-stopped state. 1 = Ignore all not idle or invalid characters in the idle sequence. Note that this bit still applies to the operation of the FPGA Interface. Further note that it is not possible to determine if a link partner is present on the FPGA Interface if invalid characters are not checked for.	R/W	0
3	Reserved	Reserved	R	0
4	PW_DIS	Port_Write Disable 0 = Port-Write Error reporting is enabled (default) 1 = Port-Write is disabled	R/W	0
5	Reserved	Reserved	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
6	SELF_RST	Self Reset Enable After four link-request reset control symbols are accepted, the device either resets itself or raises an interrupt, according to the value in this register field. 0 = Disable: RST_IRQ_b signal is asserted (if RCS_INT_EN in this register is also asserted) 1 = Enable: Device is reset	R/W	1
7	LUT_512	LUT_512 Sets the mode of the destination ID lookup table 0 = Global LUT (64-KB destIDs, assigned with resolution of 256 destIDs) 1 = One 512-entry local LUT	R/W	1
8:29	Reserved	Reserved	R	0
30	MCS_INT_EN	Multicast-Event Control Symbol Interrupt Enable 0 = Disable 1 = Enable If enabled, the interrupt signal is high when the multicast-event control symbol is received.	R/W	0
31	RCS_INT_EN	Reset Control Symbol Interrupt Enable 0 = Disable 1 = Enable If enabled, the interrupt signal is High when the four reset control symbols are received in a sequence.	R/W	0

25.7.3 RapidIO Port x Multicast-Event Control Symbol and Reset Control Symbol Interrupt CSR

This register contains the interrupt status for Multicast-Event control symbols and Reset control symbols, where *x* refers to a broadcast register (BC) and ports 0 through 8.

- Offset 10008: (BC) Broadcast to all ports
- Offset 11008-11808: Port 0-8 specific

Register name: SP{BC,0,1,2,3,4,5,6,8}_CS_INT_STATUS **Reset value: 0x0000_0000**	**Register offset: 10008, 11008, 11108, 11208, 11308, 11408, 11508, 11608, 11808**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							
24:31	Reserved						MCS	RCS

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:29	Reserved	N/A	R	0
30	MCS	Multicast Event Control Symbol Interrupt Status Indicates whether a multicast event control symbol is received on the port. Reading SPBC_CS_INT_STATUS.MCS gives the value of Port 0. All MCS interrupts from ports are ORed together. The GLOB_INT_STAT register shows the status of the combined MCS interrupts from all ports. Write 1 to clear this bit. Writing 1 to this bit in the SPBC_CS_INT_STATUS register clears the interrupt on all the ports.	R/W1C	0
31	RCS	Reset Control Symbol Received Interrupt Status Indicates that four consecutive Reset control symbols are received on the port. Reading SPBC_CS_INT_STATUS.RCS gives the value of Port 0. All RST interrupts from ports are ORed together. The "Switch Interrupt Status Register" shows the status of the combined RCS from all ports. Write 1 to clear this bit. Writing 1 to this bit in the SPBC_CS_INT_STATUS register clears the interrupt on all the ports.	R/W1C	0

25.7.4 RapidIO Port x RapidIO Watermarks Register

Where *x* refers to a broadcast register (BC) and ports 0 through 8.

- Offset 1000C: (BC) Broadcast to all ports
- Offset 1100C-1180C: Port 0-8 specific

This register controls (ingress) buffer allocation for reception of packets for each ingress port (see "Egress Watermark").

Register name: SP{BC,0,1,2,3,4,5,6,8}_RIO_WM **Reset value: 0x0001_0203**	**Register offset: 1000C, 1100C, 1110C, 1120C, 1130C, 1140C, 1150C, 1160C, 1180C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	Reserved							
8:15	Reserved					PRIO2WM		
16:23	Reserved					PRIO1WM		
24:31	Reserved					PRIO0WM		

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:12	Reserved	N/A	R	0
13:15	PRIO2WM	Priority 2 packets are accepted if the number of free buffer is greater than this value. This value must be smaller than PRIO1WM. Note: It is a programming error for this value to be greater than or equal to PRIO1WM or PRIO0WM or greater than 7. For other restrictions on the value of this register field, see "I2R Watermark and Buffer Release Management Register Value Restrictions".	R/W	1
16:20	Reserved	N/A	R	0
21:23	PRIO1WM	Priority 1packets are accepted if the number of free buffer is greater than this value. This value must be smaller than PRIO0WM. Note: It is a programming error for this value to be greater than or equal to PRIO0WM or greater than 7.	R/W	2
24:28	Reserved	N/A	R	0
29:31	PRIO0WM	Priority 0 packets are accepted if the number of free buffer is greater than this value. Note: It is a programming error for this value to be greater than 7. For other restrictions on the value of this register field, see "I2R Watermark and Buffer Release Management Register Value Restrictions".	R/W	3

Do not program this register in traffic; only program these registers after reset.

25.7.5 RapidIO Port x Route Configuration DestID CSR

Where *x* refers to a broadcast register (BC) and ports 0 through 8. This register and "RapidIO Port x Route Configuration Output Port CSR" operate together to provide indirect read and write access to the LUTs.

They are identical to "RapidIO Route Configuration DestID CSR" and "RapidIO Route Configuration Output Port CSR", except that "RapidIO Port x Route Configuration Output Port CSR" are per-port registers and they include an auto-increment bit to increment the contents of the DESTID register after a read or write operation.

Register name: SP{BC,0,1,2,3,4,5,6,8}_ROUTE_CFG_DESTID **Reset value: 0x0000_0000**	**Register offset: 10070, 11070, 11170, 11270, 11370, 11470, 11570, 11670, 11870**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	AUTO_INC	PAR_INVERT	Reserved					
08:15	Reserved							
16:23	LRG_CFG_DEST_ID[0:7]							
24:31	CFG_DEST_ID[8:15]							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	AUTO_INC	Automatically post-increment the DestID when the DestID performs either a read or a write through the SPx_ROUTE_CFG_PORT register.	R/W	0
1	PAR_INVERT	Parity Invert This bit is for testing of interrupt and/or demerit software systems. 0 = Normal operation 1 = Invert the parity bit for each LUT entry written (but not read). This causes a parity error when the LUT entry routes a packet.	R/W	0
2:15	Reserved	N/A	R	0
16:23	LRG_CFG_DEST_ID	This field specifies the most significant byte of the destination ID used to select an entry in the LUT, when the "RapidIO Route Configuration Output Port CSR" register is read or written.	R/W	0x00
24:31	CFG_DEST_ID	Specifies the destination ID used to select an entry in the LUT when the "RapidIO Route Configuration Output Port CSR" register is read or written. This value increments by one for every write to the "RapidIO Port x Route Configuration Output Port CSR" when the AUTO_INC bit is set.	R/W	0x00

25.7.6 RapidIO Port x Route Configuration Output Port CSR

Where *x* refers to a broadcast register (BC) and ports 0 through 8. This register and "RapidIO Port x Route Configuration DestID CSR" operate together to provide indirect read and write access to the LUTs.

They are identical to "RapidIO Route Configuration DestID CSR" and "RapidIO Route Configuration Output Port CSR", except that these register are per-port and they include an auto-increment bit to increment the contents of the DESTID register after a read or write operation.

Register name: SP{BC,0,1,2,3,4,5,6,8}_ROUTE_CFG_PORT **Reset value: Undefined**	**Register offset: 10074, 11074, 11174, 11274, 11374, 11474, 11574, 11674, 11874**

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							
24:31	PORT							

Bits	Name	Description	Type	Reset Value
0:23	Reserved	N/A	R	0
24:31	PORT	This is the RapidIO output port through which all messages for CONFIG_DESTID are sent. Writing a value greater or equal to PORT_TOTAL sets the LUT entry to an unmapped state. For compatibility with future IDT devices, write the value 0xFF to indicate an unmapped DestID. When reading an unmapped value from the LUT, this field is set to 0xFF.	R/W	Undefined

25.7.7 RapidIO Port x Local Routing LUT Base CSR

This register is required for RapidIO switches that operate in a large system. For small systems, this register is ignored. The RapidIO port supports local and remote routing LUT pages. The number of entries is defined by the "RapidIO Route LUT Size CAR".

Register name: SP{BC,0,1,2,3,4,5,6,8}_ROUTE_BASE **Reset value: 0x0000_0000**	**Register offset: 10078, 11078, 11178, 11278, 11378, 11478, 11578, 11678, 11878**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	BASE							
08:15	Reserved							
16:23	Reserved							
24:31	Reserved							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:7	BASE	This value represents the most significant byte of a destination ID. If the most significant upper 8 bits of an incoming 16-bit destination ID match this field, the least significant bits of the destination ID is used to index the local LUT. Otherwise, the most significant 8 bits of the destination ID is used to index the global LUT (see "Lookup Tables").	R/W	0
8:31	Reserved	N/A	R	0

25.7.8 RapidIO Multicast Write ID x Register

Where *x* refers to eight multicast IDs (0–7). This register contains the Multicast ID for which the associated multicast mask registers are applicable. The Tsi620 Switch supports eight multicast groups, therefore, the Multicast ID registers for each multicast group must contain unique values. These registers are used in every RapidIO port.

Note that there are per port copies of these registers, described in "RapidIO Serial Port x Multicast Write ID 0 Register" through "RapidIO Serial Port x Multicast Write ID 7 Register".

Register name: RIO_MC_ID{0..7} **Reset value: 0x0000_0000**	**Register offset: 10300, 10304, 10308, 1030C, 10310, 10314, 10318, 1031C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	MC_EN	LARGE_SYS	Reserved					
8:15	Reserved							
16:23	MC_ID[0:7]							
24:31	MC_ID[8:15]							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	MC_EN	Multicast can be disabled by setting this bit. 0 = Multicast disabled 1 = Multicast enabled	R/W	0
1	LARGE_SYS	This field defines multicast destination ID (MC_ID) in the Large or Small system. The MC_ID of Small system is not a subset of MC_ID of Large system, but both systems can co-exist together. 1 = Large system 0 = Small system	R/W	0
2:15	Reserved	N/A	R	0
16:31	MC_ID	This field defines the multicast destination ID for which the associated multicast mask is activated for this extended features block.	R/W	0x0000

When using these registers, it is important that no multiple identical entries exist since an addition of an association will not delete the existing one.

25.7.9 RapidIO Multicast Write Mask x Register

Where *x* refers to the eight multicast masks, 0–6 and 8. This register contains the set of egress ports (the "mask") to which a multicast packet is sent when it matches the DestID associated with the mask.

The bit descriptions apply to all packets received on a port whose DestID field maps to the multicast ID register value. A multicast packet received on an input port is sent to all exit ports whose multicast select bit is 1, except the port from which it was received, regardless of the setting of that port's multicast select bit.

This register is located in the Multicast Engine only.

Register name: RIO_MC_MSK{0..7} **Reset value: 0x0000_0000**	**Register offset: 10320, 10324, 10328, 1032C, 10330, 10334, 10338, 1033C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	MC_MSK[0:7]							
08:15	MC_MSK[8:15]							
16:23	Reserved							
24:31	Reserved							

Bits	**Name**	**Description**	**Type**	**Reset Value**
00:15	MC_MSK	Port x multicast select Where x refers to ports 0–6 and 8 (Other values are reserved.) • 0: Do not Multicast the packet to output port x • 1: Multicast the packet to output port x An output port is specified by the bit position: bit 0 = mask of output port 0 bit 1 = mask of output port 1 and so on.	R/W	0
16:31	Reserved	N/A	R	0

25.7.10 RapidIO Port x Control Independent Register

This register is used for error recovery.

Register name: SP{0,1,2,3,4,5,6,8}_CTL_INDEP Reset value: 0x0100_0000	Register offset: 13004, 13104, 13204, 13304, 13404, 13504, 13604, 13804

Bits	0	1	2	3	4	5	6	7
00:07	Reserved		SCRATCH	Reserved		FORCE_REINIT	Reserved	TRANS_MODE
08:15	DEBUG_MODE	SEND_DBG_PKT	Reserved		PORT_ERR_EN	MC_TEA_EN	LINK_INIT_NOTIFICATION_EN	LUT_PAR_ERR_EN
16:23	MAX_RETRY_THRESHOLD							
24:31	ILL_TRANS_ERR	IRQ_EN	MAX_RETRY_EN	OUTB_DEPTH_EN	INB_DEPTH_EN	INB_RDR_EN	Reserved	TEA_EN

Bits	Name	Description	Type	Reset Value
0:1	Reserved	N/A	R	0
2	SCRATCH	This bit controls no functionality. It is a read/write ‘scratch pad’ bit for software use.	R/W	0
3:4	Reserved	N/A	R	0
5	FORCE_REINIT	Force link re-initialization process. This bit is active on write and automatically returns to 0. For FPGA Interface, this bit causes the port to enter the SILENT state for link initialization, asserting the PHY_DISABLE pin. PHY_DISABLE should let the link partner know that the FPGA Interface was just reset. Note that the transmit clock signal and data/control signals continue to be driven. This bit controls no functionality for Port 8 (Internal Switch Port).	R/W1S	0
6	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
7	TRANS_MODE	Describes the transfer mode for each port. 0 = Cut-through mode 1 = Store-and-forward mode In cut-through mode, the incoming packet is forwarded through the Tsi620 Switch as soon as the routing information is received. In store-and-forward mode, the incoming packet is not sent to the Switch ISF until the whole packet is received. Note: Cut-through mode can have detrimental effect on overall switch performance if ports are operating at different speeds. This is because in cut-through mode, a slower port can use the Switch ISF for a long time relative to a faster port, incurring additional latency and possible throughput loss on the faster port.	R/W	1
8	DEBUG_ MODE	Mode of operation 0 = Normal 1 = Debug mode Debug mode unlocks the capture registers for writing and enables the debug packet generator feature. This bit controls no functionality for Port 8 (Internal Switch Port).	R/W	0
9	SEND_DBG_ PKT	Send Debug Packet 1 = Send debug packet This bit is set by software and is cleared by hardware after the debug packet is sent. Writes when the bit is already set are ignored. Debug mode only. This bit controls no functionality for Port 8 (Internal Switch Port).	R/W	0
10:11	Reserved	N/A	R	0
12	PORT_ERR_EN	Port Error Enable An interrupt is generated if there is a Port Error. This bit controls no functionality for Port 8 (Internal Switch Port).	R/W	0
13	MC_TEA_EN	Multicast TEA Enable An interrupt is generated if the Switch ISF times out trying to send the packet to its egress port.	R/W	0
14	LINK_INIT_ NOTIFICATION_ EN	Enables interrupts and port writes for LINK_INIT_NOTIFICATION events. 0 = Interrupt and port write disabled. 1 = Interrupt and port write enabled. Note: This bit still operates on the FPGA Interface with no change.	R/W	0
15	LUT_PAR_ERR_ EN	Enables interrupts for parity errors in the lookup table. 0 = interrupt disabled 1 = interrupt enabled	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
16:23	MAX_RETRY_THRESHOLD	Maximum Retry Threshold These bits provide the threshold value for reporting congestion at an outbound switch buffer caused by congestion at the link partner. When the number of consecutive retries reaches this threshold, the Tsi620 Switch generates a port-write and sends the IMP_SPEC_ERR bit in the "RapidIO Port x Error Detect CSR". 00 = Disable the RETRY_ERROR reporting 01 = Set the MAX_RETRY_THRESHOLD to 1 02 = Set the MAX_RETRY_THRESHOLD to 2 ... FF - Set the MAX_RETRY_THRESHOLD to 255 This bit controls no functionality for Port 8 (Internal Switch Port).	R/W	0x00
24	ILL_TRANS_ERR	Illegal Transfer Error Reporting Enable If enabled, the port-write and interrupt report an error when the ILL_TRANS_ERR bit is set.	R/W	0
25	IRQ_EN	Interrupt Error Report Enable If enabled, the interrupt signal is high when the IRQ_ERR bit is set to 1.	R/W	0
26	MAX_RETRY_EN	Maximum Retry Report Enable If enabled, the port-write and interrupt report an error when the MAX_RETRY_THRESHOLD is exceeded and the MAX_RETRY bit is set in the SPx_INT_STATUS register. This bit controls no functionality for Port 8 (Internal Switch Port).	R/W	0
27	OUTB_DEPTH_EN	Output Queue Depth Interrupt Enable An interrupt is generated when the OUTB_DEPTH bit is set in the SPx_INT_STATUS register.	R/W	0
28	INB_DEPTH_EN	Input Queue Depth Interrupt Enable An interrupt is generated when the INB_DEPTH bit is set in the SPx_INT_STATUS register.	R/W	0
29	INB_RDR_EN	Inbound Reorder Interrupt Enable An interrupt is generated when the INB_RDR_IRQ bit is set in the SPx_INT_STATUS register.	R/W	0
30	Reserved	Reserved	R	0
31	TEA_EN	Transfer Error Acknowledge Enable An interrupt is generated if the Switch ISF times out trying to send the packet to its egress port.	R/W	0

25.7.11 RapidIO Port x Send Multicast-Event Control Symbol Register

Where x refers to ports 0 through 8. When this register is written, it causes a Multicast-Event control symbol to send on the corresponding RapidIO output port. The port must be enabled for multicast control symbol forwarding using the SP{0..8}_CTL.MCS_EN register bit.

A write to this register is not considered complete until the multicast-event control symbol is queued to the outbound flow. There can be only one outstanding request at a time. Subsequent requests are ignored until the multicast control symbol is sent.

Register name: SP{0,1,2,3,4,5,6,8}_SEND_MCS **Reset value: 0x0000_0002**	**Register offset: 1300C, 1310C, 1320C, 1330C, 1340C, 1350C, 1360C, 1380C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							
24:31	Reserved						DONE	SEND

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:29	Reserved	N/A	R	0
30	DONE	The Tsi620 sets this field to 0 when system software sets SEND to 1. The Tsi620 sets this field to 1 once it has sent the Multicast-Event control symbol. Port 9's behavior is contingent upon the links ability to send control symbols (see "Multicast-event Control Symbols"). A value of 1 in this field indicates that the Tsi620 is ready to send another Multicast-Event control symbol.	R	1
31	SEND	Write 1 to send a multicast-event control symbol when DONE = 1.	R/W1S	0

25.7.12 RapidIO Port x LUT Parity Error Info CSR

Where x refers to ports 0 through 8. There is no broadcast address for this register. This register contains information about the lookup operation that cause the parity error, as well as the LUT information associated with the parity error.

This register's contents are frozen when a LUT parity error is indicated in "RapidIO Port x Interrupt Status Register". Writes to this register have no effect when a LUT parity error is not indicated.

Register name: SP{0,1,2,3,4,5,6,8}_LUT_PAR_ERR_INFO **Reset value: 0x0000_0000**	**Register offset: 13010, 13110, 13210, 13310, 13410, 13510, 13610, 13810**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	DESTID_MSB							
08:15	DESTID_LSB							
16:23	LG_DESTID	Reserved						
24:31	PTY_BIT	LUT_VLD	Reserved		PORT_NUM			

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:7	DESTID_MSB	Most significant byte of a 16-bit destination ID used in the lookup operation that caused the error. Only valid if the LG_DESTID field value is 1 and a LUT parity error is signaled in the "RapidIO Port x Interrupt Status Register". When the parity error is cleared in "RapidIO Port x Interrupt Status Register", the information in these bits become meaningless.	R/W	0
8:15	DESTID_LSB	Least significant byte of a 16-bit destination ID used in the lookup operation that caused the error. Only valid if a LUT parity error is signalled in the "RapidIO Port x Interrupt Status Register". When the parity error is cleared in "RapidIO Port x Interrupt Status Register", the information in these bits become meaningless.	R/W	0
16	LG_DESTID	This field is 1 if the TT code of the packet that caused the error is anything other than 0.	R/W	0
17:23	Reserved	N/A	R	0
24	PTY_BIT	The parity bit read from the LUT memory array.	R/W	0
25	LUT_VLD	The value of this field is undefined in the event of a LUT parity error.	R/W	0
26:27	Reserved	N/A	R	0
28:31	PORT_NUM	The Tsi620 port number to which packets are routed.	R/W	0

25.7.13 RapidIO Port x Control Symbol Transmit Register

Where x refers to ports 0 through 6. Writing to this register transmits a single control symbol to RapidIO. This register is only used for debug purposes.

Note that this register does not exist for the Internal Switch Port (Port 8).

All control symbol fields are defined according to the *RapidIO Interconnect Specification (Revision 1.3)*. The control symbol's CRC field is generated by hardware.

Register name: SP{0,1,2,3,4,5,6}_CS_TX **Reset value: 0x0000_0000**	**Register offset: 13014, 13114, 13214, 13314, 13414, 13514, 13614**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	STYPE_0			PAR_0				
08:15	PAR_1					STYPE_1		
16:23	CMD			CS_EMB	Reserved			
24:31	Reserved							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:2	STYPE_0	Encoding for control symbol - uses parameters PAR_0 and PAR_1.	R/W	0
3:7	PAR_0	Used in conjunction with stype0 encoding.	R/W	0
8:12	PAR_1	Used in conjunction with stype0 encoding.	R/W	0
13:15	STYPE_1	Encoding for the control symbol that uses the CMD parameter.	R/W	0
16:18	CMD	Used in conjunction with stype1 encoding to define the link maintenance commands.	R/W	0
19	CS_EMB	Embed the control symbol into a data stream. 0 = control symbol is sent out immediately 1 = control symbol is sent immediately if there is data transferring on the output port, or is inserted until after the first 32 bits of data of the next packet sent if there is currently no data transferring on the output port.	R/W	0
20:31	Reserved	N/A	R	0

Writing to this register causes control symbols to be generated that may interfere with the operation of the port. This can cause the port or its link partner to enter the input error stopped state due to reception of an unexpected control symbol.

25.7.14 RapidIO Port x Interrupt Status Register

Where *x* refers to ports 0 through 8. Note that writing 0 to any bit in the "RapidIO Port x Interrupt Generate Register" will clear the associated bit in this register.

Register name: SP{0,1,2,3,4,5,6,8}_INT_STATUS **Reset value: 0x0000_0000**	**Register offset: 13018, 13118, 13218, 13318, 13418, 13518, 13618, 13818**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved					MC_TEA	LINK_INIT_NOTIFICATION	LUT_PAR_ERR
16:23	Reserved							
24:31	ILL_TRANS_ERR	IRQ_ERR	MAX_RETRY	OUTB_DEPTH	INB_DEPTH	INB_RDR	Reserved	TEA

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:12	Reserved	N/A	R	0
13	MC_TEA	This interrupt is raised when the Multicast Engine has timed out before it could transfer a packet to the broadcast buffer. This bit is cleared by writing a 1 to it, or by clearing all bits in the "RapidIO Port x Error Detect CSR".	R/W1C	0
14	LINK_INIT_NOTIFICATION	Link Initialization Notification Once set, the LINK_INIT_NOTIFICATION bit is cleared by writing 1 to it. When the PORT_LOCKOUT bit is set in "RapidIO Serial Port x Control CSR", and a link has initialized according to the PORT_OK bit in "RapidIO Port x Error and Status CSR", the LINK_INIT_NOTIFICATION is set to 1. To stop the LINK_INIT_NOTIFICATION bit from getting set, PORT_LOCKOUT must be set to 0 and/or the link must no longer be in an initialized state. Note: This bit operates in an unchanged manner for the FPGA Interface.	R/W1C	0
15	LUT_PAR_ERR	Lookup Table Parity Error Set to one when a packet looks up its destination ID in the lookup table, and the selected lookup table entry has a parity error. This bit is cleared by writing a 1 to it, or by clearing all the bits in the "RapidIO Port x Error Detect CSR".	R/W1C	0
16:23	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
24	ILL_TRANS_ERR	Illegal Transfer Error It is set to 1 as follows: • Received transaction has reserved TT field for all but maintenance packets with hop count = 0 • DestID is unmapped (not defined in lookup table) Once set, remains set until written with logic 1 to clear. The setting of this bit generates a Port-Write and interrupt if a "RapidIO Logical and Transport Layer Error Enable CSR"/ bit ILL_TRANS_EN is set. This bit duplicates a function of "RapidIO Port x Error Detect CSR" / bit 0, but Port-Write is sent immediately when the error is detected without exceeding threshold "RapidIO Port x Error Rate Threshold CSR". This bit is cleared by writing 1. This error also reported in registers: "RapidIO Port x Error Detect CSR"/bit0.	R/W1C	0
25	IRQ_ERR	Interrupt Error Status Set to one if an error occurs (see the "Error Management"). Once set, the bit remains unchanged until all the error sources are cleared. Setting this bit generates an interrupt if the IRQ_EN bit in "RapidIO Port x Control Independent Register" is set.	R	0
26	MAX_RETRY	Maximum Retry Error Set when number of retries has reached MAX_RETRY_THRESHOLD. An interrupt is generated if MAX_RETRY_EN is set. Port-Write request can also be generated if enabled. This bit is ignored if MAX_RETRY_THRESHOLD is 0x00. This bit is cleared by writing a 1 to it, or by clearing all bits in the "RapidIO Port x Error Detect CSR". This bit will never be set for Port 8 (Internal Switch Port).	R/W1C	0
27	OUTB_DEPTH	Outbound Depth Interrupt This value is set when Output Queue Depth Count reaches the maximum number defined in the Output Queue Depth Threshold field in the Register (see "RapidIO Port x Transmitter Output Queue Depth Threshold Register"). To get an interrupt in this status register, the Outbound Depth Interrupt Enable bit in the Register (see "RapidIO Port x Control Independent Register") is set to 1. Writing a 1 to this bit clears the interrupt and clears CONG_CNTR in the Register (see"RapidIO Port x Transmitter Output Queue Congestion Status Register"). This bit will never be set for Port 8 (Internal Switch Port).	R/W1C	0

(Continued)

Bits	Name	Description	Type	Reset Value
28	INB_DEPTH	Inbound Depth Interrupt This value is set when Input Queue Depth Count reaches the maximum number defined in the Input Queue Depth Threshold field in the Register (see "RapidIO Port x Receiver Input Queue Depth Threshold Register"). To get an interrupt in this status register, the Inbound Depth Interrupt Enable bit in the Register (see "RapidIO Port x Control Independent Register") is set to 1. Writing a 1 to this bit clears the interrupt and clears INB_QD_CNT in the Register (see "RapidIO Port x Receiver Input Queue Congestion Status Register").	R/W1C	0
29	INB_RDR	Inbound Interrupt: Reordering This value is set when Reordering Count reaches the maximum number defined in the Inbound Reordering Threshold field in the Inbound Reordering Register (see "RapidIO Port x Reordering Counter Register"). To get an interrupt in this status register, the Inbound Interrupt Reordering Enable bit in the Register (see"RapidIO Port x Control Independent Register") must be set to 1. Writing a 1 to this bit clears the interrupt and clears INB_RDR_CNT in the Inbound Reordering Register (see "RapidIO Port x Reordering Counter Register").	R/W1C	0
30	Reserved	Reserved	R	0
31	TEA	This interrupt is raised when the Switch ISF has timed out before it could transfer a packet to an egress port. This bit is cleared by writing a 1 to it, or by clearing all bits in the "RapidIO Port x Error Detect CSR".	R/W1C	0

25.7.15 RapidIO Port x Interrupt Generate Register

Where *x* refers to ports 0 through 8. This register can generate the corresponding error in the "RapidIO Port x Interrupt Status Register". Behaviors associated with the error (port writes, interrupts) occur.

Note that writing 0 to any bit in this register the will clear the associated bit in "RapidIO Port x Interrupt Status Register".

Register name: SP{0,1,2,3,4,5,6,8}_INT_GEN **Reset value: 0x0000_0000**	**Register offset: 1301C, 1311C, 1321C, 1331C, 1341C, 1351C, 1361C, 1381C**

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved					MC_TEA_GEN	LINK_INIT_NOTIFICATION_GEN	LUT_PAR_ERR_GEN
16:23	Reserved							
24:31	ILL_TRANS_GEN	Reserved	MAX_RETRY_GEN	OUTB_DEPTH_GEN	INB_DEPTH_GEN	INB_RDR_GEN	Reserved	TEA_GEN

Bits	Name	Description[a]	Type	Reset Value
0:12	Reserved	N/A	R	0
13	MC_TEA_GEN	Force the MC_TEA bit to be set. Bit always reads as zero.	R/W1S	0
14	LINK_INIT_NOTIFICATION_GEN	Force the LINK_INIT_NOTIFICATION bit to be set. Bit always reads as zero.	R/W1S	0
15	LUT_PAR_ERR_GEN	Force the LUT_PAR_ERR bit to be set. Bit always reads as zero.	R/W1S	0
16:23	Reserved	N/A	R	0
24	ILL_TRANS_GEN	Force the ILL_TRANS bit to be set. Bit always reads as zero.	R/W1S	0
25	Reserved	N/A	R	0
26	MAX_RETRY_GEN	Force the MAX_RETRY bit to be set to 1. Bit always reads as zero.	R/W1S	0
27	OUTB_DEPTH_GEN	Force the OUTB_DEPTH bit to be set to 1. Bit always reads as zero.	R/W1S	0
28	INB_DEPTH_GEN	Force the INB_DEPTH bit to be set to 1. Bit always reads as zero.	R/W1S	0

(Continued)

Bits	Name	Description[a]	Type	Reset Value
29	INB_RDR_GEN	Force the INB_RDR bit to be set to 1. Bit always reads as zero.	R/W1S	0
30	Reserved	Reserved	R	0
31	TEA_GEN	Force the TEA bit to be set to 1. Bit always reads as zero.	R/W1S	0

a. All bits in this register enable bits in the "RapidIO Port x Interrupt Status Register".

25.8 IDT-Specific Performance Registers

Table 155 shows IDT-specific Performance Registers that are not defined in the *RapidIO Interconnect Specification (Revision 1.3)*.

25.8.1 RapidIO Port x Performance Statistics Counter 0 and 1 Control Register

This register controls the performance statistics counters PS0 and PS1 registers. For every performance statistics register SPx_PSCy (where y refers to the Performance Statistics counter PS0 and PS1), the following configurations are selected through this register: direction, type, and priority.

- The PSy_DIR field determines the performance statistics receiver versus transmitter direction application.
- The PSy_TYPE field assigns the type of statistics collection (for example, packet, control symbol, and multicast) to be accumulated for a given SPx_PSCy.
- The PSy_PRIO[0..3] fields determine the priority of the packets for performance statistics collection through the "RapidIO Port x Performance Statistics Counter 0 Register" and "RapidIO Port x Performance Statistics Counter 1 Register"). The SPx_PSCy can be disabled by selecting PSy_PRIO[0..3] to be set to 0. Setting PSy_PRIO[0..3] to all ones, allows for collecting performance statistics through the SPx_PSCy for all priority packets.

Register name: SP{0,1,2,3,4,5,6,8}_PSC0n1_CTRL **Reset value: 0x0000_0000**	**Register offset: 13020, 13120, 13220, 13320, 13420, 13520, 13620, 13820**

Bits	0	1	2	3	4	5	6	7
00:7	PS0_PRIO 3	PS0_PRIO 2	PS0_PRIO 1	PS0_PRIO 0	Reserved			PS0_DIR
8:15	Reserved					PS0_TYPE		
16:23	PS1_PRIO 3	PS1_PRIO 2	PS1_PRIO 1	PS1_PRIO 0	Reserved			PS1_DIR
24:31	Reserved					PS1_TYPE		

Bits	Name	Description	Type	Reset Value
0	PS0_PRIO3	Performance Stats Reg PS0 Priority 3 Selection This value represents the packet priority 3 is selected for which performance statistics are accumulated for in the SPx_PS0_CTR counter register ["RapidIO Port x Performance Statistics Counter 0 Register"]. 0 = If all PS0_PRIO[0..3] are set to zero, the SPx_PS0_CTR counter register ["RapidIO Port x Performance Statistics Counter 0 Register"] is disabled. 1 = Count priority 3 packets.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
1	PS0_PRIO2	Performance Stats Reg PS0 Priority 2 Selection This value represents the packet priority 2 is selected for which performance stats are accumulated for in the SPx_PS0_CTR counter register [“RapidIO Port x Performance Statistics Counter 0 Register”]. 0 = If all PS0_PRIO[0..3] are set to zero, the SPx_PS0_CTR counter register [“RapidIO Port x Performance Statistics Counter 0 Register”]is disabled. 1 = Count priority 2 packets.	R/W	0
2	PS0_PRIO1	Performance Stats Reg PS0 Priority 1 Selection This value represents the packet priority 1 is selected for which performance stats are accumulated for in the SPx_PS0_CTR counter register [“RapidIO Port x Performance Statistics Counter 0 Register”]. 0 = If all PS0_PRIO[0..3] are set to zero, the SPx_PS0_CTR counter register [“RapidIO Port x Performance Statistics Counter 0 Register”]is disabled. 1 = Count priority 1 packets.	R/W	0
3	PS0_PRIO0	Performance Stats Reg PS0 Priority 0 Selection This value represents the packet priority 0 is selected for which performance stats are accumulated for in the SPx_PS0_CTR counter register [“RapidIO Port x Performance Statistics Counter 0 Register”]. 0 = If all PS0_PRIO[0..3] are set to zero, the SPx_PS0_CTR counter register [“RapidIO Port x Performance Statistics Counter 0 Register”]is disabled. 1 = Count priority 0 packets.	R/W	0
4:6	Reserved	N/A	R	0
7	PS0_DIR	Performance Stats Reg PS0 Direction Selection This value selects the direction (receiver vs. transmitter) for the SPx_PS0_CTR counter register [“RapidIO Port x Performance Statistics Counter 0 Register”]. 0 = Receiver Stats Counter Register 1 = Transmitter Stats Counter Register	R/W	0
8:12	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
13:15	PS0_TYPE	Performance Stats Reg PS0 Type Selection This value determines the type of performance statistics that is collected in the SPx_PS0_CTR counter register ["RapidIO Port x Performance Statistics Counter 0 Register". Retries are not counted as part of the data. 000 = Count all unicast request packets only. The response packets, maintenance packets, and maintenance packets with hop count of 0 are excluded from this counter. 001 = Count all unicast packet types. This counter includes all request, response, maintenance packets (including the maintenance packets with hop count 0). 010 = Count all retry control symbols only[a]. This setting is not available for the Internal Switch Port (Port 8). 011 = Count all control symbols (excluding retry control symbols)[a]. This setting is not available for the Internal Switch Port (Port 8). 100 = Count every 32-bits of unicast data. This counter counts all accepted unicast packets data (including header). 101 = Count all multicast packets only. 110 = Count all multicast control symbols[a]. 111 = Count every 32-bits of multicast data. This counter includes counting the entire multicast packet (including header).	R/W	0
16	PS1_PRIO3	Performance Stats Reg PS1 Priority 3 Selection This value represents the packet priority 3 is selected for which performance stats are accumulated for in the SPx_PS1_CTR counter register ["RapidIO Port x Performance Statistics Counter 1 Register"]. 0 = If all PS1_PRIO[0..3] are set to zero, the SPx_PS1_CTR counter register ["RapidIO Port x Performance Statistics Counter 1 Register"]is disabled. 1 = Count priority 3 packets.	R/W	0
17	PS1_PRIO2	Performance Stats Reg PS1 Priority 2 Selection This value represents the packet priority 2 is selected for which performance stats are accumulated for in the SPx_PS1_CTR counter register ["RapidIO Port x Performance Statistics Counter 1 Register"]. 0 = If all PS1_PRIO[0..3] are set to zero, the SPx_PS1_CTR counter register ["RapidIO Port x Performance Statistics Counter 1 Register"]is disabled. 1 = Count priority 2 packets.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
18	PS1_PRIO1	Performance Stats Reg PS1 Priority 1 Selection This value represents the packet priority 1 is selected for which performance stats are accumulated for in the SPx_PS1_CTR counter register [“RapidIO Port x Performance Statistics Counter 1 Register”]. 0 = If all PS1_PRIO[0..3] are set to zero, the SPx_PS1_CTR counter register [“RapidIO Port x Performance Statistics Counter 1 Register”]is disabled. 1 = Count priority 1 packets.	R/W	0
19	PS1_PRIO0	Performance Stats Reg PS1 Priority 0 Selection This value represents the packet priority 0 is selected for which performance stats are accumulated for in the SPx_PS1_CTR counter register [“RapidIO Port x Performance Statistics Counter 1 Register”]. 0 = If all PS1_PRIO[0..3] are set to zero, the SPx_PS1_CTR counter register [“RapidIO Port x Performance Statistics Counter 1 Register”]is disabled. 1 = Count priority 0 packets.	R/W	0
20:22	Reserved	N/A	R	0
23	PS1_DIR	Performance Stats Reg PS1 Direction Selection This value selects the direction (receiver vs. transmitter) for the SPx_PS1_CTR counter register [“RapidIO Port x Performance Statistics Counter 1 Register”]. 0 = Receiver Stats Counter Register 1 =Transmitter Stats Counter Register	R/W	0
24:28	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
29:31	PS1_TYPE	Performance Stats Reg PS1 Type Selection This value determines the type of performance statistics that is collected in the SPx_PS1_CTR counter register [“RapidIO Port x Performance Statistics Counter 1 Register”. Retries are not counted as part of the data. 000 = Count all unicast request packets only. The response packets, maintenance packets, and maintenance packets with hop count of 0 are excluded from this counter. 001 = Count all unicast packet types. This counter includes all request, response, maintenance packets (including the maintenance packets with hop count 0). 101 = Count all retry control symbols only[a]. This setting is not available for the Internal Switch Port (Port 8). 011 = Count all control symbols (excluding retry control symbols)[a]. This setting is not available for the Internal Switch Port (Port 8). 100 = Count every 32-bits of unicast data. This counter counts all accepted unicast packets data (including header). 101 = Count all multicast packets only. 110 = Count all multicast control symbols[a]. 111 = Count every 32-bits of multicast data. This counter includes counting the entire multicast packet (including header).	R/W	0

a. Control Symbol has no priority. In this case, any non-zero setting in bit0-3 increments the counter.

25.8.2 RapidIO Port x Performance Statistics Counter 2 and 3 Control Register

This register controls the performance statistics counters PS2 and PS3 registers. For every performance stats register SPx_PSCy (where y refers to the Performance Statistics counter PS2 and PS3), the following configuration are selected through this register: direction, type and priority.

- The PSy_DIR field determines the performance stats receiver versus transmitter direction application.
- The PSy_TYPE field assigns the type of stats collection (for example, packet, control symbol, and multicast) to be accumulated for a given SPx_PSCy.
- The PSy_PRIO[0..3] fields determine the priority of the packets for performance statistics collection through the "RapidIO Port x Performance Statistics Counter 2 Register" and "RapidIO Port x Performance Statistics Counter 3 Register". The SPx_PSCy can be disabled by selecting PSy_PRIO[0..3] to be set to 0. Setting PSy_PRIO[0..3] to all ones, allows for collecting performance statistics through the SPx_PSCy for all priority packets.

Register name: SP{0,1,2,3,4,5,6,8}_PSC2n3_CTRL **Reset value: 0x0000_0000**	**Register offset: 13024, 13124, 13224, 13324, 13424, 13524, 13624, 13824**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	PS2_PRIO 3	PS2_PRIO 2	PS2_PRIO 1	PS2_PRIO 0	Reserved			PS2_DIR
8:15	Reserved					PS2_TYPE		
16:23	PS3_PRIO 3	PS3_PRIO 2	PS3_PRIO 1	PS3_PRIO 0	Reserved			PS3_DIR
24:31	Reserved					PS3_TYPE		

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	PS2_PRIO3	Performance Stats Reg PS2 Priority 3 Selection This value represents the packet priority 3 is selected for which performance stats are accumulated for in the SPx_PS2_CTR counter register ["RapidIO Port x Performance Statistics Counter 4 and 5 Control Register"]. 0 = If all PS0_PRIO[0..3] are set to zero, the SPx_PS0_CTR counter register ["RapidIO Port x Performance Statistics Counter 4 and 5 Control Register"]is disabled. 1 = Count priority 3 packets.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
1	PS2_PRIO2	Performance Stats Reg PS2 Priority 2 Selection This value represents the packet priority 2 is selected for which performance stats are accumulated for in the SPx_PS2_CTR counter register ["RapidIO Port x Performance Statistics Counter 4 and 5 Control Register"]. 0 = If all PS0_PRIO[0..3] are set to zero, the SPx_PS0_CTR counter register ["RapidIO Port x Performance Statistics Counter 4 and 5 Control Register"]is disabled. 1 = Count priority 2 packets.	R/W	0
2	PS2_PRIO1	Performance Stats Reg PS2 Priority 1 Selection This value represents the packet priority 1 is selected for which performance stats are accumulated for in the SPx_PS2_CTR counter register ["RapidIO Port x Performance Statistics Counter 4 and 5 Control Register"]. 0 = If all PS0_PRIO[0..3] are set to zero, the SPx_PS0_CTR counter register ["RapidIO Port x Performance Statistics Counter 4 and 5 Control Register"]is disabled. 1 = Count priority 1 packets.	R/W	0
3	PS2_PRIO0	Performance Stats Reg PS2 Priority 0 Selection This value represents the packet priority 0 is selected for which performance stats are accumulated for in the SPx_PS2_CTR counter register ["RapidIO Port x Performance Statistics Counter 4 and 5 Control Register"]. 0 = If all PS0_PRIO[0..3] are set to zero, the SPx_PS0_CTR counter register ["RapidIO Port x Performance Statistics Counter 4 and 5 Control Register"]is disabled. 1 = Count priority 0 packets.	R/W	0
4:6	Reserved	N/A	R	0
7	PS2_DIR	Performance Stats Reg PS2 Direction Selection This value selects the direction (receiver vs. transmitter) for the SPx_PS2_CTR counter register ["RapidIO Port x Performance Statistics Counter 4 and 5 Control Register"]. 0 = Receiver Stats Counter Register 1 =Transmitter Stats Counter Register	R/W	0
8:12	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
13:15	PS2_TYPE	Performance Stats Reg PS2 Type Selection This value determines the type of performance statistics that is collected in the SPx_PS2_CTR counter register ["RapidIO Port x Performance Statistics Counter 4 and 5 Control Register". Retries are not counted as part of the data. 000 = Count all unicast request packets only. The response packets, maintenance packets, and maintenance packets with hop count of 0 are excluded from this counter. 001 = Count all unicast packet types. This counter includes all request, response, maintenance packets (including the maintenance packets with hop count 0). 010 = Count all retry control symbols only[a]. This setting is not available for the Internal Switch Port (Port 8). 011 = Count all control symbols (excluding retry control symbols)[a]. This setting is not available for the Internal Switch Port (Port 8). 100 = Count every 32-bits of unicast data. This counter counts all accepted unicast packets data (including header). 101 = Count all multicast packets only. 110 = Count all multicast control symbols[a]. 111 = Count every 32-bits of multicast data. This counter includes counting the entire multicast packet (including header).	R/W	0
16	PS3_PRIO3	Performance Stats Reg PS3 Priority 3 Selection This value represents the packet priority 3 is selected for which performance stats are accumulated for in the SPx_PS3_CTR counter register ["RapidIO Port x Performance Statistics Counter 3 Register"]. 0 = If all PS1_PRIO[0..3] are set to zero, the SPx_PS1_CTR counter register ["RapidIO Port x Performance Statistics Counter 3 Register"]is disabled. 1 = Count priority 3 packets.	R/W	0
17	PS3_PRIO2	Performance Stats Reg PS3 Priority 2 Selection This value represents the packet priority 2 is selected for which performance stats are accumulated for in the SPx_PS3_CTR counter register ["RapidIO Port x Performance Statistics Counter 3 Register"]. 0 = If all PS1_PRIO[0..3] are set to zero, the SPx_PS1_CTR counter register ["RapidIO Port x Performance Statistics Counter 3 Register"]is disabled. 1 = Count priority 2 packets.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
18	PS3_PRIO1	Performance Stats Reg PS3 Priority 1 Selection This value represents the packet priority 1 is selected for which performance stats are accumulated for in the SPx_PS3_CTR counter register ["RapidIO Port x Performance Statistics Counter 3 Register"]. 0 = If all PS1_PRIO[0..3] are set to zero, the SPx_PS1_CTR counter register ["RapidIO Port x Performance Statistics Counter 3 Register"]is disabled. 1 = Count priority 1 packets.	R/W	0
19	PS3_PRIO0	Performance Stats Reg PS3 Priority 0 Selection This value represents the packet priority 0 is selected for which performance stats are accumulated for in the SPx_PS3_CTR counter register ["RapidIO Port x Performance Statistics Counter 3 Register"]. 0 = If all PS1_PRIO[0..3] are set to zero, the SPx_PS1_CTR counter register ["RapidIO Port x Performance Statistics Counter 3 Register"]is disabled. 1 = Count priority 0 packets.	R/W	0
20:22	Reserved	N/A	R	0
23	PS3_DIR	Performance Stats Reg PS3 Direction Selection This value selects the direction (receiver vs. transmitter) for the SPx_PS3_CTR counter register ["RapidIO Port x Performance Statistics Counter 3 Register"]. 0 = Receiver Stats Counter Register 1 =Transmitter Stats Counter Register	R/W	0
24:28	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
29:31	PS3_TYPE	Performance Stats Reg PS3 Type Selection This value determines the type of performance statistics that is collected in the SPx_PS3_CTR counter register ["RapidIO Port x Performance Statistics Counter 3 Register". Retries are not counted as part of the data. 000 = Count all unicast request packets only. The response packets, maintenance packets, and maintenance packets with hop count of 0 are excluded from this counter. 001 = Count all unicast packet types. This counter includes all request, response, maintenance packets (including the maintenance packets with hop count 0). 010 = Count all retry control symbols only[a]. This setting is not available for the Internal Switch Port (Port 8). 011 = Count all control symbols (excluding retry control symbols)[a]. This setting is not available for the Internal Switch Port (Port 8). 100 = Count every 32-bits of unicast data. This counter counts all accepted unicast packets data (including header). 101 = Count all multicast packets only. 110 = Count all multicast control symbols[a]. 111 = Count every 32-bits of multicast data. This counter includes counting the entire multicast packet (including header).	R/W	0

a. Control Symbol has no priority. In this case, any non-zero setting in bit0-3 increments the counter.

25.8.3 RapidIO Port x Performance Statistics Counter 4 and 5 Control Register

This register controls the performance statistics counters PS4 and PS5 registers. For every performance stats register SPx_PSCy (where y refers to the Performance Statistics counter PS4 to PS5), the following configuration are selected through this register: direction, type and priority.

- The PSy_DIR field determines the performance stats receiver versus transmitter direction application.
- The PSy_TYPE field assigns the type of stats collection (for example, packet, control symbol, and multicast) to be accumulated for a given SPx_PSy_CTR.
- The PSy_PRIO[0..3] fields determine the priority of the packets for performance statistics collection through the "RapidIO Port x Performance Statistics Counter 4 Register" and "RapidIO Port x Performance Statistics Counter 5 Register". The SPx_PSCy can be disabled by selecting PSy_PRIO[0..3] to be set to 0. Setting PSy_PRIO[0..3] to all ones, allows for collecting performance statistics through the SPx_PSCy for all priority packets.

Register name: SP{0,1,2,3,4,5,6,8}_PSC4n5_CTRL **Reset value: 0x0000_0000**	**Register offset: 13028, 13128, 13228, 13328, 13428, 13528, 13628, 13828**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	PS4_PRIO 3	PS4_PRIO 2	PS4_PRIO 1	PS4_PRIO 0	Reserved			PS4_DIR
8:15	Reserved					PS4_TYPE		
16:23	PS5_PRIO 3	PS5_PRIO 2	PS5_PRIO 1	PS5_PRIO 0	Reserved			PS5_DIR
24:31	Reserved					PS5_TYPE		

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	PS4_PRIO3	Performance Stats Reg PS4 Priority 3 Selection This value represents the packet priority 3 is selected for which performance stats are accumulated for in the SPx_PS4_CTR counter register ["RapidIO Port x Performance Statistics Counter 4 Register"]. 0 = If all PS0_PRIO[0..3] are set to zero, the SPx_PS0_CTR counter register ["RapidIO Port x Performance Statistics Counter 4 Register"]is disabled. 1 = Count priority 3 packets.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
1	PS4_PRIO2	Performance Stats Reg PS4 Priority 2 Selection This value represents the packet priority 2 is selected for which performance stats are accumulated for in the SPx_PS4_CTR counter register [“RapidIO Port x Performance Statistics Counter 4 Register”]. 0 = If all PS0_PRIO[0..3] are set to zero, the SPx_PS0_CTR counter register [“RapidIO Port x Performance Statistics Counter 4 Register”]is disabled. 1 = Count priority 2 packets.	R/W	0
2	PS4_PRIO1	Performance Stats Reg PS4 Priority 1 Selection This value represents the packet priority 1 is selected for which performance stats are accumulated for in the SPx_PS4_CTR counter register [“RapidIO Port x Performance Statistics Counter 4 Register”]. 0 = If all PS0_PRIO[0..3] are set to zero, the SPx_PS0_CTR counter register [“RapidIO Port x Performance Statistics Counter 4 Register”]is disabled. 1 = Count priority 1 packets.	R/W	0
3	PS4_PRIO0	Performance Stats Reg PS4 Priority 0 Selection This value represents the packet priority 0 is selected for which performance stats are accumulated for in the SPx_PS4_CTR counter register [“RapidIO Port x Performance Statistics Counter 4 Register”]. 0 = If all PS0_PRIO[0..3] are set to zero, the SPx_PS0_CTR counter register [“RapidIO Port x Performance Statistics Counter 4 Register”]is disabled. 1 = Count priority 0 packets.	R/W	0
4:6	Reserved	N/A	R	0
7	PS4_DIR	Performance Stats Reg PS4 Direction Selection This value selects the direction (receiver vs. transmitter) for the SPx_PS4_CTR counter register [“RapidIO Port x Performance Statistics Counter 4 Register”]. 0 = Receiver Stats Counter Register 1 = Transmitter Stats Counter Register	R/W	0
8:12	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
13:15	PS4_TYPE	Performance Stats Reg PS4 Type Selection This value determines the type of performance statistics that is collected in the SPx_PS4_CTR counter register ["RapidIO Port x Performance Statistics Counter 4 Register". Retries are not counted as part of the data. 000 = Count all unicast request packets only. The response packets, maintenance packets, and maintenance packets with hop count of 0 are excluded from this counter. 001 = Count all unicast packet types. This counter includes all request, response, maintenance packets (including the maintenance packets with hop count 0). 010 = Count all retry control symbols only[a]. This setting is not available for the Internal Switch Port (Port 8). 011 = Count all control symbols (excluding retry control symbols)[a]. This setting is not available for the Internal Switch Port (Port 8). 100 = Count every 32-bits of unicast data. This counter counts all accepted unicast packets data (including header). 101 = Count all multicast packets only. 110 = Count all multicast control symbols[a]. 111 = Count every 32-bits of multicast data. This counter includes counting the entire multicast packet (including header).	R/W	0
16	PS5_PRIO3	Performance Stats Reg PS5 Priority 3 Selection This value represents the packet priority 3 is selected for which performance stats are accumulated for in the SPx_PS5_CTR counter register ["RapidIO Port x Performance Statistics Counter 5 Register"]. 0 = If all PS1_PRIO[0..3] are set to zero, the SPx_PS1_CTR counter register ["RapidIO Port x Performance Statistics Counter 5 Register"]is disabled. 1 = Count priority 3 packets.	R/W	0
17	PS5_PRIO2	Performance Stats Reg PS5 Priority 2 Selection This value represents the packet priority 2 is selected for which performance stats are accumulated for in the SPx_PS5_CTR counter register ["RapidIO Port x Performance Statistics Counter 5 Register"]. 0 = If all PS1_PRIO[0..3] are set to zero, the SPx_PS1_CTR counter register ["RapidIO Port x Performance Statistics Counter 5 Register"]is disabled. 1 = Count priority 2 packets.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
18	PS5_PRIO1	Performance Stats Reg PS5 Priority 1 Selection This value represents the packet priority 1 is selected for which performance stats are accumulated for in the SPx_PS5_CTR counter register [“RapidIO Port x Performance Statistics Counter 5 Register”]. 0 = If all PS1_PRIO[0..3] are set to zero, the SPx_PS1_CTR counter register [“RapidIO Port x Performance Statistics Counter 5 Register”]is disabled. 1 = Count priority 1 packets.	R/W	0
19	PS5_PRIO0	Performance Stats Reg PS5 Priority 0 Selection This value represents the packet priority 0 is selected for which performance stats are accumulated for in the SPx_PS5_CTR counter register [“RapidIO Port x Performance Statistics Counter 5 Register”]. 0 = If all PS1_PRIO[0..3] are set to zero, the SPx_PS1_CTR counter register [“RapidIO Port x Performance Statistics Counter 5 Register”]is disabled. 1 = Count priority 0 packets.	R/W	0
20:22	Reserved	N/A	R	0
23	PS5_DIR	Performance Stats Reg PS5 Direction Selection This value selects the direction (receiver vs. transmitter) for the SPx_PS5_CTR counter register [“RapidIO Port x Performance Statistics Counter 5 Register”]. 0 = Receiver Stats Counter Register 1 =Transmitter Stats Counter Register	R/W	0
24:28	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
29:31	PS5_TYPE	Performance Stats Reg PS5 Type Selection This value determines the type of performance statistics that is collected in the SPx_PS5_CTR counter register [“RapidIO Port x Performance Statistics Counter 5 Register”. Retries are not counted as part of the data. 000 = Count all unicast request packets only. The response packets, maintenance packets, and maintenance packets with hop count of 0 are excluded from this counter. 001 = Count all unicast packet types. This counter includes all request, response, maintenance packets (including the maintenance packets with hop count 0). 010 = Count all retry control symbols only[a]. This setting is not available for the Internal Switch Port (Port 8). 011 = Count all control symbols (excluding retry control symbols)[a]. This setting is not available for the Internal Switch Port (Port 8). 100 = Count every 32-bits of unicast data. This counter counts all accepted unicast packets data (including header). 101 = Count all multicast packets only. 110 = Count all multicast control symbols[a]. 111 = Count every 32-bits of multicast data. This counter includes counting the entire multicast packet (including header).	R/W	0

a. Control Symbol has no priority. In this case, any non-zero setting in bit0-3 increments the counter.

25.8.4 RapidIO Port x Performance Statistics Counter 0 Register

This register collects performance statistics. These counters provide the means of accumulating statistics for the purposes of performance monitoring measurements: throughput and latency.

The PS0_CTR counter collects performance statistics information based on the configuration fields specified in the "RapidIO Port x Performance Statistics Counter 0 and 1 Control Register".

The PS0_CTR counter value is writable for testing purposes. This counter fills when it reaches its maximum value of 0xFFFFFFFF, and is cleared on read. The PS0_CTR is enabled when the PS0_PRIO[0..3] value in the "RapidIO Port x Performance Statistics Counter 0 and 1 Control Register" is configured to a value other than 0.

Register name: SP{0,1,2,3,4,5,6,8}_PSC0 **Reset value: 0x0000_0000**	**Register offset: 13040, 13140, 13240, 13340, 13440, 13540, 13640, 13840**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	PS0_CTR							
8:15	PS0_CTR							
16:23	PS0_CTR							
24:31	PS0_CTR							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:31	PS0_CTR	This counter collects performance statistics based on the configurations specified through the SPx_PSy_CTRL0 register ("RapidIO Port x Performance Statistics Counter 0 and 1 Control Register"). A read clears this register.	R/W	0

25.8.5 RapidIO Port x Performance Statistics Counter 1 Register

This register collects performance statistics. These counters provide the means of accumulating statistics for the purposes of performance monitoring measurements: throughput and latency.

The PS1_CTR counter collects performance statistics information based on the configuration fields specified in the "RapidIO Port x Performance Statistics Counter 0 and 1 Control Register".

The PS1_CTR counter value is writable for testing purposes. This counter fills when it reaches its maximum value of 0xFFFFFFFF, and is cleared on read. The PS1_CTR is enabled, when PS1_PRIO[0..3] value in the "RapidIO Port x Performance Statistics Counter 0 and 1 Control Register" is configured to a value other than 0.

Register name: SP{0,1,2,3,4,5,6,8}_PSC1 **Reset value: 0x0000_0000**	**Register offset: 13044, 13144, 13244, 13344, 13444, 13544, 13644, 13844**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	PS1_CTR							
8:15	PS1_CTR							
16:23	PS1_CTR							
24:31	PS1_CTR							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:31	PS1_CTR	This counter collects performance statistics based on the configurations specified through the SPx_PSy_CTRL0 register ("RapidIO Port x Performance Statistics Counter 0 and 1 Control Register"). A read clears this register.	R/W	0

25.8.6 RapidIO Port x Performance Statistics Counter 2 Register

This register collects performance statistics. These counters provide the means of accumulating statistics for the purposes of performance monitoring measurements: throughput and latency.

The PS2_CTR counter collects performance statistics information based on the configuration fields specified in the "RapidIO Port x Performance Statistics Counter 2 and 3 Control Register".

The PS2_CTR counter value is writable for testing purposes. This counter fills when it reaches its maximum value of 0xFFFFFFFF, and is cleared on read. The PS2_CTR is enabled, when PS2_PRIO[0..3] value in the "RapidIO Port x Performance Statistics Counter 2 and 3 Control Register" is configured to a value other than 0.

Register name: SP{0,1,2,3,4,5,6,8}_PSC2 **Reset value: 0x0000_0000**	**Register offset: 13048, 13148, 13248, 13348, 13448, 13548, 13648, 13848**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	PS2_CTR							
8:15	PS2_CTR							
16:23	PS2_CTR							
24:31	PS2_CTR							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:31	PS2_CTR	This counter collects performance statistics based on the configurations specified using the "RapidIO Port x Performance Statistics Counter 2 and 3 Control Register" register. A read clears this register.	R/W	0

25.8.7 RapidIO Port x Performance Statistics Counter 3 Register

This register collects performance statistics. These counters provide the means of accumulating statistics for the purposes of performance monitoring measurements: throughput and latency.

The PS3_CTR counter collects performance statistics information based on the configuration fields specified in the "RapidIO Port x Performance Statistics Counter 2 and 3 Control Register".

The PS3_CTR counter value is writable for testing purposes. This counter fills when it reaches its maximum value of 0xFFFFFFFF, and is cleared on read. The PS3_CTR is enabled, when PS3_PRIO[0..3] value in the "RapidIO Port x Performance Statistics Counter 2 and 3 Control Register" is configured to a value other than 0.

Register name: SP{0,1,2,3,4,5,6,8}_PSC3 **Reset value: 0x0000_0000**	**Register offset: 1304C, 1314C, 1324C, 1334C, 1344C, 1354C, 1364C, 1384C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	PS3_CTR							
8:15	PS3_CTR							
16:23	PS3_CTR							
24:31	PS3_CTR							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:31	PS3_CTR	This counter collects performance statistics based on the configurations specified using the SPx_PSy_CTRL0 ("RapidIO Port x Performance Statistics Counter 2 and 3 Control Register") register. A read clears this register.	R/W	0

25.8.8 RapidIO Port x Performance Statistics Counter 4 Register

This register collects performance statistics. These counters provide the means of accumulating statistics for the purposes of performance monitoring measurements: throughput and latency.

The PS4_CTR counter collects performance statistics information based on the configuration fields specified in the "RapidIO Port x Performance Statistics Counter 4 and 5 Control Register".

The PS4_CTR counter value is writable for testing purposes.This counter fills when it reaches its maximum value of 0xFFFFFFFF, and is cleared on read. The PS4_CTR is enabled, when PS4_PRIO[0..3] value in the "RapidIO Port x Performance Statistics Counter 4 and 5 Control Register" is configured to a value other than 0.

Register name: SP{0,1,2,3,4,5,6,8}_PSC4 **Reset value: 0x0000_0000**	**Register offset: 13050, 13150, 13250, 13350, 13450, 13550, 13650, 13850**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	PS4_CTR							
8:15	PS4_CTR							
16:23	PS4_CTR							
24:31	PS4_CTR							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:31	PS4_CTR	This counter collects performance statistics based on the configurations specified using the "RapidIO Port x Performance Statistics Counter 4 and 5 Control Register". A read clears this register.	R/W	0

25.8.9 RapidIO Port x Performance Statistics Counter 5 Register

This register collects performance statistics. These counters provide the means of accumulating statistics for the purposes of performance monitoring measurements: throughput and latency.

The PS5_CTR counter collects performance statistics information based on the configuration fields specified in the "RapidIO Port x Performance Statistics Counter 4 and 5 Control Register".

The PS5_CTR counter value is writable for testing purposes. This counter fills when it reaches its maximum value of 0xFFFFFFFF, and is cleared on read. The PS5_CTR is enabled, when PS5_PRIO[0..3] value in the "RapidIO Port x Performance Statistics Counter 4 and 5 Control Register" is configured to a value other than 0.

Register name: SP{0,1,2,3,4,5,6,8}_PSC5 **Reset value: 0x0000_0000**	**Register offset: 13054, 13154, 13254, 13354, 13454, 13554, 13654, 13854**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	PS5_CTR							
8:15	PS5_CTR							
16:23	PS5_CTR							
24:31	PS5_CTR							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:31	PS5_CTR	This counter collects performance statistics based on the configurations specified using the "RapidIO Port x Performance Statistics Counter 4 and 5 Control Register". A read clears this register.	R/W	0

25.8.10 RapidIO Port x Transmitter Output Queue Depth Threshold Register

Queue depth registers allow for the rapid detection and notification of congestion. This register sets the Transmitter Queue Depth threshold, which is used in conjunction with "RapidIO Port x Transmitter Output Queue Congestion Status Register" to monitor congestion on the output buffers.

This register also sets the CONG_PERIOD, which is used in conjunction with the "RapidIO Port x Transmitter Output Queue Congestion Period Register" to determine how long the output buffers are in a congestion state.

Note the for the Internal Switch Port (Port 8) that CONG_PERIOD and DEPTH do not control any functionality.

Register name: SP{0,1,2,3,4,5,6,8}_TX_Q_D_THRESH **Reset value: 0x0000_0000**	**Register offset: 13080, 13180, 13280, 13380, 13480, 13580, 13680, 13880**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	CONG_PERIOD							
8:15	CONG_PERIOD							
16:23	DEPTH				Reserved	LEAK_RT		
24:31	LEAK_RT							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:15	CONG_PERIOD	This hexadecimal value is programmed by software to indicate the maximum number of clock cycles that the output buffer can be in a continuous congestion state. The congestion state is determined based on the DEPTH and SPx_TX_Q_STATUS. The programmed CONG_PERIOD value is then used as follows: 0000 = CONG_PERIOD_CTR (in "RapidIO Port x Transmitter Output Queue Congestion Period Register") is disabled. 0001 = Reserved 0002 = For every clock cycle that the output buffer is in continuous congestion state, increment the CONG_PERIOD_CTR by 1. FFFF = For every 64K clock cycles that the output buffer is in continuous congestion state, increment the CONG_PERIOD_CTR by 1. The clock period is always 3.2 nsec. This controls no functionality for Port 8 (Internal Switch Port).	R/W	0x0000

(Continued)

Bits	Name	Description	Type	Reset Value
16:19	DEPTH	This number decides the congestion state of the output buffers. 0x0 - Disables counting. 0x1 - Increment counter if buffer has 2 or more packets 0x2 - Increment counter if buffer has 3 or more packets ... 0x7 - Increment counter if buffer has 8 packets 0x8-0xF - Reserved This field does not configure any functionality for Port 8 (Internal Switch Port).	R/W	0x0
20	Reserved	N/A	R	0
21:31	LEAK_RT	This value is the leak rate for both the receiver and transmitter congestion counters. When this time period expires, the CONG_CTR values for both transmitter ("RapidIO Port x Transmitter Output Queue Congestion Status Register") and receiver ("RapidIO Port x Receiver Input Queue Congestion Status Register") are decremented by 1. For reference clock frequency of 125 MHz, the following intervals apply: 0 = Leak rate is disabled 1 = Decrement CONG_CTR every 2.048µs 2 = Decrement CONG_CTR every 2*2.048µs = 4.192µs ... 2047- Decrement CONG_CTR every 2047*2.048µs = 4.19ms. For reference clock frequency of 156.25 MHz, the following intervals apply: 0 = Leak rate is disabled 1 = Decrement CONG_CTR every 1.6384µs 2 = Decrement CONG_CTR every 2*1.6384µs = 3.277µs ... 2047- Decrement CONG_CTR every 2047*1.6384µs = 3.35ms. Note: This controls functionality only for the receiver congestion counters for Port 8 (Internal Switch Port).	R/W	0x0

25.8.11 RapidIO Port x Transmitter Output Queue Congestion Status Register

This register monitors data congestion in the output buffer. New packets accumulate in the output buffers, destined for the Switch ISF. When the number of buffers in use equals or exceeds the threshold set in DEPTH field of the "RapidIO Port x Transmitter Output Queue Depth Threshold Register", the CONG_CTR field in "RapidIO Port x Receiver Input Queue Congestion Status Register" is incremented.

The CONG_CTR counter value is writable for testing purposes. This counter stops counting when it reaches its maximum value. Writing 1 to the OUTB_DEPTH bit in the "RapidIO Port x Interrupt Status Register" causes this counter to be reset to 0. The CONG_CTR is enabled, when CONG_THRESH value is configured to a value other than 0. The CONG_CTR value is decremented by 1 if it is not read within the Error Rate Bias frequency as specified by the ERR_RB field in the "RapidIO Port x Error Rate CSR".

If the CONG_CTR equals or exceeds the threshold CONG_THRESH, the maskable OUTB_DEPTH interrupt is generated.

Note the for the Internal Switch Port (Port 8) this register does not control any functionality.

Register name: SP{0,1,2,3,4,5,6,8}_TX_Q_STATUS **Reset value: 0x0000_0000**	**Register offset: 13084, 13184, 13284, 13384, 13484, 13584, 13684, 13884**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	CONG_CTR							
8:15	CONG_CTR							
16:23	CONG_THRESH							
24:31	CONG_THRESH							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:15	CONG_CTR	Output Queue Depth Count The number of times that the output queue exceeds the threshold DEPTH field of the "RapidIO Port x Transmitter Output Queue Depth Threshold Register". The count is incremented by 1 when a packet is received. This counter counts up to 0xFFFFand remains at 0xFFFF until reset. The counter is reset when 1 is written to the OUTB_DEPTH (see "RapidIO Port x Interrupt Status Register") status bit. The counter is enabled if CONG_THRESH is set to a value other than 0.The CONG_CTR value is decremented by 1 when the LEAK_RT [see "RapidIO Port x Transmitter Output Queue Depth Threshold Register"] time period expires. The CONG_CTR value never goes below 0. This controls no functionality for Port 8 (Internal Switch Port).	R/W	0x0000

(Continued)

Bits	Name	Description	Type	Reset Value
16:31	CONG_THRESH	Output Queue Depth Threshold If the CONG_CTR count is equal to the value in this field, an interrupt is reported to the system using the OUTB_DEPTH status bit in the SPx_INT_STATUS register (see "RapidIO Port x Interrupt Status Register"). Setting the CONG_THRES to zero, disables the CONG_CTR. This controls no functionality for Port 8 (Internal Switch Port).	R/W	0x0000

25.8.12 RapidIO Port x Transmitter Output Queue Congestion Period Register

This register monitors the duration of time that the output buffer is in congestion state.

The CONG_PERIOD_CTR counter value is incremented for every N clock cycles specified by the CONG_PERIOD field in the "RapidIO Port x Transmitter Output Queue Depth Threshold Register", while the output buffer is under congestion state. This counter represents the amount of time that the output buffer is under congestion state.

The CONG_PERIOD_CTR counter value is writable for testing purposes. This counter stops counting when it reach its maximum value. Reading the CONG_PERIOD_CTR clears the counter value. The CONG_PERIOD_CTR can be disabled when the CONG_PERIOD field in the "RapidIO Port x Transmitter Output Queue Depth Threshold Register" is set to 0.

Note the for the Internal Switch Port (Port 8) this register does not control any functionality.

Register name: SP{0,1,2,3,4,5,6,8}_TX_Q_PERIOD **Reset value: 0x0000_0000**	**Register offset: 13088, 13188, 13288, 13388, 13488, 13588, 13688, 13888**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	CONG_PERIOD_CTR							
8:15	CONG_PERIOD_CTR							
16:23	CONG_PERIOD_CTR							
24:31	CONG_PERIOD_CTR							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:31	CONG_PERIOD_CTR	Output Queue Congestion Period Count Each time the output buffer enters a congestion state, this counter is incremented for every N clock cycles (as specified in CONG_PERIOD field of the "RapidIO Port x Transmitter Output Queue Depth Threshold Register"). This counter counts up to 0xFFFFand remains at 0xFFFF until reset. Note: The counter is reset when read. The counter is enabled if CONG_PERIOD is set to a value other than 0. This controls no functionality for Port 8 (Internal Switch Port).	R/W	0x0000

25.8.13 RapidIO Port x Receiver Input Queue Depth Threshold Register

Queue depth registers allow for the rapid detection and notification of congestion. This register sets the Receiver Queue Depth threshold, which is used in conjunction with "RapidIO Port x Receiver Input Queue Congestion Status Register" to monitor congestion on the input buffers.

This register also sets the CONG_PERIOD, which is used in conjunction with the "RapidIO Port x Receiver Input Queue Congestion Period Register" to determine how long the input buffers are in a congestion state.

Register name: SP{0,1,2,3,4,5,6,8}_RX_Q_D_THRESH **Reset value: 0x0000_0000**	**Register offset: 13090, 13190, 13290, 13390, 13490, 13590, 13690, 13890**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	CONG_PERIOD							
8:15	CONG_PERIOD							
16:23	DEPTH				Reserved			
24:31	Reserved							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:15	CONG_PERIOD	This value is programmed by SW to indicate the maximum number of clock cycles that the output buffer can be in a continuous congestion state. The congestion state is determined based on the DEPTH and SPx_RX_Q_STATUS. The programmed CONG_PERIOD value is then used as follows: 0000 = CONG_PERIOD_CTR is disabled. 0001 = Reserved 0002 = For every clock cycle that the output buffer is in continuous congestion state, increment the CONG_PERIOD_CTR by 1. FFFF = For every 64K clock cycles that the output buffer is in continuous congestion state, increment the CONG_PERIOD_CTR by 1. Note: The Internal Switch Port clock period is different from the RapidIO ports. The Internal Switch Port clock period is the same as the reference clock period: • For a 156.25 MHz reference clock, the CONG_PERIOD clock period is 6.4 nsec • For a 125 MHz reference clock, the CONG_PERIOD clock period is 8 nsec For other RapidIO ports, the clock period is always 3.2 nsec.	R/W	0x0000

(Continued)

Bits	Name	Description	Type	Reset Value
16:19	DEPTH	This number decides the congestion state of the input buffers. 0x0 - Disables counting. 0x1 - Increment counter if buffer has 2 or more packets 0x2 - Increment counter if buffer has 3 or more packets ... 0x7 - Increment counter if buffer has 8 packets 0x8-0xF - Reserved	R/W	0
20:31	Reserved	N/A	R	0

25.8.14 RapidIO Port x Receiver Input Queue Congestion Status Register

This register monitors data congestion in the input buffer.

New packets accumulate in the input buffers, destined for the Switch ISF. When the number of buffers in use equals or exceeds the threshold set in DEPTH field of the "RapidIO Port x Receiver Input Queue Depth Threshold Register", the CONG_CTR field in SPx_R_Q_STATUS is incremented.

The CONG_CTR counter value is writable for testing purposes. This counter stops counting when it reaches its maximum value. Writing 1 to the INB_DEPTH bit in the "RapidIO Port x Interrupt Status Register" causes this counter to be reset to 0. The CONG_CTR is enabled, when CONG_THRESH value is configured to a value other than 0. The CONG_CTR value is decremented by 1 if it is not read within the Error Rate Bias frequency as specified by the ERR_RB field in the "RapidIO Port x Error Rate CSR".

If the CONG_CTR equals or exceeds the threshold CONG_THRESH, the maskable INB_DEPTH interrupt is generated.

Register name: SP{0,1,2,3,4,5,6,8}_RX_Q_STATUS **Reset value: 0x0000_0000**	**Register offset: 13094, 13194, 13294, 13394, 13494, 13594, 13694, 13894**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	CONG_CTR							
8:15	CONG_CTR							
16:23	CONG_THRESH							
24:31	CONG_THRESH							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:15	CONG_CTR	Input Queue Depth Count The number of times that the input queue exceeds the threshold DEPTH field of the "RapidIO Port x Receiver Input Queue Depth Threshold Register". The count is incremented by 1 when a packet is received. This counter counts up to 0xFFFFand remains at 0xFFFF until reset. The counter is reset when 1 is written to the INB_DEPTH status bit (see "RapidIO Port x Interrupt Status Register"). The counter is enabled if CONG_THRES is set to a value other than 0. The CONG_CTR value is decremented by 1 if it is not read within the Error Rate Bias frequency as specified by the ERR_RB field in the "RapidIO Port x Error Rate CSR".	R/W	0x0000

(Continued)

Bits	Name	Description	Type	Reset Value
16:31	CONG_THRESH	Input Queue Depth Threshold If the CONG_CTR count is equal to the value in this field, an interrupt is reported to the system using the INB_DEPTH status bit in the SPx_INT_STATUS register [see "RapidIO Port x Interrupt Status Register"]. Setting the CONG_THRESH to zero, disables the CONG_CTR.	R/W	0x0000

25.8.15 RapidIO Port x Receiver Input Queue Congestion Period Register

This register monitors the duration of time that the input buffer is in congestion state.

The CONG_PERIOD_CTR counter value is incremented for every N clock cycles specified by the CONG_PERIOD field in the "RapidIO Port x Receiver Input Queue Depth Threshold Register", while the input buffer is under congestion state. This counter represents the amount of time that the input buffer is under congestion state.

The CONG_PERIOD_CTR counter value is writable for testing purposes. This counter stops counting when it reach its maximum value. Reading the CONG_PRIOD_CTR clears the register. The CONG_PERIOD_CTR can be disabled when the CONG_PERIOD field in the "RapidIO Port x Receiver Input Queue Depth Threshold Register" is set to 0.

Register name: SP{0,1,2,3,4,5,6,8}_RX_Q_PERIOD **Reset value: 0x0000_0000**	**Register offset: 13098, 13198, 13298, 13398, 13498, 13598, 13698, 13898**

Bits	0	1	2	3	4	5	6	7
00:7	CONG_PERIOD_CTR							
8:15	CONG_PERIOD_CTR							
16:23	CONG_PERIOD_CTR							
24:31	CONG_PERIOD_CTR							

Bits	Name	Description	Type	Reset Value
0:31	CONG_PERIOD_CTR	Input Queue Congestion Period Count Each time the input buffer enters a congestion state, this counter is incremented for every N clock cycles (as specified in CONG_PERIOD field of the "RapidIO Port x Transmitter Output Queue Depth Threshold Register"). This counter counts up to 0xFFFFand remains at 0xFFFF until reset. The counter is reset when read. The counter is enabled if CONG_PERIOD is set to a value other than 0. Note: Reading clears this register	R/W	0x0000

25.8.16 RapidIO Port x Reordering Counter Register

When a packet cannot make forward progress due to internal switching congestion, the Switch ISF selects packets in an order different from the order in which the packets were received. Each time this occurs, it is counted as a “reorder” event in this register.

The Input Reordering Threshold (THRESH) defines the number of times the Input Reordering Count is incremented before an interrupt is generated, if enabled (see “RapidIO Port x Interrupt Status Register”).

Register name: SP{0,1,2,3,4,5,6,8}_REORDER_CTR **Reset value: 0x0000_FFFF**	**Register offset: 130A0, 131A0, 132A0, 133A0, 134A0, 135A0, 136A0, 138A0**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	CTR							
8:15	CTR							
16:23	THRESH							
24:31	THRESH							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:15	CTR	Reorder Counter This counter is updated every time the input queue is reordered. This counter counts up to 0xFFFFand remains at 0xFFFF until reset. The counter is reset when 1 is written to the INB_RDR status bit in“RapidIO Port x Interrupt Status Register”. The counter is enabled if the THRESH is configured to a value other than 0.	R/W	0x0000
16:31	THRESH	Input Reordering Threshold When CTR equals THRESH, the maskable interrupt “INB_RDR” in “RapidIO Port x Interrupt Status Register” is generated. Setting the THRESH value to 0 disables the CTR.	R/W	0xFFFF

25.9 Serial Port Electrical Layer Registers

The Serial Port Electrical Layer Registers are not defined in the *RapidIO Interconnect Specification (Revision 1.3)*.

These registers are reset by several sources of Tsi620 reset (see "Resets"). The registers within a port are also reset by a port reset, performed by powering up and down the port. For more information on Tsi620 reset design, see "Clock, Reset, Power-up, and Initialization Options". It is possible to override reset values of writable fields, and some read-only fields, using the I^2C register loading capability on boot. For more information on the use of I^2C Interface register loading capability, see "I2C Interface".

Software must not access reserved addresses or bits, as these can affect device operation in non-deterministic ways.

Table 157: IDT-Specific RapidIO Registers

Port	Register Offset	Description
BC	10000	Broadcast addresses. These registers affect all the ports.
SP0	11000	1x/4x Serial port
SP1	11100	1x Serial port
SP2	11200	1x/4x Serial port
SP3	11300	1x Serial port
SP4	11400	1x/4x Serial port
SP5	11500	1x Serial port
SP6	11600	1x/4x Serial port
SP7	11700	1x Serial port
SP8	11800	Internal Switch Port

The registers in Table 158 are accessible even when the RapidIO ports are in reset or powered down.

Table 158: Serial Port Electrical Layer Registers

MAC	Register Offset	Description
MAC0	130B0	Ports 0 and 1
MAC2	132B0	Ports 2 and 3
MAC4	134B0	Ports 4 and 5
MAC6	136B0	Ports 6 and 7

25.9.1 BYPASS_INIT Functionality

The traffic affecting SerDes controls are locked by the BYPASS_INIT bit "RapidIO SMAC x SerDes Configuration Global Register". The following bits and fields are unlocked when the BYPASS_INIT bit is set to 1:

- MPLL_CK_OFF in the SMACx_CFG_GBL register
- SERDES_RESET in the SMACx_CFG_GBL register
- MPLL_pwron in the SMACx_CFG_GBL register
- TX_EN in the SMACx_CFG_CH3 register
- TX_EN in the SMACx_CFG_CH2 register
- TX_EN in the SMACx_CFG_CH1 register
- TX_EN in the SMACx_CFG_CH0 register
- RX_PLL_PWRON in the SMACx_CFG_CH0 register
- RX_PLL_PWRON in the SMACx_CFG_CH1 register
- RX_PLL_PWRON in the SMACx_CFG_CH2 register
- RX_PLL_PWRON in the SMACx_CFG_CH3 register
- RX_EN in the SMACx_CFG_CH0 register
- RX_EN in the SMACx_CFG_CH1 register
- RX_EN in the SMACx_CFG_CH2 register
- RX_EN in the SMACx_CFG_CH3 register

25.9.2 RapidIO SMAC x SerDes Configuration Channel 0 Register

This register controls RapidIO port SerDes channel 0. For more information on port configuration after power-down, see "Port Power Down".

Register name: SMAC{0,2,4,6}_CFG_CH0 Reset value: 0xA03C_E513	Register offset: 130B0, 132B0, 134B0, 136B0

Bits	0	1	2	3	4	5	6	7
00:07	HALF_RATE	TX_CALC	Reserved		ALB_EN	TX_ATTEN[2:0]		
08:15	Reserved	TX_EN[2:0]			TX_BOOST[3:0]			
16:23	RX_PLL_PWRON	RX_EN	DPLL_RESET	Reserved		RX_EQ_VAL[2:0]		
24:31	RX_DPLL_MODE[2:0]			TX_CKO_EN	LOS_CTL[1:0]		RX_ALIGN_EN	Reserved

Bits	Name	Description	Type	Reset Value
0	HALF_RATE	Baud rate control. Set to 1 if operating at 1.25Gbps; Otherwise, set to 0 for 2.5G and 3.125Gbps. This value should be consistent with the other HALF_RATE settings within a SerDes. Different HALF_RATE settings within a SerDes configuration can lead to unpredictable behavior. Note: This bit setting overrides the setting of the IO_SPEED bit in the RapidIO MAC x SerDes Configuration Channel x register (see "RapidIO SMAC x SerDes Configuration Channel 0 Register"). This bit controls the generation of the clock signals for the FPGA Interface.	R/W	1
1	TX_CALC	A rising edge causes a recalculation of the transmitter attenuation and boost configuration. The rising edge may be generated by writing a 0 and then a 1 to the register bit. This bit is not used by the FPGA Interface.	R/W	0
2:3	Reserved	N/A	R/W	0x2
4	ALB_EN	Analog Loopback Enable 0 = Normal operation 1 = Transmit data for all lanes is looped back to Rx Note: This bit is not functional for Port 6 (FPGA Interface).	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
5:7	TX_ATTEN[2:0]	Transmit Attenuation control, provides discrete driver attenuation factors of either: {16, 14, 12, 10, 9, 8} sixteenths of full drive level. 0b000 = 16/16 0b001 = 14/16 0b010 = 12/16 0b011 = 10/16 0b100 = 9/16 0b101 = 8/16 0b11x = reserved Note: TX_ATTEN and TX_BOOST should not be used together. Setting TX_ATTEN to values other than 0b000 disables TX_BOOST (that is, 0dB of boost). This bit is not used by the FPGA Interface.	R/W	0x0
8	Reserved	N/A	R	0
9:11	TX_EN[2:0]	Transmitter enable 000 = off, no clocks operating 011 = transmitter fully enabled, all clocks operating Other values are Reserved. This bit field is not used within Port 6.	R/W	0x3
12:15	TX_BOOST[3:0]	Transmit Boost control. Programmed boost value (ratio of drive level of transition bit to non-transition bit) is: boost = -20*log(1-(tx_boost[3:0]+0.5)/32)dB, except that setting tx_boost to 0 produces 0dB of boost. This produces results up to 5.75dB in steps of ~0.37dB. Note: TX_ATTEN and TX_BOOST should not be used together. Setting TX_ATTEN to values other than 0b000 disables TX_BOOST (that is, 0dB of boost). This bit is not used by the FPGA Interface.	R/W	0xC
16	RX_PLL_PWRON	Power up/reset the receive PLL 0 = Rx PLL off 1 = Rx PLL on This bit is not used by the FPGA Interface.	R/W	0x1
17	RX_EN	Enable receive clock and data outputs. rx_en may only be asserted when both rx_pll_pwron and rx_pll_state are both asserted. Set to 1 to enable Receive Data; 0 to disable Receive Data This bit is not used by the FPGA Interface.	R/W	0x1
18	DPLL_RESET	A rising edge resets the frequency register of the DPLL. The rising edge can be generated by writing a “0” and then a “1” to the register bit. This bit is not used by the FPGA Interface.	R/W	0x1

(Continued)

Bits	Name	Description	Type	Reset Value
19:20	Reserved	N/A	R/W	0
21:23	RX_EQ_VAL [2:0]	Receive Equalization control. Internal linear equalizer boost is approximately = (rx_eq_val+1)*0.5dB Example: 0b100 = 2.5dB boost This bit is not used by the FPGA Interface.	R/W	0x5
24:26	RX_DPLL_MODE[2:0]	DPLL Mode selection When RX_EN is not asserted, this can change any time. This should not change when RX_EN is asserted, This bit is not used by the FPGA Interface.	R/W	0x0
27	TX_CKO_EN	Set to “1” to power-up clock for Transmit Domain. Set to “0” when port is not in used. Have to be set to “1” even when only 1 lane is in used.	R/W	1
28:29	LOS_CTL[1:0]	Enable LOS detector and/of filtering of raw LOS output 0b00 = LOS detector disabled 0b11 = Heavy filtering for generic LOS signaling Other values are reserved. This bit is not used by the FPGA Interface.	R/W	0x0
30	RX_ALIGN_EN	Enable Word Alignment This bit must be turned off during PRBS test (“Bit Error Rate Testing (BERT)”). This bit is not used by the FPGA Interface.	R/W	1
31	Reserved	N/A	R/W	1

25.9.3 RapidIO SMAC x SerDes Configuration Channel 1 Register

This register controls RapidIO port SerDes channel 1. For more information on port configuration after power down, see "Port Power Down".

Register name: SMAC{0,2,4}_CFG_CH1 Reset value: 0xA03C_E513	Register offset: 130B4, 132B4, 134B4

Bits	0	1	2	3	4	5	6	7
00:07	HALF_RATE	TX_CALC	Reserved			TX_ATTEN[2:0]		
08:15	Reserved	TX_EN[2:0]			TX_BOOST[3:0]			
16:23	RX_PLL_PWRON	RX_EN	DPLL_RESET	Reserved		RX_EQ_VAL[2:0]		
24:31	RX_DPLL_MODE[2:0]			TX_CKO_EN	LOS_CTL[1:0]		RX_ALIGN_EN	Reserved

Bits	Name	Description	Type	Reset Value
0	HALF_RATE	Baud rate control. Set to 1 if operating at 1.25Gbps; Otherwise, set to 0 for 2.5Gbps and 3.125Gbps. This value should be consistent with the other HALF_RATE settings within a SerDes. Different HALF_RATE settings within a SerDes configuration can lead to unpredictable behavior. Note: This bit setting overrides the setting of the IO_SPEED field in the "RapidIO SMAC x Digital Loopback and Clock Selection Register".	R/W	1
1	TX_CALC	A rising edge causes a recalculation of the transmitter attenuation and boost configuration. The rising edge can be generated by writing a "0" and then a "1" to the register bit.	R/W	0
2:4	Reserved	N/A Bit 3 and 4: N/A Reset value: 0x0	R/W	0x4
5:7	TX_ATTEN[2:0]	Transmit Attenuation control, provides discrete driver attenuation factors of either: {16, 14, 12, 10, 9, 8} sixteenths of full drive level. 000 = 16/16 001 = 14/16 010 = 12/16 011 = 10/16 100 = 9/16 101 = 8/16 11x = reserved	R/W	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
8	Reserved	N/A	R	0
9:11	TX_EN[2:0]	Transmitter enable 000 = Off, no clocks operating 011 = Transmitter fully enabled, all clocks operating Other values are Reserved. This bit is read only unless SMACx_CFG_GBL[BYPASS_INIT] is set to 1.	R/W	0x3
12:15	TX_BOOST[3:0]	Transmit Boost control. Programmed boost value (ratio of drive level of transition bit to non-transition bit) is: boost = -20*log(1-(tx_boost[3:0]+0.5)/32)dB, except that setting tx_boost to 0 produces 0dB of boost. This produces results up to 5.75dB in steps of ~0.37dB.	R/W	0xC
16	RX_PLL_PWRON	Power up/reset the receive PLL 0 = Rx PLL off 1 = Rx PLL on This bit is read only unless SMACx_CFG_GBL[BYPASS_INIT] is set to 1.	R/W	0x1
17	RX_EN	Enable receive clock and data outputs. rx_en can only be asserted when both rx_pll_pwron and rx_pll_state are both asserted. Set to 1 to enable Receive Data; 0 to disable Receive Data. This bit is read only unless SMACx_CFG_GBL[BYPASS_INIT] is set to 1.	R/W	0x1
18	DPLL_RESET	A rising edge resets the frequency register of the DPLL. The rising edge can be generated by writing a “0” and then a “1” to the register bit.	R/W	0x1
19:20	Reserved	N/A	R/W	0
21:23	RX_EQ_VAL[2:0]	Receive Equalization control. Internal linear equalizer boost is approximately = (rx_eq_val+1)*0.5dB For example, 0b100 = 2.5dB boost	R/W	0x5
24:26	RX_DPLL_MODE[2:0]	DPLL Mode selection When RX_EN is not asserted, this can change any time. This should not change when RX_EN is asserted,	R/W	0x0
27	TX_CKO_EN	Set to “1” to power-up clock for Transmit Domain. Set to “0” when port is not in used. Have to be set to “1” even when only 1 lane is in used.	R/W	1
28:29	LOS_CTL[1:0]	Enable LOS detector and/of filtering of raw LOS output 0b00 = LOS detector disabled 0b11 = Heavy filtering for generic LOS signaling Other values are reserved.	R/W	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
30	RX_ALIGN_EN	Enable Word Alignment This bit must be turned off during PRBS test (“Bit Error Rate Testing (BERT)”).	R/W	1
31	Reserved	N/A	R/W	1

25.9.4 RapidIO SMAC x SerDes Configuration Channel 2 Register

This register controls RapidIO port SerDes channel 2. For more information on port configuration after power-down, see "Port Power Down".

Register name: SMAC{0,2,4}_CFG_CH2 Reset value: 0xA03C_E513	Register offset: 130B8, 132B8, 134B8

Bits	0	1	2	3	4	5	6	7
00:07	HALF_RATE	TX_CALC	Reserved			TX_ATTEN[2:0]		
08:15	Reserved	TX_EN[2:0]			TX_BOOST[3:0]			
16:23	RX_PLL_PWRON	RX_EN	DPLL_RESET	Reserved		RX_EQ_VAL[2:0]		
24:31	RX_DPLL_MODE[2:0]			TX_CKO_EN	LOS_CTL[1:0]		RX_ALIGN_EN	Reserved

Bits	Name	Description	Type	Reset Value
0	HALF_RATE	Baud rate control. Set to 1 if operating at 1.25Gbps; Otherwise, set to 0 for 2.5G and 3.125Gbps. This value should be consistent with the other HALF_RATE settings within a SerDes. Different HALF_RATE settings within a SerDes configuration can lead to unpredictable behavior. Note: This bit setting overrides the setting of the IO_SPEED field in the "RapidIO SMAC x Digital Loopback and Clock Selection Register".	R/W	1
1	TX_CALC	A rising edge causes a recalculation of the transmitter attenuation and boost configuration. The rising edge can be generated by writing a "0" and then a "1" to the register bit.	R/W	0
2:4	Reserved	N/A Bit 3 and 4: N/A Reset value: 0x0	R/W	0x4
5:7	TX_ATTEN[2:0]	Transmit Attenuation control, provides discrete driver attenuation factors of either: {16, 14, 12, 10, 9, 8} sixteenths of full drive level. 000 = 16/16 001 = 14/16 010 = 12/16 011 = 10/16 100 = 9/16 101 = 8/16 11x = reserved	R/W	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
8	Reserved	N/A	R	0
9:11	TX_EN[2:0]	Transmitter enable 000 = Off, no clocks operating 011 = Transmitter fully enabled, all clocks operating Other values are Reserved. This bit is read only unless SMACx_CFG_GBL[BYPASS_INIT] is set to 1.	R/W	0x3
12:15	TX_BOOST[3:0]	Transmit Boost control. Programmed boost value (ratio of drive level of transition bit to non-transition bit) is: boost = -20*log(1-(tx_boost[3:0]+0.5)/32)dB, except that setting tx_boost to 0 produces 0dB of boost. This produces results up to 5.75dB in steps of ~0.37dB.	R/W	0xC
16	RX_PLL_PWRON	Power up/reset the receive PLL 0 = Rx PLL off 1 = Rx PLL on This bit is read only unless SMACx_CFG_GBL[BYPASS_INIT] is set to 1.	R/W	0x1
17	RX_EN	Enable receive clock and data outputs. rx_en can only be asserted when both rx_pll_pwron and rx_pll_state are both asserted. Set to 1 to enable Receive Data; 0 to disable Receive Data. This bit is read only unless SMACx_CFG_GBL[BYPASS_INIT] is set to 1.	R/W	0x1
18	DPLL_RESET	A rising edge resets the frequency register of the DPLL. The rising edge can be generated by writing a “0” and then a “1” to the register bit.	R/W	0x1
19:20	Reserved	N/A	R/W	0
21:23	RX_EQ_VAL[2:0]	Receive Equalization control. Internal linear equalizer boost is approximately = (rx_eq_val+1)*0.5dB For example, 100 = 2.5dB boost	R/W	0x5
24:26	RX_DPLL_MODE[2:0]	DPLL Mode selection When RX_EN is not asserted, this can change any time. This should not change when RX_EN is asserted,	R/W	0x0
27	TX_CKO_EN	Set to “1” to power-up clock for Transmit Domain. Set to “0” when port is not in used. Have to be set to “1” even when only 1 lane is in used.	R/W	1
28:29	LOS_CTL[1:0]	Enable LOS detector and/of filtering of raw LOS output 00 = LOS detector disabled 11 = Heavy filtering for generic LOS signaling Other values are reserved.	R/W	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
30	RX_ALIGN_EN	Enable Word Alignment This bit must be turned off during PRBS test (“Bit Error Rate Testing (BERT)”).	R/W	1
31	Reserved	N/A	R/W	1

25.9.5 RapidIO SMAC x SerDes Configuration Channel 3 Register

This register controls RapidIO port SerDes channel 3. For more information on port configuration after power-down, see "Port Power Down".

Register name: SMAC{0,2,4}_CFG_CH3 Reset value: 0xA03C_E513	Register offset: 130BC, 132BC, 134BC

Bits	0	1	2	3	4	5	6	7
00:07	HALF_RATE	TX_CALC	Reserved			TX_ATTEN[2:0]		
08:15	Reserved	TX_EN[2:0]			TX_BOOST[3:0]			
16:23	RX_PLL_PWRON	RX_EN	DPLL_RESET	Reserved		RX_EQ_VAL[2:0]		
24:31	RX_DPLL_MODE[2:0]			TX_CKO_EN	LOS_CTL[1:0]		RX_ALIGN_EN	Reserved

Bits	Name	Description	Type	Reset Value
0	HALF_RATE	Baud rate control. Set to 1 if operating at 1.25Gbps; Otherwise, set to 0 for 2.5G and 3.125Gbps. This value should be consistent with the other HALF_RATE settings within a SerDes. Different HALF_RATE settings within a SerDes configuration can lead to unpredictable behavior. Note: This bit setting overrides the setting of the IO_SPEED field in the "RapidIO SMAC x Digital Loopback and Clock Selection Register".	R/W	1
1	TX_CALC	A rising edge causes a recalculation of the transmitter attenuation and boost configuration. The rising edge can be generated by writing a "0" and then a "1" to the register bit.	R/W	0
2:4	Reserved	N/A Bit 3 and 4: N/A Reset value: 0x0	R/W	0x4
5:7	TX_ATTEN[2:0]	Transmit Attenuation control, provides discrete driver attenuation factors of either: {16, 14, 12, 10, 9, 8} sixteenths of full drive level. 000 = 16/16 001 = 14/16 010 = 12/16 011 = 10/16 100 = 9/16 101 = 8/16 11x = reserved	R/W	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
8	Reserved	N/A	R	0
9:11	TX_EN[2:0]	Transmitter enable 000 = off, no clocks operating 011 = transmitter fully enabled, all clocks operating Other values are Reserved. This bit is read only unless SMACx_CFG_GBL[BYPASS_INIT] is set to 1.	R/W	0x3
12:15	TX_BOOST[3:0]	Transmit Boost control. Programmed boost value (ratio of drive level of transition bit to non-transition bit) is: boost = -20*log(1-(tx_boost[3:0]+0.5)/32)dB, except that setting tx_boost to 0 produces 0dB of boost. This produces results up to 5.75dB in steps of ~0.37dB.	R/W	0xC
16	RX_PLL_PWRON	Power up/reset the receive PLL 0 = Rx PLL off 1 = Rx PLL on This bit is read only unless SMACx_CFG_GBL[BYPASS_INIT] is set to 1.	R/W	0x1
17	RX_EN	Enable receive clock and data outputs. rx_en can only be asserted when both rx_pll_pwron and rx_pll_state are both asserted. Set to 1 to enable Receive Data; 0 to disable Receive Data. This bit is read only unless SMACx_CFG_GBL[BYPASS_INIT] is set to 1.	R/W	0x1
18	DPLL_RESET	A rising edge resets the frequency register of the DPLL. The rising edge can be generated by writing a “0” and then a “1” to the register bit.	R/W	0x1
19:20	Reserved	N/A	R/W	0
21:23	RX_EQ_VAL[2:0]	Receive Equalization control. Internal linear equalizer boost is approximately = (rx_eq_val+1)*0.5dB For example, 0b100 = 2.5dB boost	R/W	0x5
24:26	RX_DPLL_MODE[2:0]	DPLL Mode selection When RX_EN is not asserted, this can change any time. This should not change when RX_EN is asserted,	R/W	0x0
27	TX_CKO_EN	Set to “1” to power-up clock for Transmit Domain. Set to “0” when port is not in used. Have to be set to “1” even when only 1 lane is in used.	R/W	1
28:29	LOS_CTL[1:0]	Enable LOS detector and/of filtering of raw LOS output 00 = LOS detector disabled 11 = Heavy filtering for generic LOS signaling Other values are reserved.	R/W	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
30	RX_ALIGN_EN	Enable Word Alignment This bit must be turned off during PRBS test (“Bit Error Rate Testing (BERT)”).	R/W	1
31	Reserved	N/A	R/W	1

25.9.6 RapidIO SMAC x SerDes Configuration Global Register

This register configures the SerDes of all four ports.

Register name: SMAC{0,2,4,6}_CFG_GBL Reset value: undefined	Register offset: 130C0, 132C0, 134C0, 136C0

Bits	0	1	2	3	4	5	6	7
00:07	SERDES_ RSTN	BYPASS_ INIT	Reserved	TX_LVL[4:0]				
08:15	Reserved			ACJT_LVL[4:0]				
16:23	Reserved			LOS_LVL[4:0]				
24:31	MPLL_ PWR_ON	MPLL_CK_ OFF	Reserved					

Bits	Name	Description	Type	Reset Value
0	SERDES_ RSTN	Active low reset signal to the SerDes It is used in the FPGA Interface as part of the reset of the FPGA port. This bit only functions when BYPASS_INIT is set to 1.	R/W	1
1	BYPASS_ INIT	Control bit to bypass initialization logic 0 = (default) SerDes initialization is determined by SP_IO_SPEED[1,0] 1 = Bypass initialization logic set by the SP_IO_SPEED[1,0] pins and allow direct control to SerDes. For more information on this bit, see "BYPASS_INIT Functionality".	R/W	0
2	Reserved	N/A	R	0
3:7	TX_LVL[4:0]	Fine Resolution setting of Tx signal level. Equation: Pk-Pk output level (without attenuation) = 1230 x (48 + tx_lvl/2)/63.5 mV Vdiff-pp Note: TX_LVL should be set to >= 0x1010 (which results in an output of 1Vp-p). For more information on available settings, see Table 159. Note: This bit is not used by the FPGA Interface.	R/W	0x0A
8:10	Reserved	N/A	R/W	0
11:15	ACJT_LVL[4: 0]	ACJT Receiver Comparator Level This sets the hysterisis level for AC JTAG (Table 160). For setting the correct voltage levels, see the IEEE 1149.6 specification. Note: This bit is not used by the FPGA Interface.	R/W	0x06
16:18	Reserved	N/A	R	0x0

(Continued)

Bits	Name	Description	Type	Reset Value
19:23	LOS_LVL[4:0]	Control the LOS detection threshold. Level at which LOS is asserted falls between the programmed threshold + 2mV and programmed threshold + 55mV. Programmed threshold = ((LOS_LVL+1)/(32*16))*1.21 Vpk Note: This bit is not used by the FPGA Interface.	R/W	0x0
24	MPLL_PWR_ON	0 = power down 1 = normal operation This bit is read only unless BYPASS_INIT is set to 1. It is used as part of power down of a port.	R/W	1
25	MPLL_CK_OFF	0 = turns on MPLL clock. 1 = stops the reference clock. Should transit to 0 only when MPLL_NCY, MPLL_NCY5 and MPLL_PRESCALE are all set. This bit is read only unless BYPASS_INIT in is set to 1. It is used by the FPGA Interface to control reference clock generation.	R/W	0
26:31	Reserved	N/A	R/W	Undefined

Table 159: Tx_LVL Values

Tx_lvl	Hex	Tx_lvl[0:4]	Vdiff-pp (mV)
0	0x00	5'b00000	929.8
1	0x01	5'b00001	939.4
2	0x02	5'b00010	949.1
3	0x03	5'b00011	958.8
4	0x04	5'b00100	968.5
5	0x05	5'b00101	978.2
6	0x06	5'b00110	987.9
7	0x07	5'b00111	997.6
8	0x08	5'b01000	1007.2
9	0x09	5'b01001	1016.9
10	0xA	5'b01010	1026.6
11	0xB	5'b01011	1036.3
12	0xC	5'b01100	1046.0

Table 159: Tx_LVL Values (Continued)

Tx_lvl	Hex	Tx_lvl[0:4]	Vdiff-pp (mV)
13	0xD	5'b01101	1055.7
14	0xE	5'b01110	1065.4
15	0xF	5'b01111	1075.0
16	0x10	5'b10000	1084.7
17	0x11	5'b10001	1094.4
18	0x12	5'b10010	1104.1
19	0x13	5'b10011	1113.8
20	0x14	5'b10100	1123.5
21	0x15	5'b10101	1133.1
22	0x16	5'b10110	1142.8
23	0x17	5'b10111	1152.5
24	0x18	5'b11000	1162.2
25	0x19	5'b11001	1171.9
26	0x1A	5'b11010	1181.6
27	0x1B	5'b11011	1191.3
28	0x1C	5'b11100	1200.9
29	0x1D	5'b11101	1210.6
30	0x1E	5'b11110	1220.3
31	0x1F	5'b11111	1230.0

Table 160: AC JTAG level programmed by ACJT_LVL[4:0]

ACJT_LVL[4:0]	Vmin level peak-to-peak differential (mV)	Vmin level peak single-ended (mV)
5’h02	310	77
5’h03	353	80
5’h04	395	100

Table 160: AC JTAG level programmed by ACJT_LVL[4:0] (Continued)

ACJT_LVL[4:0]	Vmin level peak-to-peak differential (mV)	Vmin level peak single-ended (mV)
5’h05	437	111
5’h06	478	121
5’h07	521	133
5’h08	563	144
5’h09	605	155
5’h0A	648	165
5’h0B	692	176
5’h14	605	100
5’h15	670	111
5’h16	735	121
5’h17	800	133
5’h18	865	144
5’h19	932	155
5’h1A	997	165
5’h1B	1065	3176

25.9.7 RapidIO SMAC x SerDes Configuration GlobalB Register

This register configures the SerDes of all RapidIO ports.

Register name: SMAC{0,2,4}_CFG_GBLB Reset value: Undefined	Register offset: 130C4, 132C4, 134C4

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved		MPLL_PRESCALE[1:0]		Reserved			
16:23	Reserved							
24:31	MPLL_NCY5[1:0]		Reserved					

Bits	Name	Description	Type	Reset Value
0:9	Reserved	N/A	R	0
10:11	MPLL_PRESCALE[1:0]	Control of MPLL's refclk prescaler. Should be set to 0b10 in Tsi620. Mapping: 10 = Divide ref_clk by 2 11 = Unused Transition only during RESET or when MPLL Is disabled. Note: This bit controls generation of the system clock for the FPGA Interface.	R/W	0x2
12:23	Reserved	N/A	R/W	0xC05
24:25	MPLL_NCY5[1:0]	Should be set to "0" MPLL multiplier is: prescale*(4*(ncy+1)+ncy5, with ncy+1>=ncy5 Transition only during RESET of SerDes of when MPLL is disabled. Note: This bit controls generation of the system clock for the FPGA Interface.	R/W	0x0
26:31	Reserved	N/A	R	Undefined

25.9.8 RapidIO SMAC 6 SerDes Configuration GlobalB Register

This register configures the FPGA Interface ports.

Register name: SMAC6_CFG_GBLB Reset value: Undefined	Register offset: 136C4

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	CLK_SEL		MPLL_PRESCALE[1:0]		Reserved			
16:23	Reserved							
24:31	MPLL_NCY5[1:0]		Reserved					

Bits	Name	Description	Type	Reset Value
0:7	Reserved	N/A	R	0
8:9	CLK_SEL	Clock selection value 0b00 - 125 MHz Reference Clock Frequency 0b01 - 156.25 MHz Reference Clock Frequency 0b10, 0b11 - Reserved	R/W	Undefined
10:11	MPLL_PRESCALE[1:0]	Control of MPLL's refclk prescaler. Should be set to 0b10 in Tsi620. Mapping: 10 = Divide ref_clk by 2 11 = Unused Transition only during RESET or when MPLL Is disabled. Note: This bit controls generation of the system clock for the FPGA Interface.	R/W	0x2
12:23	Reserved	N/A	R/W	0xC05
24:25	MPLL_NCY5[1:0]	Should be set to "0" MPLL multiplier is: prescale*(4*(ncy+1)+ncy5, with ncy+1>=ncy5 Transition only during RESET of SerDes of when MPLL is disabled. Note: This bit controls generation of the system clock for the FPGA Interface.	R/W	0x0
26:31	Reserved	N/A	R	Undefined

25.9.9 RapidIO SMAC x Digital Loopback and Clock Selection Register

This register consists of controls for Dead Link Timer and Digital Equipment Loopback (TX -> RX) as well as Clock Selection on a per port basis.

Register name: SMAC{0,2,4,6,8}_DLOOP_CLK_SEL **Reset value: Undefined**	**Register offset:130C8, 132C8, 134C8, 136C8,138C8**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	DLT_EN	DLT_THRESH						
08:15	DLT_THRESH							
16:23	LINE_LB[3:0]				Reserved	MAC_ MODE	DLB_ ODD_EN	DLB_ EVEN_EN
24:31	SWAP_TX	SWAP_RX	SOFT_RST _X1	SOFT_RST _X4	PWDN_X1	PWDN_X4	IO_SPEED	

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	DLT_EN	Dead Link Timer Enable 0 = Disabled (Default) 1 = When enabled, this timer determines when a link is powered up and enabled, but dead (that is, there is no link partner re7sponding). When a link is declared dead, the transmitting port on the Tsi620 removes all packets from its transmit queue and ensure that all new packets sent to port are dropped rather than placed in the transmit queue. This feature affects both RapidIO ports sharing the SMAC. This feature is not limited to lane 0 of the SerDes despite the register it occurs to be located in. The dead link timer is operational for Port 6. If the link cannot initialize with the link partner within the DLT_THRESH time period, the link should be handled as dead. The dead link timer is not operational for Port 8. This field controls no functionality for Port 8.	R/W	0
1:15	DLT_ THRESH	Dead Link Timer Threshold Each time a silence is detected on a link, the counter is reloaded from this register and starts to count down. When the count reaches 0, the link is declared dead, which means that all packets are flushed from the transmit queue and no new packets are admitted to the queue until the link comes up. The duration of the dead link timer is computed as: 2^^13 * DLT_THRESH * P_CLK period. If DLT_THRESH = 0, even when DLT_EN = 1, the counter is still disabled. The dead link timer is not operational for Port 8. This field controls no functionality for Port 8.	R/W	0x7FFF

(Continued)

Bits	Name	Description	Type	Reset Value
16:19	LINE_LB	Line Loopback 0 = Disabled 1 = Enabled Line Loopback LINE_LB[3..0] = lane[D, C, B, A] (see Figure 9). Note that for Port 6, there is an additional line loopback mode (see the SWAP_TX field in this register). This field controls no functionality for Port 8.	R/W	0
20	Reserved	N/A	R/W	0
21	MAC_MODE	0 = MAC supports a single 1x/4x port. 1 = MAC supports 2 independent 1x ports. After the Tsi620 is reset, this field indicates the configuration of the SPx_MODE_SEL. Writing to this register overrides the pin settings of SPx_MODE_SEL. This field controls no functionality for Port 8. The Port 6 reset value of this field must not be changed.	R/W	0
22	DLB_ODD_EN	Digital Equipment Loopback Mode Odd-numbered Port Digital equipment loopback mode connects Tx data flow to Rx data flow before the 8B10B encoder/decoder. 0 = Normal operation 1 = Loopback enabled for the odd numbered port served by this MAC. Note: The loopback path does not include the 8b/10B encoder/decoder. (see "Port Loopback Testing"). This field controls no functionality for Port 6 or Port 8.	R/W	0
23	DLB_EVEN_EN	Digital Equipment Loopback Mode Even-numbered Port Digital equipment loopback mode connects Tx data flow to Rx data flow before the 8B10b encoder/decoder. 0 = Normal operation 1 = Loopback enabled for the even-numbered port served by this MAC. The loopback path does not include the 8b/10B encoder/decoder (see "Port Loopback Testing"). This field controls no functionality for Port 8.	R/W	0
24	SWAP_TX	For ports 0-5, this bit allows software control for transmitter lane swap functionality for this MAC. This field initially indicates the sampled value of the SP_TX_SWAP pin. 0 = A, B, C, D 1 = D, C, B, A For more information, see "Lane Swapping". For Port 6, this bit controls line loopback closer to the pins. This field controls no functionality for Port 6 or Port 8.	R/W	Undefined

(Continued)

Bits	Name	Description	Type	Reset Value
25	SWAP_RX	Software control for receiver lane swap functionality for this MAC. This field initially indicates the sampled value of the SP_RX_SWAP pin. 0 = A, B, C, D 1 = D, C, B, A For more information, see "Lane Swapping". This field controls no functionality for Port 6 or 8.	R/W	Undefined
26	SOFT_RST_X1	Reset Control Odd-numbered Port Reset control for the odd-numbered port using this MAC. 0 = Normal mode of operation 1 = Odd-numbered port held in reset Note: This bit resets the core only. This field controls no functionality for Port 6 or 8. When SOFT_RST_X1 is 1, then the following per-port registers are reset. These registers are not accessible when the port is in reset: • "RapidIO Physical Layer Registers" • "RapidIO Error Management Extension Registers" • "IDT-Specific RapidIO Registers" • "IDT-Specific Performance Registers" • "Per Port Copies of Global Registers" The following per-port registers are not reset. These registers are still accessible when the port is in reset: • "Serial Port Electrical Layer Registers" • "Switch ISF Registers" • "Multicast Registers" • "SerDes Per Lane Register" •	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
27	SOFT_RST_X4	Reset control for the even-numbered port using this MAC, and for the SerDes. 0 = Normal mode of operation 1 = Even-numbered port and SerDes held in reset. The odd-numbered port, if being used, is affected since the SerDes is used by both the even- and odd-numbered ports. This bit resets the SerDes, core, and PRBS. It does not reset the registers. This field controls no functionality for Port 8.When SOFT_RST_X4 is 1, then the following per-port registers are reset. These registers are not accessible when the port is in reset: • "RapidIO Physical Layer Registers" • "RapidIO Error Management Extension Registers" • "IDT-Specific RapidIO Registers" • "IDT-Specific Performance Registers" • "FPGA PRBS/BERT Control Registers" • "Per Port Copies of Global Registers" The following per-port registers are not reset. These registers are still accessible when the port is in reset: • "Serial Port Electrical Layer Registers" • "Switch ISF Registers" • "Multicast Registers" • "SerDes Per Lane Register" •	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
28	PWDN_X1	Power-down control for the odd-numbered port using this MAC. This field initially indicates the sampled value of the SPx_PWDN pin, where “x” is 1, 3, and 5. Writing to this register overrides the configuration provided by the pin. 0 = Normal mode of operation 1 = Port powered down Note: For Port 6, this bit is always 0. This bit does not exist for Port 8. When PWDN_X1 is 1, then the following per-port registers are reset. These registers are not accessible when the port is in reset: • “RapidIO Physical Layer Registers” • “RapidIO Error Management Extension Registers” • “IDT-Specific RapidIO Registers” • “IDT-Specific Performance Registers” • “Per Port Copies of Global Registers” The following per-port registers are not reset. These registers are still accessible when the port is in reset: • “Serial Port Electrical Layer Registers” • “Switch ISF Registers” • “Multicast Registers” • “SerDes Per Lane Register” •	R/W	Undefined

(Continued)

Bits	Name	Description	Type	Reset Value
29	PWDN_X4	Power-down control for both the even-numbered and odd-numbered ports using this MAC. This field initially indicates the sampled value of the SPx_PWDN pin, where “x” is 0, 2, 4, and 6. Writing to this register overrides the configuration provided by both the even and odd numbered pins connected to this MAC. 0 = Normal mode of operation 1 = Both the even and odd ports are powered down For more information on how this bit works within Internal Switch Port (Port 8), see “Link Maintenance Functions” and “Bridge Shutdown”. When PWDN_X4 is 1, then the following per-port registers are reset. These registers are not accessible when the port is in reset: • “RapidIO Physical Layer Registers” • “RapidIO Error Management Extension Registers” • “IDT-Specific RapidIO Registers” • “IDT-Specific Performance Registers” • “FPGA PRBS/BERT Control Registers” • “Per Port Copies of Global Registers” The following per-port registers are not reset. These registers are still accessible when the port is in reset: • “Serial Port Electrical Layer Registers” • “Switch ISF Registers” • “Multicast Registers” • “SerDes Per Lane Register” •	R/W	Undefined
30:31	IO_SPEED	This field determines the lane speed for the RapidIO port: 00 = 1.25 Gbps 01 = 2.5 Gbps 10 = 3.125 Gbps 11 = Reserved (Indicates the value on SP_IO_SPEED after reset.) Writing to this register overrides a power-up value of SP_IO_SPEED speed selection. Note: This field controls no functionality for Port 8.	R/W	Undefined

25.9.10 RapidIO SMAC SerDes Output Pins Register

Register name: SMAC{0,2,4,6}_SERDES_OUTPUT_PINS Reset value: Undefined	Register offset: 130CC, 132CC, 134CC, 136CC

Bits	0	1	2	3	4	5	6	7
00:07	RX_COMMA_DET							
08:15	TX_DONE				TX_RXPRES			
16:23	RX_PLL_STATE				RX_VALID			
24:31	LOS				OP_DONE	POWER_GOOD	Reserved	

Bits	Name	Description	Type	Reset Value
0:7	RX_COMMA_DET	This field contains random information. Note: This field is always 0 for port 6 (the FPGA Interface).	R	Undefined
8:11	TX_DONE	Transmitter operation is complete. Transitions (either direction) when the transmitter has completed a requested power state transition or operation requested through the tx_ena[] pins when a tx_calc or a tx_clk_align operation is complete.Transitions synchronously to tx_ck, but may be sampled asynchronously if necessary. Note: This field is always 0 for port 6 (the FPGA Interface).	R	Undefined
12:15	TX_RXPRES	Result of receiver detection. Latched internally before the assertion of tx_done. Transitions asynchronously to tx_ck. (non-functional) Note: This field is always 0 for port 6 (the FPGA Interface).	R	Undefined
16:19	RX_PLL_STATE	Current state of rx_pll. This l matches rx_pll_pwron once the PLL reaches the requested state. rx_en can only be asserted when both rx_pll_pwron and rx_pll_state are both asserted. Note: This field is always 0 for port 6 (the FPGA Interface).	R	Undefined
20:23	RX_VALID	Receive DPLL is bit-locked Transitions synchronous to rx_ck Note: This field is always 0 for port 6 (the FPGA Interface).	R	Undefined
24:27	LOS	Loss of signal output. Transitions asynchronous to rx_ck, but synchronous to refclk. Note: This field is always 0 for port 6 (the FPGA Interface).	R	Undefined

(Continued)

Bits	Name	Description	Type	Reset Value
28	OP_DONE	Signals that the requested operation is complete by transitioning in either direction. Supported operations: • mpll_pwron assertion: transitions after cko_word is operating, and at the correct frequency • mpll_pwron de-assertion: asynchronously transitions after cko_word is parked • rtune_do_tune assertion: transitions when resistor tuning is complete • reset de-assertion: resistor tuning and MPLL power up complete	R	Undefined
29	POWER_GOOD	Internal POR result.Indicates that both 1.0/1.2 and 2.5/3.3 power supplies are reasonable, and end of internally generated RESET.	R	Undefined
30:31	Reserved	N/A	R	Undefined

25.10 FPGA PRBS/BERT Control Registers

PRBS functionality is required to test the FPGA Interface. In SMAC, PRBS functionality is part of the SerDes.

25.10.1 RapidIO SMAC 6 PRBS Control Register

For more information on the usages of this register, see “FPGA Interface Bit Error Rate Testing (BERT)”.

Register name:SMAC6_PRBS_CTRL **Reset value: 0x0000_0000**	**Register offset: 0x137C8**

Bits	0	1	2	3	4	5	6	7
00:07	Reserved	PATTERN_SEL3			START_ PRBS3	SYNC_ PRBS3	STOP_ PRBS3_ CTR	CLR_ PRBS3_ CTR
08:15	Reserved	PATTERN_SEL2			START_ PRBS2	SYNC_ PRBS2	STOP_ PRBS2_ CTR	CLR_ PRBS2_ CTR
16:23	Reserved	PATTERN_SEL1			START_ PRBS1	SYNC_ PRBS1	STOP_ PRBS1_ CTR	CLR_ PRBS1_ CTR
24:31	Reserved	PATTERN_SEL0			START_ PRBS0	SYNC_ PRBS0	STOP_ PRBS0_ CTR	CLR_ PRBS0_ CTR

Bits	Name	Description	Type	Reset Value
0	Reserved	N/A	R	0
1:3	PATTERN_ SEL3	Channel 3 Transmitter Pattern Select 000 = PRBS not active 001 = PRBS pattern 010 = Reserved 011 = Reserved 100 = Fixed character 101 = Reserved 110 = Reserved 111 = Reserved	R/W	0
4	START_ PRBS3	Channel 3 Start PRBS Writing 1 asserts Start PRBS to the PRBS generator and releases it from the initial state. Writing 0 forces the PRBS generator to the initial state.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
5	SYNC_ PRBS3	Channel 3 SYNC PRBS Writing 1 asserts SYNC_PRBS to the PRBS Checker for 4 cycles. The bit automatically returns to 0 and therefore appears to be a write-only field.	R/W1S	0
6	STOP_ PRBS3_CTR	Channel 3 STOP PRBS Writing 1 stops PRBS Error and Code-Group counters. Writing 0 enables counters after assertion of SYNC_PRBS. Counters start to accept events after writing 1 to the SYNC_PRBS.	R/W	0
7	CLR_PRBS3_ CTR	Channel 3 CLEAR PRBS Writing 1 clears PRBS Error and Code-Group counters. This bit automatically returns to 0.	R/W1S	0
8	Reserved	N/A	R	0
9:11	PATTERN_ SEL2	Channel 2 Transmitter Pattern Select See description for bits 1:3 — PATTERN_SEL3.	R/W	0
12	START_ PRBS2	Channel 2 Start PRBS See description for bit 4 — START_PRBS3.	R/W	0
13	SYNC_ PRBS2	Channel 2 SYNC PRBS See description for bit 5 — SYNC_PRBS3.	R/W1S	0
14	STOP_ PRBS2_CTR	Channel 2 STOP PRBS See description for bit 6 — STOP_PRBS3_CTR.	R/W	0
15	CLR_PRBS2_ CTR	Channel 2 CLEAR PRBS See description for bit 7 — CLR_PRBS3_CTR.	R/W1S	0
16	Reserved	N/A	R	0
17:19	PATTERN_ SEL1	Channel 1 Transmitter Pattern Select See description for bits 1:3 — PATTERN_SEL3.	R/W	0
20	START_ PRBS1	Channel 1 Start PRBS See description for bit 4 — START_PRBS3.	R/W	0
21	SYNC_ PRBS1	Channel 1 SYNC PRBS See description for bit 5 — SYNC_PRBS3.	R/W1S	0
22	STOP_ PRBS1_CTR	Channel 1 STOP PRBS See description for bit 6 — STOP_PRBS3_CTR.	R/W	0
23	CLR_PRBS1_ CTR	Channel 1 CLEAR PRBS See description for bit 7 — CLR_PRBS3_CTR.	R/W1S	0
24	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
25:27	PATTERN_ SEL0	Channel 0 Transmitter Pattern Select See description for bits 1:3 — PATTERN_SEL3.	R/W	0
28	START_ PRBS0	Channel 0 Start PRBS See description for bit 4 — START_PRBS3.	R/W	0
29	SYNC_ PRBS0	Channel 0 SYNC PRBS See description for bit 5 — SYNC_PRBS3.	R/W1S	0
30	STOP_ PRBS0_CTR	Channel 0 STOP PRBS See description for bit 6 — STOP_PRBS3_CTR.	R/W	0
31	CLR_PRBS0_ CTR	Channel 0 CLEAR PRBS See description for bit 7 — CLR_PRBS3_CTR.	R/W1S	0

25.10.2 RapidIO SMAC 6 PRBS Channel 0 Counter 0 Register

The PRBS counter 0 and 1 calculate Bit Error Rate (BERT) of the link. The PRBS generator must be stopped by writing 1 to STOP_PRBS0_CTR in the "RapidIO SMAC 6 PRBS Control Register" before reading this register.

Register name:SMAC6_CH0_PRBS_CTR0 **Reset value: 0x0000_0000**	**Register offset: 0x137CC**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	PRBS_CG_CTR[47:40]							
08:15	PRBS_CG_CTR[39:32]							
16:23	PRBS_ERR_CTR[15:8]							
24:31	PRBS_ERR_CTR[7:0]							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:15	PRBS_CG_CTR	Most significant PRBS_ERR_CTR[47:32] bits of 48-bit code group counter. This counter is cleared by writing 1 to CLR_PRBS_CTR in the "RapidIO SMAC 6 PRBS Control Register".	R	0x0000
16:31	PRBS_ERR_CTR	Error counter for PRBS checker. This counter is cleared by writing 1 to CLR_PRBS_CTR in the "RapidIO SMAC 6 PRBS Control Register".	R	0x0000

25.10.3 RapidIO SMAC 6 PRBS Channel 0 Counter 1 Register

The PRBS counter 0 and 1 calculate BERT. The PRBS generator must be stopped by writing 1 to STOP_PRBS0_CTR in the "RapidIO SMAC 6 PRBS Control Register" before reading this register.

Register name:SMAC6_CH0_PRBS_CTR1 **Reset value: 0x0000_0000**	**Register offset: 0x137D0**

Bits	0	1	2	3	4	5	6	7
00:07	PRBS_CG_CTR[31:24]							
08:15	PRBS_CG_CTR[23:16]							
16:23	PRBS_CG_CTR[15:08]							
24:31	PRBS_CG_CTR[07:00]							

Bits	Name	Description	Type	Reset Value
0:31	PRBS_CG_CTR[31:0]	Least significant PRBS_CG_CTR[31:0] bits of 48-bit code group counter This counter is cleared by writing 1 to CLR_PRBS_CTR in the "RapidIO SMAC 6 PRBS Control Register".	R	0x0000_0000

25.10.4 RapidIO SMAC 6 PRBS Channel 1 Counter 0 Register

The PRBS counter 0 and 1 calculate BERT. The PRBS generator must be stopped by writing 1 to STOP_PRBS1_CTR in the "RapidIO SMAC 6 PRBS Control Register" before reading this register.

Register name:SMAC6_CH1_PRBS_CTR0 **Reset value: 0x0000_0000**	**Register offset: 0x137D4**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	PRBS_CG_CTR[47:40]							
08:15	PRBS_CG_CTR[39:32]							
16:23	PRBS_ERR_CTR[15:8]							
24:31	PRBS_ERR_CTR[7:0]							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:15	PRBS_CG_CTR	Most significant PRBS_ERR_CTR[47:32] bits of 48-bit code group counter. This counter is cleared by writing 1 to CLR_PRBS_CTR in the "RapidIO SMAC 6 PRBS Control Register".	R	0x0000
16:31	PRBS_ERR_CTR	Error counter for PRBS checker. This counter is cleared by writing 1 to CLR_PRBS_CTR in the "RapidIO SMAC 6 PRBS Control Register".	R	0x0000

25.10.5 RapidIO SMAC 6 PRBS Channel 1 Counter 1 Register

The PRBS counter 0 and 1 calculate BER. The PRBS generator must be stopped by writing 1 to STOP_PRBS1_CTR in the "RapidIO SMAC 6 PRBS Control Register" before reading this register.

Register name: SMAC6_CH1_PRBS_CTR1 **Reset value: 0x0000_0000**	**Register offset: 0x137D8**

Bits	0	1	2	3	4	5	6	7
00:07	PRBS_CG_CTR[31:24]							
08:15	PRBS_CG_CTR[23:16]							
16:23	PRBS_CG_CTR[15:08]							
24:31	PRBS_CG_CTR[07:00]							

Bits	Name	Description	Type	Reset Value
0:31	PRBS_CG_CTR[31:0]	Least significant PRBS_CG_CTR[31:0] bits of 48-bit code group counter This counter is cleared by writing 1 to CLR_PRBS_CTR in the "RapidIO SMAC 6 PRBS Control Register".	R	0x0000_0000

25.10.6 RapidIO SMAC 6 PRBS Channel 2 Counter 0 Register

The PRBS counter 0 and 1 calculate BERT. The PRBS generator must be stopped by writing 1 to STOP_PRBS2_CTR in the "RapidIO SMAC 6 PRBS Control Register" before reading this register.

Register name:SMAC6_CH2_PRBS_CTR0 **Reset value: 0x0000_0000**	**Register offset: 0x137DC**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	PRBS_CG_CTR[47:40]							
08:15	PRBS_CG_CTR[39:32]							
16:23	PRBS_ERR_CTR[15:8]							
24:31	PRBS_ERR_CTR[7:0]							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:15	PRBS_CG_CTR	Most significant PRBS_ERR_CTR[47:32] bits of 48-bit code group counter. This counter is cleared by writing 1 to CLR_PRBS_CTR in the "RapidIO SMAC 6 PRBS Control Register".	R	0x0000
16:31	PRBS_ERR_CTR	Error counter for PRBS checker. This counter is cleared by writing 1 to CLR_PRBS_CTR in the "RapidIO SMAC 6 PRBS Control Register".	R	0x0000

25.10.7 RapidIO SMAC 6 PRBS Channel 2 Counter 1 Register

The PRBS counter 0 and 1 calculate BERT. The PRBS generator must be stopped by writing 1 to STOP_PRBS2_CTR in the "RapidIO SMAC 6 PRBS Control Register" before reading this register.

Register name:SMAC6_CH2_PRBS_CTR1 **Reset value: 0x0000_0000**	**Register offset: 0x137E0**

Bits	0	1	2	3	4	5	6	7
00:07	PRBS_CG_CTR[31:24]							
08:15	PRBS_CG_CTR[23:16]							
16:23	PRBS_CG_CTR[15:08]							
24:31	PRBS_CG_CTR[07:00]							

Bits	Name	Description	Type	Reset Value
0:31	PRBS_CG_CTR[31:0]	Least significant PRBS_CG_CTR[31:0] bits of 48-bit code group counter This counter is cleared by writing 1 to CLR_PRBS_CTR in the "RapidIO SMAC 6 PRBS Control Register".	R	0x0000_0000

25.10.8 RapidIO SMAC 6 PRBS Channel 3 Counter 0 Register

The PRBS counter 0 and 1 calculate BERT. The PRBS generator must be stopped by writing 1 to STOP_PRBS3_CTR in the "RapidIO SMAC 6 PRBS Control Register" before reading this register.

Register name:SMAC6_CH3_PRBS_CTR0 **Reset value: 0x0000_0000**	**Register offset: 0x137E4**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	PRBS_CG_CTR[47:40]							
08:15	PRBS_CG_CTR[39:32]							
16:23	PRBS_ERR_CTR[15:8]							
24:31	PRBS_ERR_CTR[7:0]							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:15	PRBS_CG_CTR	Most significant PRBS_ERR_CTR[47:32] bits of 48-bit code group counter. This counter is cleared by writing 1 to CLR_PRBS_CTR in the "RapidIO SMAC 6 PRBS Control Register".	R	0x0000
16:31	PRBS_ERR_CTR	Error counter for PRBS checker. This counter is cleared by writing 1 to CLR_PRBS_CTR in the "RapidIO SMAC 6 PRBS Control Register".	R	0x0000

25.10.9 RapidIO SMAC 6 PRBS Channel 3 Counter 1 Register

The PRBS counter 0 and 1 calculate BERT. The PRBS generator must be stopped by writing 1 to STOP_PRBS3_CTR in the "RapidIO SMAC 6 PRBS Control Register" before reading this register.

Register name: SMAC6_CH3_PRBS_CTR1 Reset value: 0x0000_0000	Register offset: 0x137E8

Bits	0	1	2	3	4	5	6	7
00:07	PRBS_CG_CTR[31:24]							
08:15	PRBS_CG_CTR[23:16]							
16:23	PRBS_CG_CTR[15:08]							
24:31	PRBS_CG_CTR[07:00]							

Bits	Name	Description	Type	Reset Value
0:31	PRBS_CG_CTR[31:0]	Least significant PRBS_CG_CTR[31:0] bits of 48-bit code group counter This counter is cleared by writing 1 to CLR_PRBS_CTR in the "RapidIO SMAC 6 PRBS Control Register".	R	0x0000_0000

25.10.10 RapidIO SMAC 6 BERT Data Register for Channel 0 Register

This register is used during a BERT test to define a repeating fixed pattern sent by the transmitter.

Register name:SMAC6_CH0_BERT_DATA Reset value: 0x0000_0000	Register offset: 0x137EC

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							PAT1
08:15	PAT1							
16:23	Reserved							PAT0
24:31	PAT0							

Bits	Name	Description	Type	Reset Value
0:6	Reserved	N/A	R	0
7:15	PAT1	Pattern 1 See text for PAT0.	R/W	0
16:22	Reserved	N/A	R	0
23:31	PAT0	Pattern 0 During a BERT test caused by setting the PATTERN_SEL field in the "RapidIO SMAC 6 PRBS Control Register" to 0b100, the affected SerDes lane sends the entire PAT0 value starting with bit 22, followed by the entire PAT1 value starting at bit 6. This represents two bytes to be transmitted, with the least significant 8 bits representing a data value and the most significant bit indicating that a K character (1) or data character (0) is sent. The pattern repeats until the BERT is disabled through the "RapidIO SMAC 6 PRBS Control Register". For more information on the usage model of this register, see "Bit Error Rate Testing (BERT)".	R/W	0

25.10.11 RapidIO SMAC 6 BERT Data Register for Channel 1 Register

This register is used during a BERT test to define a repeating fixed pattern sent by the transmitter.

Register name: SMAC6_CH1_BERT_DATA Reset value: 0x0000_0000	Register offset: 0x137F0

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							PAT1
08:15	PAT1							
16:23	Reserved							PAT0
24:31	PAT0							

Bits	Name	Description	Type	Reset Value
0:6	Reserved	N/A	R	0
7:15	PAT1	Pattern 1 See text for PAT0.	R/W	0
16:22	Reserved	N/A	R	0
23:31	PAT0	Pattern 0 During a BERT test caused by setting the PATTERN_SEL field in the "RapidIO SMAC 6 PRBS Control Register" to 0b100, the affected SerDes lane sends the entire PAT0 value starting with bit 22, followed by the entire PAT1 value starting at bit 6. This represents two bytes to be transmitted, with the least significant 8 bits representing a data value and the most significant bit indicating that a K character (1) or data character (0) is sent. The pattern repeats until the BERT is disabled through the "RapidIO SMAC 6 PRBS Control Register". For more information on the usage model of this register, see "Bit Error Rate Testing (BERT)".	R/W	0

25.10.12 RapidIO SMAC 6 BERT Data Register for Channel 2 Register

This register is used during a BERT test to define a repeating fixed pattern sent by the transmitter.

Register name:SMAC6_CH2_BERT_DATA Reset value: 0x0000_0000	Register offset: 0x137F4

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							PAT1
08:15	PAT1							
16:23	Reserved							PAT0
24:31	PAT0							

Bits	Name	Description	Type	Reset Value
0:6	Reserved	N/A	R	0
7:15	PAT1	Pattern 1 See text for PAT0.	R/W	0
16:22	Reserved	N/A	R	0
23:31	PAT0	Pattern 0 During a BERT test caused by setting the PATTERN_SEL field in the "RapidIO SMAC 6 PRBS Control Register" to 0b100, the affected SerDes lane sends the entire PAT0 value starting with bit 22, followed by the entire PAT1 value starting at bit 6. This represents two bytes to be transmitted, with the least significant 8 bits representing a data value and the most significant bit indicating that a K character (1) or data character (0) is sent. The pattern repeats until the BERT is disabled through the "RapidIO SMAC 6 PRBS Control Register". For more information on the usage model of this register, see "Bit Error Rate Testing (BERT)".	R/W	0

25.10.13 RapidIO SMAC 6 BERT Data Register for Channel 3 Register

This register is used during a BERT test to define a repeating fixed pattern sent by the transmitter.

Register name:SMAC6_CH3_BERT_DATA Reset value: 0x0000_0000	Register offset: 0x137F8

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							PAT1
08:15	PAT1							
16:23	Reserved							PAT0
24:31	PAT0							

Bits	Name	Description	Type	Reset Value
0:6	Reserved	N/A	R	0
7:15	PAT1	Pattern 1 See text for PAT0.	R/W	0
16:22	Reserved	N/A	R	0
23:31	PAT0	Pattern 0 During a BERT test caused by setting the PATTERN_SEL field in the "RapidIO SMAC 6 PRBS Control Register" to 0b100, the affected SerDes lane sends the entire PAT0 value starting with bit 22, followed by the entire PAT1 value starting at bit 6. This represents two bytes to be transmitted, with the least significant 8 bits representing a data value and the most significant bit indicating that a K character (1) or data character (0) is sent. The pattern repeats until the BERT is disabled through the "RapidIO SMAC 6 PRBS Control Register". For more information on the usage model of this register, see "Bit Error Rate Testing (BERT)".	R/W	0

25.11 Per Port Copies of Global Registers

Due to the design of the SMAC and the FPGA Interface, some registers which, according to the RapidIO specification, affect the entire device, have local copies in each port. These registers allow read write access to the per port copies of these global registers.

These registers are used to check/correct register values after a port is powered down and back up. They are not meant to be the standard method for programming these registers within the Tsi620.

25.11.1 RapidIO Serial Port x Multicast Write ID 0 Register

This register is the per port copy of the "RapidIO Multicast Write ID x Register", where 'x' is 0. It contains the Multicast ID for which the associated multicast mask registers are applicable. The Tsi620 Switch supports eight multicast groups, therefore, the Multicast ID registers for each multicast group must contain unique values. These registers are used in every RapidIO port.

Register name: SP{0,1,2,3,4,5,6,8}_RIO_MC_ID0 **Reset value: 0x0000_0000**	**Register offset: 16000, 16100, 16200, 16300, 16400, 16500, 16600, 16800**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	MC_EN	LARGE_ SYS	Reserved					
8:15	Reserved							
16:23	MC_ID[0:7]							
24:31	MC_ID[8:15]							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	MC_EN	Multicast can be disabled by setting this bit. 0 = Multicast disabled 1 = Multicast enabled	R/W	0
1	LARGE_SY S	This field defines MC_ID in the Large or Small system. The MC_ID of Small system is not a subset of MC_ID of Large system, but both systems can co-exist together. 1 = Large system 0 = Small system	R/W	0
2:15	Reserved	N/A	R	0
16:31	MC_ID	This field defines the multicast ID for which the associated multicast mask is activated for this extended features block.	R/W	0x0000

When using this register, multiple identical entries must not exist since an addition of an association will not delete the existing one.

25.11.2 RapidIO Serial Port x Multicast Write ID 1 Register

This register is the per port copy of the "RapidIO Multicast Write ID x Register", where 'x' is 1. It contains the Multicast ID for which the associated multicast mask registers are applicable. The Tsi620 Switch supports eight multicast groups, therefore, the Multicast ID registers for each multicast group must contain unique values. These registers are used in every RapidIO port.

Register name: SP{0,1,2,3,4,5,6,8}_RIO_MC_ID1 **Reset value: 0x0000_0000**	**Register offset: 16004, 16104, 16204, 16304, 16404, 16504, 16604, 16804**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	MC_EN	LARGE_ SYS	Reserved					
8:15	Reserved							
16:23	MC_ID[0:7]							
24:31	MC_ID[8:15]							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	MC_EN	Multicast can be disabled by setting this bit. 0 = Multicast disabled 1 = Multicast enabled	R/W	0
1	LARGE_SY S	This field defines MC_ID in the Large or Small system. The MC_ID of Small system is not a subset of MC_ID of Large system, but both systems can co-exist together. 1 = Large system 0 = Small system	R/W	0
2:15	Reserved	N/A	R	0
16:31	MC_ID	This field defines the multicast ID for which the associated multicast mask is activated for this extended features block.	R/W	0x0000

When using this register, multiple identical entries must not exist since an addition of an association will not delete the existing one.

25.11.3 RapidIO Serial Port x Multicast Write ID 2 Register

This register is the per port copy of the "RapidIO Multicast Write ID x Register", where 'x' is 2. It contains the Multicast ID for which the associated multicast mask registers are applicable. The Tsi620 Switch supports eight multicast groups, therefore, the Multicast ID registers for each multicast group must contain unique values. These registers are used in every RapidIO port.

Register name: SP{0,1,2,3,4,5,6,8}_RIO_MC_ID2 **Reset value: 0x0000_0000**	**Register offset: 16008, 16108, 16208, 16308, 16408, 16508, 16608, 16808**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	MC_EN	LARGE_ SYS	Reserved					
8:15	Reserved							
16:23	MC_ID[0:7]							
24:31	MC_ID[8:15]							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	MC_EN	Multicast can be disabled by setting this bit. 0 = Multicast disabled 1 = Multicast enabled	R/W	0
1	LARGE_SY S	This field defines MC_ID in the Large or Small system. The MC_ID of Small system is not a subset of MC_ID of Large system, but both systems can co-exist together. 1 = Large system 0 = Small system	R/W	0
2:15	Reserved	N/A	R	0
16:31	MC_ID	This field defines the multicast ID for which the associated multicast mask is activated for this extended features block.	R/W	0x0000

When using this register, multiple identical entries must not exist since an addition of an association will not delete the existing one.

25.11.4 RapidIO Serial Port x Multicast Write ID 3 Register

This register is the per port copy of the "RapidIO Multicast Write ID x Register", where 'x' is 3. It contains the Multicast ID for which the associated multicast mask registers are applicable. The Tsi620 Switch supports eight multicast groups, therefore, the Multicast ID registers for each multicast group must contain unique values. These registers are used in every RapidIO port.

Register name: SP{0,1,2,3,4,5,6,8}_RIO_MC_ID3 **Reset value: 0x0000_0000**	**Register offset: 1600C, 1610C, 1620C, 1630C, 1640C, 1650C, 1660C, 1680C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	MC_EN	LARGE_ SYS	Reserved					
8:15	Reserved							
16:23	MC_ID[0:7]							
24:31	MC_ID[8:15]							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	MC_EN	Multicast can be disabled by setting this bit. 0 = Multicast disabled 1 = Multicast enabled	R/W	0
1	LARGE_SY S	This field defines MC_ID in the Large or Small system. The MC_ID of Small system is not a subset of MC_ID of Large system, but both systems can co-exist together. 1 = Large system 0 = Small system	R/W	0
2:15	Reserved	N/A	R	0
16:31	MC_ID	This field defines the multicast ID for which the associated multicast mask is activated for this extended features block.	R/W	0x0000

When using this register, multiple identical entries must not exist since an addition of an association will not delete the existing one.

25.11.5 RapidIO Serial Port x Multicast Write ID 4 Register

This register is the per port copy of the "RapidIO Multicast Write ID x Register", where 'x' is 4. It contains the Multicast ID for which the associated multicast mask registers are applicable. The Tsi620 Switch supports eight multicast groups, therefore, the Multicast ID registers for each multicast group must contain unique values. These registers are used in every RapidIO port.

Register name: SP{0,1,2,3,4,5,6,8}_RIO_MC_ID4 **Reset value: 0x0000_0000**	**Register offset: 16010, 16110, 16210, 16310, 16410, 16510, 16610, 16810**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	MC_EN	LARGE_ SYS	Reserved					
8:15	Reserved							
16:23	MC_ID[0:7]							
24:31	MC_ID[8:15]							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	MC_EN	Multicast can be disabled by setting this bit. 0 = Multicast disabled 1 = Multicast enabled	R/W	0
1	LARGE_SY S	This field defines MC_ID in the Large or Small system. The MC_ID of Small system is not a subset of MC_ID of Large system, but both systems can co-exist together. 1 = Large system 0 = Small system	R/W	0
2:15	Reserved	N/A	R	0
16:31	MC_ID	This field defines the multicast ID for which the associated multicast mask is activated for this extended features block.	R/W	0x0000

When using this register, multiple identical entries must not exist since an addition of an association will not delete the existing one.

25.11.6 RapidIO Serial Port x Multicast Write ID 5 Register

This register is the per port copy of the "RapidIO Multicast Write ID x Register", where 'x' is 5. It contains the Multicast ID for which the associated multicast mask registers are applicable. The Tsi620 Switch supports eight multicast groups, therefore, the Multicast ID registers for each multicast group must contain unique values. These registers are used in every RapidIO port.

Register name: SP{0,1,2,3,4,5,6,8}_RIO_MC_ID5 **Reset value: 0x0000_0000**	**Register offset: 16014, 16114, 16214, 16314, 16414, 16514, 16614, 16814**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	MC_EN	LARGE_ SYS	Reserved					
8:15	Reserved							
16:23	MC_ID[0:7]							
24:31	MC_ID[8:15]							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	MC_EN	Multicast can be disabled by setting this bit. 0 = Multicast disabled 1 = Multicast enabled	R/W	0
1	LARGE_SY S	This field defines MC_ID in the Large or Small system. The MC_ID of Small system is not a subset of MC_ID of Large system, but both systems can co-exist together. 1 = Large system 0 = Small system	R/W	0
2:15	Reserved	N/A	R	0
16:31	MC_ID	This field defines the multicast ID for which the associated multicast mask is activated for this extended features block.	R/W	0x0000

When using this register, multiple identical entries must not exist since an addition of an association will not delete the existing one.

25.11.7 RapidIO Serial Port x Multicast Write ID 6 Register

This register is the per port copy of the "RapidIO Multicast Write ID x Register", where 'x' is 6. It contains the Multicast ID for which the associated multicast mask registers are applicable. The Tsi620 Switch supports eight multicast groups, therefore, the Multicast ID registers for each multicast group must contain unique values. These registers are used in every RapidIO port.

Register name: SP{0,1,2,3,4,5,6,8}_RIO_MC_ID6 **Reset value: 0x0000_0000**	**Register offset: 16018, 16118, 16218, 16318, 16418, 16518, 16618, 16818**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	MC_EN	LARGE_ SYS	Reserved					
8:15	Reserved							
16:23	MC_ID[0:7]							
24:31	MC_ID[8:15]							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	MC_EN	Multicast can be disabled by setting this bit. 0 = Multicast disabled 1 = Multicast enabled	R/W	0
1	LARGE_SY S	This field defines MC_ID in the Large or Small system. The MC_ID of Small system is not a subset of MC_ID of Large system, but both systems can co-exist together. 1 = Large system 0 = Small system	R/W	0
2:15	Reserved	N/A	R	0
16:31	MC_ID	This field defines the multicast ID for which the associated multicast mask is activated for this extended features block.	R/W	0x0000

When using this register, multiple identical entries must not exist since an addition of an association will not delete the existing one.

25.11.8 RapidIO Serial Port x Multicast Write ID 7 Register

This register is the per port copy of the "RapidIO Multicast Write ID x Register", where 'x' is 7. It contains the Multicast ID for which the associated multicast mask registers are applicable. The Tsi620 Switch supports eight multicast groups, therefore, the Multicast ID registers for each multicast group must contain unique values. These registers are used in every RapidIO port.

Register name: SP{0,1,2,3,4,5,6,8}_RIO_MC_ID7 **Reset value: 0x0000_0000**	**Register offset: 1601C, 1611C, 1621C, 1631C, 1641C, 1651C, 1661C, 1681C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	MC_EN	LARGE_ SYS	Reserved					
8:15	Reserved							
16:23	MC_ID[0:7]							
24:31	MC_ID[8:15]							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	MC_EN	Multicast can be disabled by setting this bit. 0 = Multicast disabled 1 = Multicast enabled	R/W	0
1	LARGE_SY S	This field defines MC_ID in the Large or Small system. The MC_ID of Small system is not a subset of MC_ID of Large system, but both systems can co-exist together. 1 = Large system 0 = Small system	R/W	0
2:15	Reserved	N/A	R	0
16:31	MC_ID	This field defines the multicast ID for which the associated multicast mask is activated for this extended features block.	R/W	0x0000

When using this register, multiple identical entries must not exist since an addition of an association will not delete the existing one.

25.11.9 RapidIO Serial Port x Switch Port Link Timeout Control CSR

This register is the per port copy of the "RapidIO Switch Port Link Timeout Control CSR". Note that this register does not exist for Switch Port 8.

Register name: SP{0,1,2,3,4,5,6}_RIO_SW_LT_CTL Reset value: 0xFFFF_FF00	Register offset: 16020, 16120, 16220, 16320, 16420, 16520, 16620

Bits	0	1	2	3	4	5	6	7
00:07	TVAL							
08:15	TVAL							
16:23	TVAL							
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0:23	TVAL	Timeout Interval Value Timeout = (32/F) * TVAL, where F is the register bus frequency. For a 156.25 MHz reference clock frequency, the register bus frequency is 78.125 MHz and the default value of this register gives a timeout of 6.9 seconds. For a 125 MHz reference clock frequency, the register bus frequency is 62.5 MHz and the default value of this register gives a timeout of 8.6 seconds. When TVAL is 0, the timer is disabled.	R/W	0xFFFFFF
24:31	Reserved	N/A	R	0

25.11.10 RapidIO Serial Port x Port Write Target Device ID CSR

This register is the per port copy of the "RapidIO Port Write Target Device ID CSR". Note that this register controls no functionality in the Internal Switch Port (Port 8).

Register name: SP{0,1,2,3,4,5,6,8}_RIO_PW_DESTID **Reset value: 0x0000_0000**	**Register offset: 16028, 16128, 16228, 16328, 16428, 16528, 16628, 16828**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	DESTID_MSB							
08:15	DESTID_LSB							
16:23	LARGE_ DESTID	Reserved						
24:31	Reserved							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:7	DESTID_MSB	Most Significant Byte of Port-Write Target Device ID. Used only when LARGE_DESTID is 1.	R/W	0
8:15	DESTID_LSB	If LARGE_DESTID is 0, the DESTID_LSB field is the 8-bit DESTID used in locally-generated Port-Write requests. If LARGE_DESTID is 1, the DESTID_LSB field forms the least significant bits of a 16-bit DestID used in locally-generated Port-Write requests.	R/W	0
16	LARGE_DESTID	0 = Port-write transactions are generated with an 8-bit destination ID. 1 = Port-write transactions are generated with a 16-bit destination ID.	R/W	0
17:31	Reserved	N/A	R	0

25.11.11 RapidIO Serial Port x Switch Port General Control CSR

This register is the per port copy of the "RapidIO Switch Port General Control CSR".

Register name: SP{0,1,2,3,4,5,6,8}_RIO_SW_GEN_CTL Reset value: 0x0000_0000	Register offset: 1603C, 1613C, 1623C, 1633C, 1643C, 1653C, 1663C, 1683C

Bits	0	1	2	3	4	5	6	7
00:07	Reserved		DISC	Reserved				
08:15	Reserved							
16:23	Reserved							
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0:1	Reserved	N/A	R	0
2	DISC	Discovered The Tsi620 is located by the processing element that configures the system. 1 = Device discovered by system host 0 = Device not discovered	R/W	0
3:31	Reserved	N/A	R	0

25.11.12 RapidIO Serial Port x Component Tag CSR

This register is the per port copy of the "RapidIO Component Tag CSR". Note that this register controls no functionality in the Internal Switch Port (Port 8).

Register name: SP{0,1,2,3,4,5,6,8}_RIO_COMP_TAG **Reset value: 0x0000_0000**	**Register offset: 1606C, 1616C, 1626C, 1636C, 1646C, 1656C, 1666C, 1686C**

Bits	0	1	2	3	4	5	6	7
00:07	CTAG[0:7]							
08:15	CTAG[8:15]							
16:23	CTAG[16:23]							
24:31	CTAG[24:31]							

Bits	Name	Description	Type	Reset Value
00:31	CTAG	Component Tag	R/W	0

25.11.13 RapidIO Serial Port x Route LUT Attributes (Default Port) CSR

This register is the per port copy of the "RapidIO Route LUT Attributes (Default Port) CSR"..

Register name: SP{0,1,2,3,4,5,6,8}_RIO_LUT_ATTR Reset value: 0x0000_00FF	Register offset: 16078, 16178, 16278, 16378, 16478, 16578, 16678, 16878

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							
24:31	DEFAULT_PORT							

Bits	Name	Description	Type	Reset Value
0:23	Reserved	Reserved	R	0
24:31	DEFAULT_PORT	Default Output Port All transactions with destination IDs not defined in the LUT are routed to this predefined default output port. DEFAULT_PORT can be set to "unmapped" with a value greater than RIO_SW_PORT[PORT_TOTAL]. For compatibility with future IDT devices, it is recommended that the value 0xFF be used to indicate "unmapped". If a packet needs to consult the default route and the default route is "unmapped", the packet is discarded. For a mapping of port numbers to physical ports, see "RapidIO Port Numbering".	R/W	0xFF

25.12 Switch ISF Registers

These registers provide control and status information concerning timeout errors in data crossing the Switch ISF.

25.12.1 Switch ISF Control Register

The TEA signal is asserted when a timeout is detected on the Switch ISF due to the requested destination being blocked. When this signal is asserted, it indicates to the source of the transaction that the requested transaction could not be completed and is removed from the request queue.

The TEA error is reported through a port-write and/or an interrupt.

Register name: FAB_CTL **Reset value: 0x0F01_0200**	**Register offset: 1AA00**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved				RDR_LIMIT			
08:15	RDR_LIMIT_EN	Reserved	IN_ARB_MODE		Reserved		TEA_INT_EN	TEA_EN
16:23	TEA_OUT[0:7]							
24:31	TEA_OUT[8:15]							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:3	Reserved	N/A	R	0x0
4:7	RDR_LIMIT	Reorder Limit. When packets arrive at an ingress port they are sent to the fabric in order. The fabric can change the order due to packet priority (if enabled through IN_ARB_MODE), and the fabric can change the order to avoid head-of-line blocking. For the latter case, a limit can be placed on the number of times a packet allows a lower or same priority packet to be placed ahead of it. This can be used to provide an upper bound on packet latency. If RDR_LIMIT_EN is set to 1, then the value in RDR_LIMIT is the maximum number of times a packet with lower or same priority can be moved ahead of a packet.	R/W	0xF
8	RDR_LIMIT_EN	Reorder Limit Enable. 0 = No limit 1 = Reordering of lower or same priority packets is limited by the value in RDR_LIMIT (recommended)	R/W	0
9	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
10:11	IN_ARB_MODE	Input Arbitration Mode. This field selects the arbitration scheme used by the fabric's ingress arbiters. 0 = First come, first served 1 = Strict Priority 1 2 = Reserved 3 = Strict Priority 2	R/W	0
12:13	Reserved	N/A	R	0
14	TEA_INT_EN	Interrupt Enable for TEA 0 = Disabled 1 = An interrupt is produced when a TEA event occurs.	R/W	0
15	TEA_EN	TEA Enable. 0 = TEA timer is disabled, similar to writing all 0s to the TEA_OUT field. 1 = TEA timer is enabled.	R/W	1
16:31	TEA_OUT	TEA period This value is multiplied by 2^15 to determine the number of Switch ISF clock cycles a request waits for an acknowledge before a transaction error acknowledge (TEA) occurs. For example, assume the Switch ISF clock is operating at maximum frequency of 156.25 MHz, and TEA_OUT is at its default value of 0x0200. The TEA timeout period is: (0x0200) * 2^15 * 6.4 ns = 107.4 ms. A value of 0x0000 disables the TEA timer.	R/W	0x0200

25.12.2 Switch ISF Interrupt Status Register

This register contains a status bit for every port on the fabric. The status bits indicate on which port(s) a Transaction Error Acknowledge (TEA) has occurred. Writing 1 to a bit clears it. The status bits are “ORed” together to produce the IRQ signal.

Register name: FAB_INT_STAT **Reset value: 0x0000_0000**	**Register offset: 1AA04**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							PORT8_ IRQ
24:31	Reserved	PORT6_ IRQ	PORT5_ IRQ	PORT4_ IRQ	PORT3_ IRQ	PORT2_ IRQ	PORT1_ IRQ	PORT0_ IRQ

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:22	Reserved	Reserved	R	0
23	PORT8_IRQ	Serial port 8 IRQ	R/W1C	0
24	Reserved	N/A	R	0
25	PORT6_IRQ	Serial port 6 IRQ	R/W1C	0
26	PORT5_IRQ	Serial port 5 IRQ	R/W1C	0
27	PORT4_IRQ	Serial port 4 IRQ	R/W1C	0
28	PORT3_IRQ	Serial port 3 IRQ	R/W1C	0
29	PORT2_IRQ	Serial port 2 IRQ	R/W1C	0
30	PORT1_IRQ	Serial port 1 IRQ	R/W1C	0
31	PORT0_IRQ	Serial port 0 IRQ	R/W1C	0

25.12.3 Switch ISF Broadcast Buffer Maximum Latency Expired Error Register

This register is a bit vector of ports that have had their maximum latency timer expire. If the AUTODEAD bit is set in the "RapidIO Multicast Maximum Latency Counter CSR", these ports do not receive multicast packets from the broadcast buffers. If any of these bits are set it causes an exception in the Global Interrupt Mask Status (see "RapidIO Multicast Maximum Latency Counter CSR").

Register name: RIO_MC_LAT_ERR **Reset value: 0x0000_0000**	**Register offset: 1AA08**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							P8_ERR
24:31	Reserved	P6_ERR	P5_ERR	P4_ERR	P3_ERR	P2_ERR	P1_ERR	P0_ERR

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:22	Reserved	N/A	R	0
23	P8_ERR	Port 8 violated the maximum multicast latency time, and will not be multicast to.	R/W1C	0
24	Reserved	N/A	R	0
25	P6_ERR	Port 6 violated the maximum multicast latency time, and will not be multicast to.	R/W1C	0
26	P5_ERR	Port 5 violated the maximum multicast latency time, and will not be multicast to.	R/W1C	0
27	P4_ERR	Port 4 violated the maximum multicast latency time, and will not be multicast to.	R/W1C	0
28	P3_ERR	Port 3 violated the maximum multicast latency time, and will not be multicast to.	R/W1C	0
29	P2_ERR	Port 2 violated the maximum multicast latency time, and will not be multicast to.	R/W1C	0
30	P1_ERR	Port 1 violated the maximum multicast latency time, and will not be multicast to.	R/W1C	0
31	P0_ERR	Port 0 violated the maximum multicast latency time, and will not be multicast to.	R/W1C	0

25.12.4 Switch ISF Broadcast Buffer Maximum Latency Expired Override Register

Writing to this register causes the corresponding bits in the "Switch ISF Broadcast Buffer Maximum Latency Expired Error Register" to be set. This bit causes the corresponding broadcast buffer to be purged of all data currently held in the broadcast buffer. Any packet in the process of being transferred to the broadcast buffer is also purged. The packet being sent out from the broadcast buffer, however, finishes transmission and is not purged. If the AUTODEAD bit is set in the "RapidIO Multicast Maximum Latency Counter CSR", the ports are prevented from receiving multicast packets.

Register name: RIO_MC_LAT_ERR_SET **Reset value: 0x0000_0000**	**Register offset: 1AA0C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							P8_SET
24:31	Reserved	P6_SET	P5_SET	P4_SET	P3_SET	P2_SET	P1_SET	P0_SET

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:22	Reserved	N/A	R	0
23	P8_SET	Port 8 multicast mask is overridden each time a 1 is written to this bit.	R/W1S	0
24	Reserved	N/A	R	0
25	P6_SET	Port 6 multicast mask is overridden each time a 1 is written to this bit.	R/W1S	0
26	P5_SET	Port 5 multicast mask is overridden each time a 1 is written to this bit.	R/W1S	0
27	P4_SET	Port 4 multicast mask is overridden each time a 1 is written to this bit.	R/W1S	0
28	P3_SET	Port 3 multicast mask is overridden each time a 1 is written to this bit.	R/W1S	0
29	P2_SET	Port 2 multicast mask is overridden each time a 1 is written to this bit.	R/W1S	0
30	P1_SET	Port 1 multicast mask is overridden each time a 1 is written to this bit.	R/W1S	0
31	P0_SET	Port 0 multicast mask is overridden each time a 1 is written to this bit.	R/W1S	0

25.13 Switch Utility Registers

The Switch Utility block contains global registers for interrupts and clocking.

25.13.1 Switch Interrupt Status Register

This register indicates which block within the Tsi620 has generated an interrupt. The interrupt requests from a specific block are “ORed” together and the value of the output is indicated in this register.

Register name: GLOB_INT_STATUS **Reset value: 0x0000_0000**	**Register offset: 1AC00**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved				RCS	MCS	I2C	TEA
08:15	Reserved						MC_LAT	Reserved
16:23	Reserved							PORT8
24:31	Reserved	PORT6	PORT5	PORT4	PORT3	PORT2	PORT1	PORT0

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:3	Reserved	N/A Note: These upper 4 bits are reserved to be compatible with the “I2C Interface”.	R	0
4	RCS	Combined 4 Reset Control Symbols interrupt status from all ports	R	0
5	MCS	Combined Multicast-Event Control Symbol interrupt status from all ports	R	0
6	I2C	I^2C Interrupt Port	R	0
7	TEA	TEA occurred in fabric. To determine what port(s) incurred the TEA, see “Switch ISF Interrupt Status Register”.	R	0
8:13	Reserved	N/A	R	0
14	MC_LAT	At least one broadcast buffer has exceeded its maximum multicast latency timeout. To determine which ports have incurred this error, see “Switch ISF Broadcast Buffer Maximum Latency Expired Error Register”.	R	0
15:22	Reserved	N/A	R	0
23	PORT8	Port 8 Interrupt	R	0
24	Reserved	N/A	R	0
25	PORT6	Port 6 Interrupt	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
26	PORT5	Port 5 Interrupt	R	0
27	PORT4	Port 4 Interrupt	R	0
28	PORT3	Port 3 Interrupt	R	0
29	PORT2	Port 2 Interrupt	R	0
30	PORT1	Port 1 Interrupt	R	0
31	PORT0	Port 0 Interrupt	R	0

25.13.2 Switch Interrupt Enable Register

This register controls which Tsi620 Switch events will cause an event to be asserted at the top level (see "Interrupt Controller").

Register name: GLOB_INT_ENABLE Reset value: 0x0000_0000	Register offset: 1AC04

Bits	0	1	2	3	4	5	6	7
00:07	Reserved				RCS_EN	MCS_EN	I2C_EN	TEA_EN
08:15	Reserved						MC_LAT_EN	Reserved
16:23	Reserved							PORT8_EN
24:31	Reserved	PORT6_EN	PORT5_EN	PORT4_EN	PORT3_EN	PORT2_EN	PORT1_EN	PORT0_EN

Bits	Name	Description	Type	Reset Value
0:3	Reserved	N/A	R	0
4	RCS_EN	Four Reset Control Symbols Interrupt Enable	R/W	0
5	MCS_EN	Multicast Event Control Symbol Interrupt Enable	R/W	0
6	I2C_EN	I2C Interrupt Port Enable	R/W	0
7	TEA_EN	TEA interrupt Enable	R/W	0
8:13	Reserved	N/A	R	0
14	MC_LAT_EN	Multicast Latency Interrupt Enable	R/W	0
15:22	Reserved	N/A	R	0
23	PORT8_EN	Port 8 Interrupt Enable	R/W	0
24	Reserved	N/A	R	0
25	PORT6_EN	Port 6 Interrupt Enable	R/W	0
26	PORT5_EN	Port 5 Interrupt Enable	R/W	0
27	PORT4_EN	Port 4 Interrupt Enable	R/W	0
28	PORT3_EN	Port 3 Interrupt Enable	R/W	0
29	PORT2_EN	Port 2 Interrupt Enable	R/W	0
30	PORT1_EN	Port 1 Interrupt Enable	R/W	0
31	PORT0_EN	Port 0 Interrupt Enable	R/W	0

25.13.3 Switch Port Write Timeout Control Register

This register defines port-write timeout value. When a port-write is pending, this timer begins counting. When this timer expires and the port write has not yet been cleared, another port-write is sent and the timer begins counting again.

Register name: RIO_PW_TIMEOUT **Reset value: 0x0000_0000**	**Register offset: 1AC14**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	PW_TIMER				Reserved			
08:15	Reserved							
16:23	Reserved							
24:31	Reserved							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:3	PW_TIMER	Port-Write Timer This field defines the time period to repeat sending an error reporting Port-Write request for software assistance. The timer is stopped by software writing to the error detect registers (for information, see "Servicing Port Writes"). The timeout value is computed by = {[167772160 ns x pw_timer_value (in decimal)] + 2x(10 nsec)}, where 10 ns = clock cycle period of the register bus clock. Reference clock frequency is 156.25 MHz 0000 = Disabled. Port-Write is sent once only per event. 0001 = 214.75 ms 0010 = 429.5 ms 0100 = 859 ms 1000 = 1.72 s 1111 = 2.048 us (Debug only) Other values are reserved.	R/W	0
4:31	Reserved	N/A	R	0

25.13.4 Switch Port Write Outstanding Request Register

This register displays the port number that has an outstanding port-write still in the port-write arbiter. After a port-write is sent, any remaining port-write requests from any port set a bit in the register.

Register name: RIO_PW_OREQ_STATUS **Reset value: 0x0000_0000**	**Register offset: 1AC18**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved							
16:23	PORTX_OREG							
24:31	PORTX_OREG							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:15	Reserved	N/A	R	0
16:31	PORTX_OREG	Port X Port Write Outstanding Request When a bit is set, it indicates that an outstanding port-write still exists in the Port-Write arbiter. These bits are read-only. Bit 16:22: Reserved Bit 23: Port 8, Bit 24: Port 7, ... Bit 31: Port 0.	R	0

25.13.5 Switch MCES Pin Control Register

This register controls the operation of the MCES pin.

Register name: MCES_PIN_CTRL Reset value: 0x0000_0000	Register offset: 130D0

Bits	0	1	2	3	4	5	6	7
00:07	Reserved		MCES_CTRL		Reserved			
08:15	Reserved							
16:23	Reserved							
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0:1	Reserved	N/A	R	0
2:3	MCES_CTRL	MCES Pin Control 00 = Disabled; MCES Pin does not affect generation or receipt of Multicast Event Control Symbol 01 = MCES Pin is set as Input (“Generating an MCS”) 10 = MCES Pin is set as Output (“MCS Reception”) 11 = Reserved	R/W	0
4:31	Reserved	N/A	R	0

The MCES_CTRL setting should be completed before traffic. Changing MCES_CTRL setting during operation may result in spurious Multicast Event Control Symbols being sent.

25.14 Multicast Registers

25.14.1 RapidIO Multicast Register Version CSR

This register identifies the multicast register interface version of the IDT specific registers that is supported by this device.

Register name: RIO{0,1,2,3,4,5,6,8}_MC_REG_VER Reset value: 0x0000_0001	Register offset: 1B000,1B100, 1B200, 1B300, 1B400, 1B500, 1B600, 1B800

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							
24:31	REG_VER							

Bits	Name	Description	Type	Reset Value
0:23	Reserved	N/A	R	0
24:31	REG_VER	IDT MC Register Interface Version supported by this device	R	0x01

25.14.2 RapidIO Multicast Maximum Latency Counter CSR

This register identifies the maximum time a packet copy can wait at the head of a broadcast buffer. If this time limit is exceeded the multicast packet and packet copies in flight to the broadcast buffer are dropped, and an interrupt is raised/port-write packet sent. Optionally, the port is removed from future multicast operations until software clears the error.

Register name: RIO{0,1,2,3,4,5,6,8}_MC_LAT_LIMIT **Reset value: 0x0000_0000**	**Register offset: 1B004,1B104, 1B204, 1B304, 1B404, 1B504, 1B604, 1B804**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	AUTODEAD	Reserved						
08:15	MAX_MC_LAT							
16:23	MAX_MC_LAT							
24:31	MAX_MC_LAT							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	AUTODEAD	Remove Port from Multicast if Latency Timer Expires Enable 0 = Do not remove port from multicast operations if the multicast maximum latency timer expires for this port. 1 = Remove this port from future multicast operations if the multicast maximum latency timer expires for this port.	R/W	0
1:7	Reserved	N/A	R	0
8:31	MAX_MC_LAT	The time period after which the oldest packet copy residing in the broadcast buffer is deemed to have expired. If MAX_MC_LAT == 0x0000, the multicast maximum latency feature is disabled. The timeout period is MAX_MC_LAT = 6.4ns * MAX_MC_LAT When the multicast maximum latency counter expires, all packet copies in the broadcast buffer are discarded. Packet copies that are partially transferred to the broadcast buffer are also discarded. A Port-Write packet or interrupt may be issued to report an error. Optionally, the port can be removed from future multicast operations until software recovers the port. Ports with an expired maximum latency timer are marked in the "Switch ISF Broadcast Buffer Maximum Latency Expired Error Register". Note that, if MAX_MC_LAT is set to its maximum value, this is equivalent to the maximum time-to-live timeout value for packets.	R/W	0

25.14.3 RapidIO Port x Switch ISF Watermarks Register

This register controls (egress) buffer allocation for reception of packets from the Switch ISF for each port. Note that for the Internal Switch Port/SREP link, this register does not exist in the Internal Switch Port ISF registers. Instead, watermarks are implemented in the R2I watermark registers in the SREP.

Register name: SP{0,1,2,3,4,5,6}_ISF_WM **Reset value: 0x0001_0203**	**Register offset: 1B008,1B108, 1B208, 1B308, 1B408, 1B508, 1B608**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	Reserved							
8:15	Reserved					PRIO2WM		
16:23	Reserved					PRIO1WM		
24:31	Reserved					PRIO0WM		

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:12	Reserved	N/A	R	0
13:15	PRIO2WM	Priority 2 packets are accepted if the number of free buffer is greater than this value. This value must be smaller than PRIO1WM. Note: It is a programming error for this value to be greater than or equal to PRIO1WM or PRIO0WM or greater than 7.	R/W	0x1
16:20	Reserved	N/A	R	0
21:23	PRIO1WM	Priority 1 packets are accepted if the number of free buffer is greater than this value. This value must be smaller than PRIO0WM. Note: It is a programming error for this value to be greater than or equal to PRIO0WM or greater than 7.	R/W	0x2
24:28	Reserved	N/A	R	0
29:31	PRIO0WM	Priority 0 packets are accepted if the number of free buffer is greater than this value. Note: It is a programming error for this value to be greater than 7..	R/W	0x3

Do not program this register while traffic is flowing; only program these registers after reset.

25.14.4 RapidIO Port x Prefer Unicast and Multicast Packet Priority 0 Register

This register is used by the egress arbitration to control desired percentage of packets of either multicast or unicast within the same priority group (see "Arbitration for Egress Port").

Register name: SP{0,1,2,3,4,5,6,8}_WRR_0 Reset value: 0x0000_0000	Register offset: 1B010,1B110, 1B210, 1B310, 1B410, 1B510, 1B610, 1B810

Bits	0	1	2	3	4	5	6	7
00:07	Reserved						WRR_EN	CHOOSE_UC
08:15	Reserved							
16:23	Reserved							
24:31	Reserved				WEIGHT			

Bits	Name	Description	Type	Reset Value
0:5	Reserved	N/A	R	0
6	WRR_EN	Weighted Round Robin Enable WRR_EN=0 Weighted Round Robin is disabled and no preference is given to multicast or unicast packets. The registers WEIGHT and CHOOSE_UC will have no effect. WRR_EN=1 Weight Round Robin is enabled and the WEIGHT will be applied to the preferred traffic chosen by CHOOSE_UC.	R/W	0
7	CHOOSE_UC	Set the preferred traffic type within the same priority group. 0 = Multicast 1 = Unicast	R/W	0
8:27	Reserved	N/A	R	0
28:31	WEIGHT	This sets the number of packets of the chosen type to be sent between non-chosen type.	R/W	0

25.14.5 RapidIO Port x Prefer Unicast and Multicast Packet Priority 1 Register

This register is used by the egress arbitration to control desired percentage of packets of either multicast or unicast within the same priority group (see "Arbitration for Egress Port").

Register name: SP{0,1,2,3,4,5,6,8}_WRR_1 Reset value: 0x0000_0000	Register offset: 1B014,1B114, 1B214, 1B314, 1B414, 1B514, 1B614, 1B814

Bits	0	1	2	3	4	5	6	7
00:07	Reserved						WRR_EN	CHOOSE_UC
08:15	Reserved							
16:23	Reserved							
24:31	Reserved				WEIGHT			

Bits	Name	Description	Type	Reset Value
0:5	Reserved	N/A	R	0
6	WRR_EN	0 = Weighted Round Robin is disabled and no preference is given to multicast nor unicast packets. The registers WEIGHT and CHOOSE_UC will have no effect. 1 = Weight Round Robin is enabled and the WEIGHT will be applied to the preferred traffic chosen by CHOOSE_UC.	R/W	0
7	CHOOSE_UC	Set the preferred traffic type within the same priority group. 0 = Multicast 1 = Unicast	R/W	0
8:27	Reserved	N/A	R	0
28:31	WEIGHT	This sets the number of packets of the chosen type to be sent between non-chosen type.	R/W	0

25.14.6 RapidIO Port x Prefer Unicast and Multicast Packet Priority 2 Register

This register is used by the egress arbitration to control desired percentage of packets of either multicast or unicast within the same priority group (see "Arbitration for Egress Port").

Register name: SP{0,1,2,3,4,5,6,8}_WRR_2 Reset value: 0x0000_0000	Register offset: 1B018,1B118, 1B218, 1B318, 1B418, 1B518, 1B618, 1B818

Bits	0	1	2	3	4	5	6	7
00:07	Reserved						WRR_EN	CHOOSE_UC
08:15	Reserved							
16:23	Reserved							
24:31	Reserved				WEIGHT			

Bits	Name	Description	Type	Reset Value
0:5	Reserved	N/A	R	0
6	WRR_EN	0 = Weighted Round Robin is disabled and no preference is given to multicast nor unicast packets. The registers WEIGHT and CHOOSE_UC will have no effect. 1 = Weight Round Robin is enabled and the WEIGHT will be applied to the preferred traffic chosen by CHOOSE_UC.	R/W	0
7	CHOOSE_UC	Set the preferred traffic type within the same priority group. 0 = Multicast 1 = Unicast	R/W	0
8:27	Reserved	N/A	R	0
28:31	WEIGHT	This sets the number of packets of the chosen type to be sent between non-chosen type.	R/W	0

25.14.7 RapidIO Port x Prefer Unicast and Multicast Packet Priority 3 Register

This register is used by the egress arbitration to control desired percentage of packets of either multicast or unicast within the same priority group (see "Arbitration for Egress Port").

Register name: SP{0,1,2,3,4,5,6,8}_WRR_3 Reset value: 0x0000_0000	Register offset: 1B01C,1B11C, 1B21C, 1B31C, 1B41C, 1B51C, 1B61C, 1B81C

Bits	0	1	2	3	4	5	6	7
00:07	Reserved						WRR_EN	CHOOSE_UC
08:15	Reserved							
16:23	Reserved							
24:31	Reserved				WEIGHT			

Bits	Name	Description	Type	Reset Value
0:5	Reserved	N/A	R	0
6	WRR_EN	0 = Weighted Round Robin is disabled and no preference is given to multicast nor unicast packets. The registers WEIGHT and CHOOSE_UC will have no effect. 1 = Weight Round Robin is enabled and the WEIGHT will be applied to the preferred traffic chosen by CHOOSE_UC.	R/W	0
7	CHOOSE_UC	Set the preferred traffic type within the same priority group. 0 = Multicast 1 = Unicast	R/W	0
8:27	Reserved	N/A	R	0
28:31	WEIGHT	This sets the number of packets of the chosen type to be sent between non-chosen type.	R/W	0

25.15 SerDes Per Lane Register

This section discusses the access registers that control the functionality of the SerDes in the Tsi620.

The SerDes register offsets in this section are based on lane 0. In order to define lanes 1, 2, and 3 the offset is incremented by 0x40 for each lane. For example, 0x1E000 represents lane 0 of SerDes 0, 0x1E040 represents lane 1 of SerDes 0, 0x1E080 represents lane 2 of SerDes 0, and 0x1E0C0 represents lane 3 of SerDes 0.

The reset values of the registers listed in this section are only valid when the SerDes are fully initialized. Any read operations to these registers before the SerDes is initialized returns meaningless values. The SerDes is fully initialized when MPLL_PWR_ON=1 (“RapidIO SMAC x SerDes Configuration Global Register”).

When software has powered-down a port, 10us must pass before the port is powered-up again.

Table 161: SerDes Register Map

Channel	Register Offset	Notes
0	1E000 - 1E03F	SerDes Per Lane Register
1	1E040 - 1E07F	
2	1E080 - 1E0BF	
3	1E0C0 - 1E0FF	
Clock	1E100 - 1E13F	SerDes Clock Register

25.15.1 SerDes N Lane 0 Pattern Generator Control Register

This register controls the Pattern Generator in each lane.

Register name: SMAC{0,2,4}_PG_CTL_0 Reset value: 0x0000_0000	Register offset: 1E020, 1E220, 1E420

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved							
16:23	Reserved			PAT0				
24:31	PAT0				TRIGGER_ERR	MODE		

Bits	Name	Description	Type	Reset Value
0:17	Reserved	NA	R	0x0
18:27	PAT0	Pattern for modes 3-5. Program the desired pattern in these 10 bits when using modes 3-5. Note: This field returns to its reset value on reset	R/W	0x0
28	TRIGGER_ERR	Insert a single error into a LSB Note: This field returns to its reset value on reset	R/W	0x0
29:31	MODE	Pattern to Generate 0 = Disabled 1 = lfsr15 ($x^{15}+x^{14}+1$) 2 = lfsr7 ($x^{7}+x^{6}+1$) 3 = Fixed word (pat0) 4 = DC balanced word (pat0, ~pat0) 5 = Fixed pattern: (000, pat0, 3FF, ~pat0) 6:7 = Reserved Note: This field returns to its reset value on reset	R/W	0x0

25.15.2 SerDes N Lane 0 Pattern Matcher Control Register

This register contains the controls the Pattern Matcher and the error counters associated with the corresponding matcher in each lane.

Register name: SMAC{0,2,4}_PM_CTL_0 **Reset value: 0x0000_0000**	**Register offset: 1E030, 1E230, 1E430**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	OV14	COUNT						
08:15	COUNT							
16:23	Reserved							
24:31	Reserved				SYNC	MODE		

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	OV14	1= multiply COUNT by 128. When OV14=1 & count = 2^{15}-1, signal overflows Note: Read operations on this register is pipelined. Two reads needed to get current value. The values are volatile and the value may change at any time	R/W	0x0
1:15	COUNT	Current error count If OV14 field is active, multiply count by 128. Note: Read operation on this register is pipelined. Two reads needed to get “current” value. The values are volatile (that is, value may change at any time)	R/W	0x0
16:27	Reserved	NA	R	0x0
28	SYNC	Synchronize pattern matcher LFSR with incoming data. Must be turned on then off to enable checking. RX_ALIGN_EN must be disabled when checking PRBS patterns Note: This bit returns to its reset value on reset	R/W	0x0
29:31	MODE	Pattern to match 0 = Disabled 1 = lfsr15 2 = lfsr7 3 = d[n] = d[n-10] 4 = d[n] = !d[n-10] 5:7 = Reserved Note: This field returns to its reset value on reset	R/W	0x0

25.15.3 SerDes N Lane 0 Frequency and Phase Value Register

This register contains the frequency and phase of the incoming eyes on the SerDes.

Register name: SMAC{0,2,4}_FP_VAL_0 Reset value: 0x0000_0000	Register offset: 1E034, 1E234, 1E434

Bits	0	1	2	3	4	5	6	7
00:07	Reserved		FVAL					
08:15	FVAL							DTHR_0
16:23	Reserved					PVAL		
24:31	PVAL							DTHR_1

Bits	Name	Description	Type	Reset Value
0:1	Reserved	NA	R	0x0
2:14	FVAL	Frequency is 1.526*FVAL ppm from the reference Note: Read operations on this register is pipelined. Two reads needed to get current value. The values are volatile and the value may change at any time	R/W	0x0
15	DTHR_0	Bits below the useful resolution Note: Read operations on this register is pipelined. Two reads needed to get current value. The values are volatile and the value may change at any time	R/W	0
16:20	Reserved	NA	R	0x0
21:30	PVAL	Phase is 0.78125*pval ps from zero reference Note: Read operations on this register is pipelined. Two reads needed to get current value. The values are volatile and the value may change at any time	R/W	0x0
31	DTHR_1	Bits below the useful resolution Note: Read operations on this register is pipelined. Two reads needed to get current value. The values are volatile and the value may change at any time	R/W	0x0

25.15.4 SerDes N Lane 1 Pattern Generator Control Register

This register controls the Pattern Generator in each lane.

Register name: SMAC{0,2,4}_PG_CTL_1 Reset value: 0x0000_0000	Register offset: 1E060, 1E260, 1E460

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved							
16:23	Reserved		PAT0					
24:31	PAT0				TRIGGER_ERR	MODE		

Bits	Name	Description	Type	Reset Value
0:17	Reserved	NA	R	0x0
18:27	PAT0	Pattern for modes 3-5. Program the desired pattern in these 10 bits when using modes 3-5. Note: This field returns to its reset value on reset.	R/W	0x0
28	TRIGGER_ERR	Insert a single error into a LSB Note: This field returns to its reset value on reset.	R/W	0x0
29:31	MODE	Pattern to Generate 0 = Disabled 1 = lfsr15 ($x^{15}+x^{14}+1$) 2 = lfsr7 ($x^{7}+x^{6}+1$) 3 = Fixed word (pat0) 4 = DC balanced word (pat0, ~pat0) 5 = Fixed pattern: (000, pat0, 3FF, ~pat0) 6:7 = Reserved Note: This field returns to its reset value on reset.	R/W	0x0

25.15.5 SerDes N Lane 1 Pattern Matcher Control Register

This register contains the controls the Pattern Matcher and the error counters associated with the corresponding matcher in each lane.

Register name: SMAC{0,2,4}_PM_CTL_1 Reset value: 0x0000_0000	Register offset: 1E070, 1E270, 1E470

Bits	0	1	2	3	4	5	6	7
00:07	OV14	COUNT						
08:15	COUNT							
16:23	Reserved							
24:31	Reserved				SYNC	MODE		

Bits	Name	Description	Type	Reset Value
0	OV14	1= multiply COUNT by 128. When OV14=1 & count = 2^{15}-1, signal overflows Note: Read operations on this register is pipelined. Two reads needed to get current value. The values are volatile and the value may change at any time	R/W	0x0
1:15	COUNT	Current error count If OV14 field is active, multiply count by 128. Note: Read operation on this register is pipelined. Two reads needed to get “current” value. The values are volatile (that is, value may change at any time)	R/W	0x0
16:27	Reserved	NA	R	0x0
28	SYNC	Synchronize pattern matcher LFSR with incoming data. Must be turned on then off to enable checking. RX_ALIGN_EN must be disabled when checking PRBS patterns Note: This bit returns to its reset value on reset	R/W	0x0
29:31	MODE	Pattern to match 0 = Disabled 1 = lfsr15 2 = lfsr7 3 = d[n] = d[n-10] 4 = d[n] = !d[n-10] 5:7 = Reserved Note: This field returns to its reset value on reset	R/W	0x0

25.15.6 SerDes N Lane 1 Frequency and Phase Value Register

This register contains the frequency and phase of the incoming eyes on the SerDes.

Register name: SMAC{0,2,4}_FP_VAL_1 Reset value: 0x0000_0000	Register offset: 1E074, 1E274, 1E474

Bits	0	1	2	3	4	5	6	7
00:07	Reserved		FVAL					
08:15	FVAL							DTHR_0
16:23	Reserved					PVAL		
24:31	PVAL							DTHR_1

Bits	Name	Description	Type	Reset Value
0:1	Reserved	NA	R	0x0
2:14	FVAL	Frequency is 1.526*FVAL ppm from the reference Note: Read operations on this register is pipelined. Two reads needed to get current value. The values are volatile and the value may change at any time	R/W	0x0
15	DTHR_0	Bits below the useful resolution Note: Read operations on this register is pipelined. Two reads needed to get current value. The values are volatile and the value may change at any time	R/W	0
16:20	Reserved	NA	R	0x0
21:30	PVAL	Phase is 0.78125*pval ps from zero reference Note: Read operations on this register is pipelined. Two reads needed to get current value. The values are volatile and the value may change at any time	R/W	0x0
31	DTHR_1	Bits below the useful resolution Note: Read operations on this register is pipelined. Two reads needed to get current value. The values are volatile and the value may change at any time	R/W	0x0

25.15.7 SerDes N Lane 2 Pattern Generator Control Register

This register controls the Pattern Generator in each lane.

Register name: SMAC{0,2,4}_PG_CTL_2 Reset value: 0x0000_0000	Register offset: 1E0A0, 1E2A0, 1E4A0

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved							
16:23	Reserved			PAT0				
24:31	PAT0				TRIGGER_ERR	MODE		

Bits	Name	Description	Type	Reset Value
0:17	Reserved	NA	R	0x0
18:27	PAT0	Pattern for modes 3-5. Program the desired pattern in these 10 bits when using modes 3-5. Note: This field returns to its reset value on reset	R/W	0x0
28	TRIGGER_ERR	Insert a single error into a LSB Note: This field returns to its reset value on reset	R/W	0x0
29:31	MODE	Pattern to Generate 0 = Disabled 1 = lfsr15 ($x^{15}+x^{14}+1$) 2 = lfsr7 ($x^{7}+x^{6}+1$) 3 = Fixed word (pat0) 4 = DC balanced word (pat0, ~pat0) 5 = Fixed pattern: (000, pat0, 3FF, ~pat0) 6:7 = Reserved Note: This field returns to its reset value on reset	R/W	0x0

25.15.8 SerDes N Lane 2 Pattern Matcher Control Register

This register contains the controls the Pattern Matcher and the error counters associated with the corresponding matcher in each lane.

Register name: SMAC{0,2,4}_PM_CTL_2 **Reset value: 0x0000_0000**	**Register offset: 1E0B0, 1E2B0, 1E4B0**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	OV14	COUNT						
08:15	COUNT							
16:23	Reserved							
24:31	Reserved				SYNC	MODE		

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	OV14	1= multiply COUNT by 128. When OV14=1 & count = 2^{15}-1, signal overflows Note: Read operations on this register is pipelined. Two reads needed to get current value. The values are volatile and the value may change at any time	R/W	0x0
1:15	COUNT	Current error count If OV14 field is active, multiply count by 128. Note: Read operation on this register is pipelined. Two reads needed to get “current” value. The values are volatile (that is, value may change at any time)	R/W	0x0
16:27	Reserved	NA	R	0x0
28	SYNC	Synchronize pattern matcher LFSR with incoming data. Must be turned on then off to enable checking. RX_ALIGN_EN must be disabled when checking PRBS patterns Note: This bit returns to its reset value on reset	R/W	0x0
29:31	MODE	Pattern to match 0 = Disabled 1 = lfsr15 2 = lfsr7 3 = d[n] = d[n-10] 4 = d[n] = !d[n-10] 5:7 = Reserved Note: This field returns to its reset value on reset	R/W	0x0

25.15.9 SerDes N Lane 2 Frequency and Phase Value Register

This register contains the frequency and phase of the incoming eyes on the SerDes.

Register name: SMAC{0,2,4}_FP_VAL_2 Reset value: 0x0000_0000	Register offset: 1E0B4, 1E2B4, 1E4B4

Bits	0	1	2	3	4	5	6	7
00:07	Reserved		FVAL					
08:15	FVAL							DTHR_0
16:23	Reserved					PVAL		
24:31	PVAL							DTHR_1

Bits	Name	Description	Type	Reset Value
0:1	Reserved	NA	R	0x0
2:14	FVAL	Frequency is 1.526*FVAL ppm from the reference Note: Read operations on this register is pipelined. Two reads needed to get current value. The values are volatile and the value may change at any time	R/W	0x0
15	DTHR_0	Bits below the useful resolution Note: Read operations on this register is pipelined. Two reads needed to get current value. The values are volatile and the value may change at any time	R/W	0
16:20	Reserved	NA	R	0x0
21:30	PVAL	Phase is 0.78125*pval ps from zero reference Note: Read operations on this register is pipelined. Two reads needed to get current value. The values are volatile and the value may change at any time	R/W	0x0
31	DTHR_1	Bits below the useful resolution Note: Read operations on this register is pipelined. Two reads needed to get current value. The values are volatile and the value may change at any time	R/W	0x0

25.15.10 SerDes N Lane 3 Pattern Generator Control Register

This register controls the Pattern Generator in each lane.

Register name: SMAC{0,2,4}_PG_CTL_3 Reset value: 0x0000_0000	Register offset: 1E0E0, 1E2E0, 1E4E0

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved							
16:23	Reserved		PAT0					
24:31	PAT0				TRIGGER_ERR	MODE		

Bits	Name	Description	Type	Reset Value
0:17	Reserved	NA	R	0x0
18:27	PAT0	Pattern for modes 3-5. Program the desired pattern in these 10bits when using modes 3-5. Note: This field returns to its reset value on reset	R/W	0x0
28	TRIGGER_ERR	Insert a single error into a LSB Note: This field returns to its reset value on reset	R/W	0x0
29:31	MODE	Pattern to Generate 0 = Disabled 1 = lfsr15 ($x^{15}+x^{14}+1$) 2 = lfsr7 ($x^{7}+x^{6}+1$) 3 = Fixed word (pat0) 4 = DC balanced word (pat0, ~pat0) 5 = Fixed pattern: (000, pat0, 3FF, ~pat0) 6:7 = Reserved Note: This field returns to its reset value on reset	R/W	0x0

25.15.11 SerDes N Lane 3 Pattern Matcher Control Register

This register contains the controls the Pattern Matcher and the error counters associated with the corresponding matcher in each lane.

Register name: SMAC{0,2,4}_PM_CTL_3 Reset value: 0x0000_0000	Register offset: 1E0F0, 1E2F0, 1E4F0

Bits	0	1	2	3	4	5	6	7
00:07	OV14	COUNT						
08:15	COUNT							
16:23	Reserved							
24:31	Reserved				SYNC	MODE		

Bits	Name	Description	Type	Reset Value
0	OV14	1= multiply COUNT by 128. When OV14=1 & count = 2^{15}-1, signal overflows Note: Read operations on this register is pipelined. Two reads needed to get current value. The values are volatile and the value may change at any time	R/W	0x0
1:15	COUNT	Current error count If OV14 field is active, multiply count by 128. Note: Read operation on this register is pipelined. Two reads needed to get “current” value. The values are volatile (that is, value may change at any time)	R/W	0x0
16:27	Reserved	NA	R	0x0
28	SYNC	Synchronize pattern matcher LFSR with incoming data. Must be turned on then off to enable checking. RX_ALIGN_EN must be disabled when checking PRBS patterns Note: This bit returns to its reset value on reset	R/W	0x0
29:31	MODE	Pattern to match 0 = Disabled 1 = lfsr15 2 = lfsr7 3 = d[n] = d[n-10] 4 = d[n] = !d[n-10] 5:7 = Reserved Note: This field returns to its reset value on reset	R/W	0x0

25.15.12 SerDes N Lane 3 Frequency and Phase Value Register

This register contains the frequency and phase of the incoming eyes on the SerDes.

Register name: SMAC{0,2,4}_FP_VAL_3 Reset value: 0x0000_0000	Register offset: 1E0F4, 1E2F4, 1E4F4

Bits	0	1	2	3	4	5	6	7
00:07	Reserved		FVAL					
08:15	FVAL							DTHR_0
16:23	Reserved					PVAL		
24:31	PVAL							DTHR_1

Bits	Name	Description	Type	Reset Value
0:1	Reserved	NA	R	0x0
2:14	FVAL	Frequency is 1.526*FVAL ppm from the reference Note: Read operations on this register is pipelined. Two reads needed to get current value. The values are volatile and the value may change at any time	R/W	0x0
15	DTHR_0	Bits below the useful resolution Note: Read operations on this register is pipelined. Two reads needed to get current value. The values are volatile and the value may change at any time	R/W	0
16:20	Reserved	NA	R	0x0
21:30	PVAL	Phase is 0.78125*pval ps from zero reference Note: Read operations on this register is pipelined. Two reads needed to get current value. The values are volatile and the value may change at any time	R/W	0x0
31	DTHR_1	Bits below the useful resolution Note: Read operations on this register is pipelined. Two reads needed to get current value. The values are volatile and the value may change at any time	R/W	0x0

26. SREP Registers

Topics discussed include the following:

26.1 Overview

The SREP registers are defined at the offset that a RapidIO Maintenance Read/Write would use to access them. Different applications may make use of different address maps for devices.

26.1.1 Reserved Register Addresses and Fields

The following rules apply to the SREP registers that fall into the RapidIO standard register space, address offset 0x00000 through 0x0FFFF:

- Reads to reserved register addresses return "0"
- Writes to reserved register addresses complete without error and do not affect the operation of the SREP.

The following rules apply to the SREP registers, which are implementation specific:

- Reads to reserved register addresses return undefined data and may result in undefined behavior
- Writes to reserved register addresses may result in undefined operation.

For all registers, reserved fields within registers return 0 when read. Reserved fields should be written as 0 (unless otherwise specified).

Table 162 shows the defined register access types.

Table 162: Register Access Types

Abbreviation	Description
R	Read Only.
RS	Read Only, Sticky (only reset on Power-up Reset)
R/W	Read or Write.
R/WS	Read or Write, Sticky (only reset on Power-up Reset)
R/W1C	Readable. Write 1 to Clear.
R/W1CS	Readable. Write 1 to Clear. Sticky (only reset on Power-up Reset)
R/W0C	Readable. Write 0 to Clear.
R/W1S	Readable. Write 1 to Set (Writing a 1 triggers an event).
RC	When read these bits are automatically cleared.

This register specification uses direct addressing of 32-bit registers. The *RapidIO Interconnect Specification (Revision 1.2)* uses 64-bit addressing of registers. Table 163 shows the rules that associate the register offsets in both specifications.

Table 163: Address rules

SREP FS Address — Register Offset	RapidIO Specification Address — Register Offset
0x*XXXX*0	0x*XXXX*0, Word 0
0x*XXXX*4	0x*XXXX*0, Word 1
0x*XXXX*8	0x*XXXX*8, Word 0
0x*XXXX*C	0x*XXXX*8, Word 1

26.1.2 Conventions

Register fields are named using “dotted” notation: <registerName>.<fieldName>. For example, the MSB field of the DESTID register is DESTID.MSB.

Often, there are multiple instances of a register, for example, Base Address Registers (BARs). Two notations refer to such registers.

- In the first notation, a lower-case letter such as “x” is used as a wildcard character. For example, Px_DESTID refers to P0_DESTID, P1_DESTID, P2_DESTID, and so on.
- In the second notation, the names of the instances are listed. For example, P{BC,0..2}_DESTID refers to registers PBC_DESTID, P0_DESTID, P1_DESTID, and P2_DESTID.

26.2 Register Map

The register addressing scheme places the standard registers at their standard locations when accessed from RapidIO using maintenance transactions. However, within the Tsi620 register map, the registers occur in a contiguous 4 KB block.

Table 164 gives an overview of the SREP register map when accessed using RapidIO maintenance transactions.

Table 164: SREP Register Map Overview

Register Group	Start Address	End Address
RapidIO Logical Layer and Transport Layer Registers (Parts I and III of the RapidIO Specification)	0x00000	0x000FF
RapidIO Physical Layer Registers Extended Features Block (Parts IV and VI of the RapidIO Specification)	0x00100	0x001FF
RapidIO Error Management Extension Registers (Part VIII of the RapidIO Specification)	0x00200	0x002FF
Reserved	0x00300	0x102FF
IDT-Specific RapidIO Registers	0x10300	0x10FFF
Reserved	0x11000	0xFFFFFF

Table 165 gives an overview of the SREP register map when accessed from other ports on the Tsi620, or when accessed using RapidIO NREAD/NWRITE/NWRITE_R transactions.

Table 165: Tsi620 RapidIO Register Map Overview

Register Group	Start Address	End Address
RapidIO Logical Layer and Transport Layer Registers (Parts I and III of the RapidIO Specification)	0xXX000	0xXX0FF
RapidIO Physical Layer Registers Extended Features Block (Parts IV and VI of the RapidIO Specification)	0xXX100	0xXX1FF
RapidIO Error Management Extension Registers (Part VIII of the RapidIO Specification)	0xXX200	0xXX2FF
IDT-Specific RapidIO Registers	0xXX300	0xXXFFF

All registers are documented with addresses as visible from the RapidIO port. "SREP Registers" lists the SREP registers from the RapidIO perspective.

26.3 RapidIO Capability Registers (0x00000–0x0003F)

Every processing element contains a set of capability registers (CARs) that allows another processing element to determine its capabilities through maintenance read operations. All registers are 32 bits wide and are organized and accessed in 32-bit quantities. CARs are read-only. CARs are big-endian — bit 0 is the most significant bit.

A processing element contains a set of command and status registers (CSRs) that allows another processing element to control and determine the status of its internal hardware. All registers are organized and accessed in the same way as the CARs. All of the CAR registers are defined in the *RapidIO Interconnect Specification (Revision 1.3)*.

Reads to reserved register addresses will return 0, writes to reserved register addresses will complete without error and will not affect the operation of the SREP.

Table 166: RapidIO Capability Registers – Register Map

Offset	Register Name	See
0x000	SREP_RIO_DEV_ID	"SREP Device Identity CAR"
0x004	SREP_RIO_DEV_INFO	"SREP Device Information CAR"
0x008	SREP_RIO_ASBLY_ID	"SREP Assembly Identity CAR"
0x00C	SREP_RIO_ASBLY_INFO	"SREP Assembly Information CAR"
0x010	SREP_RIO_PE_FEAT	"SREP Processing Element Features CAR"

Table 166: RapidIO Capability Registers – Register Map

Offset	Register Name	See
0x018	SREP_RIO_SRC_OP	"SREP Source Operation CAR"
0x01C	SREP_RIO_DEST_OP	"SREP Destination Operation CAR"

26.3.1 SREP Device Identity CAR

This register identifies the device and vendor information for the SREP.

Register name: SREP_RIO_DEV_ID Reset value: 0x0620_000D	Register offset: 000

Bits	0	1	2	3	4	5	6	7
00:7	DEV_ID							
07:15	DEV_ID							
16:23	DEV_VEN_ID							
24:31	DEV_VEN_ID							

Bits	Name	Description	Type	Reset Value
0:15	DEV_ID	Device Identifier This field contains the IDT-assigned part number of the device.	R	0x0620
16:31	DEV_VEN_ID	Device Vendor Identifier Identifies IDT Semiconductor Corporation as the vendor that manufactured the device. This value is assigned by the RapidIO Trade Association.	R	0x000D

26.3.2 SREP Device Information CAR

This register identifies version information about the device.

Register name: SREP_RIO_DEV_INFO Reset value: 0x0000_0000	Register offset: 004

Bits	0	1	2	3	4	5	6	7
00:07	UNUSED							
08:15	UNUSED							
16:23	UNUSED							
24:31	SILICON_REV				METAL_REV			

Bits	Name	Description	Type	Reset Value
0:23	UNUSED	Unused.	R	0
24:27	SILICON_REV	Indicates the Major version of silicon used in the device. 0 = A revision Note: This field should have the same value as the "RapidIO Device Information CAR" in the Internal Switch Port (Port 8), the SREP, and the Revision ID value in "PCI Class Register".	R	0
28:31	METAL_REV	Indicates the Minor version of the metal layers for the current silicon version. 0 = First revision of metal layers.Note: This field should have the same value as the "RapidIO Device Information CAR" in the Internal Switch Port (Port 8), the SREP, and the Revision ID value in "PCI Class Register".	R	0

26.3.3 SREP Assembly Identity CAR

This register is writable during I^2C register initialization, and read-only after that time. The register identifies the vendor that manufactured the assembly or subsystem, and the type of assembly that contains the device.

This register should be writable from I^2C before the BOOT_COMPLETE signal is asserted.

Register name: SREP_RIO_ASBLY_ID **Reset value: 0x0001_000D**	**Register offset: 008**

Bits	0	1	2	3	4	5	6	7
00:07	ASBLY_ID							
08:15	ASBLY_ID							
16:23	ASBLY_VEN_ID							
24:31	ASBLY_VEN_ID							

Bits	Name	Description	Type	Reset Value
0:15	ASBLY_ID	Assembly ID. Identifies the type of assembly from the vendor specified by the ASBLY_VEN_ID field. I^2C load from EEPROM	R	0x0001
16:31	ASBLY_VEN_ID	Assembly Vendor ID Identifies the vendor that manufactured the assembly or subsystem that contains the device. I^2C load from EEPROM	R	0x000D

26.3.4 SREP Assembly Information CAR

This register contains additional information about the assembly.

Register name: SREP_RIO_ASBLY_INFO Reset value: 0x0000_0100	Register offset: 00C

Bits	0	1	2	3	4	5	6	7
00:07	ASBLY_REV							
08:15	ASBLY_REV							
16:23	EXT_FEAT_PTR							
24:31	EXT_FEAT_PTR							

Bits	Name	Description	Type	Reset Value
0:15	ASBLY_REV	Assembly Revision Level	R	0x0000
16:31	EXT_FEAT_PTR	Extended Features Pointer Pointer to the first entry in the extended features list.	R	0x0100

26.3.5 SREP Processing Element Features CAR

This register identifies the major functionality provided by the processing element.

Register name: SREP_RIO_PE_FEAT Reset value: 0xC000_0019	Register offset: 010

Bits	0	1	2	3	4	5	6	7
00:07	BRDG	MEM	PROC	SW	Reserved			
08:15	Reserved							
16:23	Reserved							
24:31	Reserved	NO_ RECOV	CRF	CTLS	EXT_FEA	EXT_AS		

Bits	Name	Description	Type	Reset Value
0	BRDG	Bridge 0 = The processing element is not a bridge 1 = The processing element can bridge to another interface.	R	1
1	MEM	Endpoint 0 = Not a RapidIO endpoint addressable for reads and writes 1 = The processing element has physically addressable local address space and can be accessed as an endpoint through non-maintenance (that is, NREAD and NWRITE) transactions	R	1
2	PROC	Processor 0 = Not a processor 1 = Physically contains a local processor or similar device that executes code. A device that bridges to an interface that connects to a processor does not count (see bit 0).	R	0
3	SW	Switching Capabilities Does the device have the ability to bridge to another external RapidIO port. 0 = Not capable. Ftype 8 request packets with any hop count value are routed to the register bus. 1 = Ftype 8 packets with hop count equal to 0 are routed to the register bus	R	0
4:24	Reserved	N/A	R	0
25	NO_RECOV	PE Supports suppression of error recovery on packet CRC errors 0 = The error recovery suppression option is not supported by the PE 1 = The error recovery suppression option is supported by the PE	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
26	CRF	PE Supports Critical Request Flow (CRF) indicator 0 = Critical Request Flow is not supported 1 = Critical Request Flow is supported	R	0
27	CTLS	Common Transport Large System Support 0 = Device supports 8-bit destination IDs only 1 = Device supports 8- and 16-bit destination ID's	R	1
28	EXT_FEA	Extended Features Pointer is valid Pointer to the first entry in the extended features list.	R	1
29:31	EXT_AS	Extended Addressing Support. 0b001 = Supports 34-bit addresses	R	0b001

26.3.6 SREP Source Operation CAR

The SREP can originate RapidIO Logical I/O requests and Port-Write packets, as indicated by this register.

Register name: SREP_RIO_SRC_OP Reset value: 0x0000_F404	Register offset: 018

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved						IMPLEMENT_DEF	
16:23	READ	WRITE	STRM_WR	WR_RES	D_MSG	DBELL	Reserved	A_TSWAP
24:31	A_INC	A_DEC	A_SET	A_CLEAR	Reserved	PORT_WR	Reserved	

Bits	Name	Description	Type	Reset Value
0:13	Reserved	N/A	R	0
14:15	IMPLEMENT_ DEF	Implementation defined	R	0
16	READ	Read operation supported	R	1
17	WRITE	Write operation supported	R	1
18	STRM_WR	Streaming write operation supported	R	1
19	WR_RES	Write-with-response operation supporteD	R	1
20	D_MSG	Data messaging	R	0
21	DBELL	Doorbell	R	1
22	Reserved	N/A	R	0
23	A_TSWAP	Atomic (test-and-swap) operation supported	R	0
24	A_INC	Atomic (increment) operation supported	R	0
25	A_DEC	Atomic (decrement) operation supported	R	0
26	A_SET	Atomic (set) operation supported	R	0
27	A_CLEAR	Atomic (clear) operation supported	R	0
28	Reserved	N/A	R	0
29	PORT_WR	Port-Write operation The RapidIO port supports generation of port-write packets.	R	1
30:31	Reserved	Implementation defined	R	0

26.3.7 SREP Destination Operation CAR

The SREP can receive RapidIO Logical I/O requests and receive Port-Write packets, as indicated by this register.

Register name: SREP_RIO_DEST_OP Reset value: 0x0000_F404	Register offset: 01C

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved						IMPLEMENT_DEF	
16:23	READ	WRITE	STRM_WR	WR_RES	D_MSG	DBELL	Reserved	A_TSWAP
24:31	A_INC	A_DEC	A_SET	A_CLEAR	Reserved	PORT_WR	Reserved	

Bits	Name	Description	Type	Reset Value
0:13	Reserved	N/A	R	0
14:15	IMPLEMENT_ DEF	Implementation defined	R	0
16	READ	Read operation supported	R	1
17	WRITE	Write operation supported	R	1
18	STRM_WR	Streaming write operation supported	R	1
19	WR_RES	Write-with-response operation supported	R	1
20	D_MSG	Data messaging	R	0
21	DBELL	Doorbell	R	1
22	Reserved	N/A	R	0
23	A_TSWAP	Atomic (test-and-swap) operation supported	R	0
24	A_INC	Atomic (increment) operation supported	R	0
25	A_DEC	Atomic (decrement) operation supported	R	0
26	A_SET	Atomic (set) operation supported	R	0
27	A_CLEAR	Atomic (clear) operation supported	R	0
28	Reserved	N/A	R	0
29	PORT_WR	Port-Write operation The RapidIO ports supports reception of port-write packets.	R	1
30:31	Reserved	N/A	R	0

26.4 RapidIO Device Control Registers (0x00040–0x000FC)

Table 167: Register Map for RapidIO Device Control Registers

Offset	Register Name	See
0x04C	SREP_RIO_LIO_ADDRSZ	"SREP Processing Element Logical Layer Control CSR"
0x05C	SREP_RIO_LIO_REGACC	"SREP Local Configuration Space Base Address CSR"
0x060	SREP_RIO_DEST_ID	"SREP Base Device ID CSR"
0x068	SREP_RIO_HOST_BASE_ID_LOCK	"SREP Host Base Device ID Lock CSR"
0x06C	SREP_RIO_COMP_TAG	"SREP Component Tag CSR"

26.4.1 SREP Processing Element Logical Layer Control CSR

This register controls the address range used by an endpoint. The SREP only supports 34-bit RapidIO addresses, so this register is read only.

Register name: SREP_RIO_LIO_ADDRSZ **Reset value: 0x0000_0001**	**Register offset: 04C**

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							
24:31	Reserved					EXT_ADDR_CTL		

Bits	Name	Description	Type	Reset Value
0:28	Reserved	Reserved	R	0x000000
29:31	EXT_ADDR_CTL	RapidIO Logical I/O Address Size 0b001 = Use 34-bit addresses	R	0b001

26.4.2 SREP Local Configuration Space Base Address CSR

This register controls the logical I/O address range that will be responded to by an endpoint's registers. This register must be read and written using 4-byte accesses. Smaller accesses lead to unexpected results.

If this BAR overlaps any of the other R2I BARs, then this BAR takes precedence (see "Bridging Logical I/O Requests to the Bridge ISF").

This BAR controls a 256 KB range of memory. The address that it responds to is 256-KB aligned. The least significant 15 bits of the ADDR field do not affect the address at which this BAR responds.

Register name: SREP_RIO_LIO_REGACC **Reset value: 0x0000_0000**	**Register offset: 05C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved	ADDR						
08:15	ADDR							
16:23	ADDR							
24:31	ADDR							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	Reserved	N/A	R	0
1:31	ADDR	Most significant 31 address bits of the 34 bit RapidIO Logical I/O address at which the SREP registers can be read using Logical I/O packets. Note: The least significant 15 bits of this field do not affect the address at which the devices registers are located. This register controls at 256-KB aligned address space.	R/W	0

26.4.3 SREP Base Device ID CSR

This register contains the destination ID value supported by this processing element. The SREP can check the destination IDs of packets received against this destination ID, and report an error if the DEST_ID value does not match that of the packet.

This register is also the sourceID of packets originated by the SREP. The DEST_ID or LG_DEST_ID fields are used depending on the setting of the TT field in the "SREP I2R Upper LUT Entry Translation Register". For more information on using this register, see "8/16-bit Destination ID Support".

Register name: SREP_RIO_DEST_ID **Reset value: Undefined**	**Register offset: 060**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	DEST_ID							
16:23	LG_DEST_ID							
24:31	LG_DEST_ID							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:7	Reserved	Reserved	R	0x00
8:15	DEST_ID	8-bit Destination ID that this processing element should respond to. The reset value of this field is determined by configuration signals. (see "Destination ID Initialization").	R/W	Undefined
16:31	LG_DEST_ID	16-bit Destination ID that this processing element should respond to. The reset value of this field is determined by configuration signals (see "Destination ID Initialization").	R/W	Undefined

26.4.4 SREP Host Base Device ID Lock CSR

This register contains the base device ID value for the processing element in the system that initializes this processing element.

The HOST_BASE_ID field is a write-once/reset field. Once the HOST_BASE_ID field is written, all subsequent writes to the field are ignored, except when the value written matches the value in the field. In this case, the register is re-initialized to 0xFFFF.

Note that writing 0xFFFF to this register does not result in a lock being obtained.

After writing the HOST_BASE_ID field, a processing element must read the Host Base Device ID Lock CSR to verify that it owns the lock before attempting to initialize this processing element.

Register name: SREP_RIO_HOST_BASE_ID_LOCK **Reset value: 0x0000_FFFF**	**Register offset: 068**

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved							
16:23	HOST_BASE_ID[16:23]							
24:31	HOST_BASE_ID[24:31]							

Bits	Name	Description	Type	Reset Value
0:15	Reserved	Reserved	R	0x0000
16:31	HOST_BASE_ID	Base Device ID for the processing element that is initializing this endpoint.	R/W	0xFFFF

The HOST_BASE_ID set in this register does not enforce exclusive access to the device. It coordinates device identification during initialization and discovery.

26.4.5 SREP Component Tag CSR

This register is written by software. It contains a unique identifier for each component in a RapidIO system for purposes of identification. The contents of this register are included in automatically generated port-write transactions (see "Port-Writes").

Register name: SREP_RIO_COMP_TAG Reset value: 0x0000_0000	Register offset: 06C

Bits	0	1	2	3	4	5	6	7
00:07	CTAG[0:7]							
08:15	CTAG[8:15]							
16:23	CTAG[16:23]							
24:31	CTAG[24:31]							

Bits	Name	Description	Type	Reset Value
00:31	CTAG	Component Tag	R/W	0

26.5 RapidIO Physical Layer Extension Block Registers (0x00100–0x001FC)

This section specifies the Physical Layer Extension Block Command and Status Register (CSR) set. All registers in the set are 32-bits long and aligned to a 32-bit boundary. These registers allow an external processing element to determine the capabilities, configuration, and status of a processing element using the LP-Serial physical layer. The registers can be accessed using the maintenance operations defined in *Part I: Input/Output Logical Specification*. The register can also be accessed using logical I/O read/write operations defined in *Part I: Input/Output Logical Specification.*

Reads to reserved register addresses return 0, while writes to reserved register addresses complete without error and do not affect the operation of theSREP

Table 168: Register Map for RapidIO Physical Layer Extension Block Registers

Offset	Register Name	See
0x100	SREP_RIO_SW_MB_HEAD	"SREP 1x or 4x RapidIO Port Maintenance Block Header Register"
0x120	SREP_RIO_SW_LT_CTL	"SREP Link Timeout Control CSR"
0x124	SREP_RIO_RESPTO_CTL	"SREP Response Timeout Control CSR"
0x13C	SREP_RIO_PORT_GEN_CTL	"SREP General Control CSR"
0x140	SREP_RIO_PORT_LM_REQ	"SREP Link Maintenance Request CSR"
0x144	SREP_RIO_PORT_LM_RESP	"SREP Link Maintenance Response CSR"
0x148	SREP_RIO_PORT_ACKID_STAT	"SREP Local ackID Status CSR"
0x158	SREP_RIO_ERR_STATUS	"SREP Error and Status CSR"
0x15C	SREP_RIO_CTL	"SREP Port Control CSR"

26.5.1 SREP 1x or 4x RapidIO Port Maintenance Block Header Register

This register contains the block header information.

Register name: SREP_RIO_SW_MB_HEAD Reset value: 0x0200_0002	Register offset: 100

Bits	0	1	2	3	4	5	6	7
00:07	EF_PTR							
08:15	EF_PTR							
16:23	EF_ID							
24:31	EF_ID							

Bits	Name	Description	Type	Reset Value
0:15	EF_PTR	Extended Features Pointer Hard wired pointer to the next block in the features data structure.	R	0x0200
16:31	EF_ID	Hard-wired extended features ID 0x0002 = Endpoint with Software Assisted Error Recovery Option	R	0x0002

26.5.2 SREP Link Timeout Control CSR

This register is not used by the Tsi620 Physical Layer (see "Link Maintenance Functions").

Register name: SREP_RIO_SW_LT_CTL Reset value: 0xFFFF_FF00	Register offset: 120

Bits	0	1	2	3	4	5	6	7
00:07	TVAL							
08:15	TVAL							
16:23	TVAL							
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0:23	TVAL	Timeout Interval Value Timeout = (32/F) * TVAL, where F is the register bus frequency, 100 MHz. When F = 100 MHz, the default value of this register gives a timeout of 5.4 seconds. When TVAL is 0, the timer is disabled.	R/W	0xFFFFFF
24:31	Reserved	N/A	R	0

26.5.3 SREP Response Timeout Control CSR

This register contains the logical layer response timeout timer value for the SREP. This timeout is from transmitting a request packet to receiving a corresponding response packet. The reset value is the maximum timeout interval, and represents between three and six seconds.

Register name: SREP_RIO_RESPTO_CTL **Reset value: 0xFFFF_FF00**	**Register offset: 124**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	TVAL							
08:15	TVAL							
16:23	TVAL							
24:31	Reserved							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:23	TVAL	Timeout Interval Value This timer runs on the Bridge ISF Clock. The timeout period is computed by: 7 * TVAL * (ISF Clock Interval * 4). For a clock frequency of 125 MHz, the maximum timeout period is 4.29 seconds and the granularity is 256 nsec. For a clock frequency of 156.25 MHz, the maximum timeout period is 3.44 seconds and the granularity is 204.8 nsec. Note: If TVAL is programmed to 0, the logical layer response timeout is disabled. Requests wait forever for responses.	R/W	0xFFFFFF
24:31	Reserved	N/A	R	0

26.5.4 SREP General Control CSR

This register is used during system exploration and initialization to

- Identify which device can operate as a host
- Control when a device is allowed to master RapidIO transactions
- Indicate when a device is discovered by a host.

The reset value of the HOST bit is selected through the SP_HOST signal, which is latched after a reset. The reset value of the MAST_EN bit is selected through the SP_MAST_EN signal, which is latched after a reset.

Register name: SREP_RIO_PORT_GEN_CTL **Reset value: Undefined**	**Register offset: 13C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	HOST	MAST_EN	DISC	Reserved				
08:15	Reserved							
16:23	Reserved							
24:31	Reserved							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	HOST	Host Bit A host device is a device that explores, initializes, and maintains a system. Agent or slave devices are initialized by Host devices. 0 = Agent or slave device 1 = Host device	R/W	Undefined
1	MAST_EN	Master Enable The Master Enable bit controls whether or not a device is allowed to issue requests into the system. If the Master Enable is not set the device may only respond to requests. 0b0 = Cannot issue requests 0b1 = Can issue requests	R/W	Undefined
2	DISC	Discovered bit The Tsi620 is located by the processing element that configures the system. 0b1 = Device discovered by system host 0b0 = Device not discovered	R/W	0
3:31	Reserved	N/A	R	0

26.5.5 SREP Link Maintenance Request CSR

This register is for sending a reset or a link-request/input status to the Internal Switch Port (see "Link Maintenance Functions").

Register name: SREP_RIO_PORT_LM_REQ Reset value: 0x0000_0000	Register offset: 140

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							
24:31	Reserved					CMD		

Bits	Name	Description	Type	Reset Value
0:28	Reserved	N/A	R	0
29:31	CMD	Command Command to be sent in the link-request control symbol. If read, this field returns the last written value. 0b011 = Reset. Writing this value causes the device to send four consecutive reset control symbols. 0b100 = Input-status Other values are reserved. Note: The SREP discards requests to transmit reserved command values.	R/W	0

26.5.6 SREP Link Maintenance Response CSR

This register has minimal functionality in the Tsi620 (see "Link Maintenance Functions").

Register name: SREP_RIO_PORT_LM_RESP Reset value: 0x0000_0000	Register offset: 144

Bits	0	1	2	3	4	5	6	7
00:07	RESP_VLD	Reserved						
08:15	Reserved							
16:23	Reserved						ACK_ID_STAT	
24:31	ACK_ID_STAT			LINK_STAT				

Bits	Name	Description	Type	Reset Value
0	RESP_VLD	For more information on the operation of this field, see "Link Maintenance Functions".	RC	0
1:21	Reserved	N/A	R	0
22:26	ACK_ID_STAT	For more information on the operation of this field, see "Link Maintenance Functions".	R	0
27:31	LINK_STAT	For more information on the operation of this field, see "Link Maintenance Functions".	R	0

26.5.7 SREP Local ackID Status CSR

This register has minimal functionality in the Tsi620 (see "Link Maintenance Functions").

Register name: SREP_RIO_PORT_ACKID_STAT Reset value: 0x0000_0000	Register offset: 148

Bits	0	1	2	3	4	5	6	7
00:07	CLR_PKTS	Reserved		INBOUND				
08:15	Reserved							
16:23	Reserved			OUTSTANDING				
24:31	Reserved			OUTBOUND				

Bits	Name	Description	Type	Reset Value
0	CLR_PKTS	This field controls no functionality in SREP. Note that this field always reads as 0.	R/W1S	0
1:2	Reserved	N/A	R	0
3:7	INBOUND	This field controls no functionality in SREP.	R/W	0
8:18	Reserved	N/A	R	0
19:23	OUTSTANDING	This field controls no functionality in SREP.	R	0
24:26	Reserved	N/A	R	0
27:31	OUTBOUND	This field controls no functionality in SREP.	R/W	0

26.5.8 SREP Error and Status CSR

This register contains the port error and status information. This register returns 0x0000001 if it is read when the port is powered down. Only some bits in this register are operational (see "Link Maintenance Functions").

Register name: SREP_RIO_ERR_STATUS Reset value: 0x0000_0002	Register offset: 158

Bits	0	1	2	3	4	5	6	7
00:07	Reserved					OUTPUT_DROP	OUTPUT_FAIL	OUTPUT_DEG
08:15	Reserved			OUTPUT_RE	OUTPUT_R	OUTPUT_RS	OUTPUT_ERR	OUTPUT_ERR_STOP
16:23	Reserved					INPUT_RS	INPUT_ERR	INPUT_ERR_STOP
24:31	Reserved			PORT_W_PEND	Reserved	PORT_ERR	PORT_OK	PORT_UNINIT

Bits	Name	Description	Type	Reset Value
0:4	Reserved	N/A	R	0
5	OUTPUT_DROP	Output port has discarded a packet. The packet is dropped when the "time to live" counter has expired, or when a TEA error has occurred. Write 1 to clear this bit. This bit will never be set.	R/W1C	0
6	OUTPUT_FAIL	Output Failed Encountered Output port encountered a failed condition, meaning that the failed port error threshold is reached in the Port *x* Error Rate Threshold register. Write 1 to clear this bit. This bit will never be set.	R/W1C	0
7	OUTPUT_DEG	OUTPUT_DEG Output Degraded Encountered Output port encountered a degraded condition, meaning that the degraded port error threshold is reached in "SREP Error Rate CSR". Write 1 to clear this bit. This bit will never be set.	R/W1C	0
8:10	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
11	OUTPUT_RE	Output Retry-encountered Outbound port encountered a retry condition. Set when bit 13, Output Retry-stopped, is set. Write 1 to clear this bit. This bit will never be set.	R/W1C	0
12	OUTPUT_R	Output Retried Outbound port received a packet-retry control symbol and cannot make forward progress. This bit is set when bit 13, Output Retry-stopped, is set, and cleared after receiving a packet-accepted or packet-not-accepted control symbol. This bit will never be set.	R	0
13	OUTPUT_RS	Output Retry-stopped Outbound port received a packet-retry control symbol and is in the output retry-stopped state. This bit will never be set.	R	0
14	OUTPUT_ERR	Output Error-encountered Outbound port encountered (and possibly recovered from) a transmission error. This bit is set when bit 15, Output Error-stopped, is set. Write 1 to clear this bit	R/W1C	0
15	OUTPUT_ERR_ STOP	Output Error-stopped Outbound port is in the output error-stopped state.For more information about this bit, see “Link Maintenance Functions”.	R	0
16:20	Reserved	N/A	R	0
21	INPUT_RS	Input Retry-stopped Inbound port is in the input retry-stopped state. This bit will never be set.	R	0
22	INPUT_ERR	Input Error-encountered Inbound port has encountered (and possibly recovered from) a transmission error. This bit is set when bit 23, Input Error-stopped, is set. Write 1 to clear this bit.	R/W1C	0
23	INPUT_ERR_STOP	Input Error-stopped Inbound port is in the input error-stopped state.For more information about this bit, see “Link Maintenance Functions”.	R	0
24:26	Reserved	N/A	R	0
27	PORT_W_PEND	Port-Write Pending Port has encountered a condition that required it to issue an I/O logical port-write maintenance request. Write 1 to clear this bit.	R/W1C	0

(Continued)

Bits	Name	Description	Type	Reset Value
28	Reserved	N/A	R	0
29	PORT_ERR	Port Error Inbound or Outbound port has encountered an error from which the hardware was unable to recover (fatal error). The following fatal errors are included: • Four link-request tries with link-response, but no outstanding ackID • Four link-request tries with timeout error for link-response Write 1 to clear this bit. This bit will never be set.	R/W1C	0
30	PORT_OK	Port OK Inbound and Outbound ports are initialized and can communicate with the adjacent device. This bit and bit 31, Port Un-initialized, are mutually exclusive. For more information about this bit, see “Link Maintenance Functions”.	R	1
31	PORT_UNINIT	Port Un-initialized Inbound and Outbound ports are not initialized. This bit and bit 30, Port OK, are mutually exclusive. This bit is set to a 1 after reset.For more information about this bit, see “Link Maintenance Functions”.	R	0

26.5.9 SREP Port Control CSR

This register returns a default value when read in power-down mode. This register returns 0x0000001 when read while the port is powered down.

Register name: SREP_RIO_CTL **Reset value: 0x5060_0001**	**Register offset: 15C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	PORT_WIDTH		INIT_PWIDTH			OVER_PWIDTH		
08:15	PORT_ DIS	OUTPUT_ EN	INPUT_EN	ERR_DIS	SPARE	Reserved	ENUM_B	Reserved
16:23	Reserved							
24:31	Reserved				STOP_ FAIL_EN	DROP_EN	PORT_ LOCKOUT	PORT_ TYPE

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:1	PORT_WIDTH	Port Width This field determines the port mode after reset. • 00 = Single-lane port - Port is 1x mode only. • 01 = Four-lane port - Port has 1x/4x mode and can operate in 1x or 4x mode The port width of the SREP is always a 1x/4x port.	R	0b01
2:4	INIT_PWIDTH	Initialization Port Width	R	0b010
5:7	OVER_PWIDTH	Override Port Width Software port configuration that overrides the hardware size. • 000 = No override (stay in current operation mode, either 1x or 4x) • 001 = Reserved • 010 = Force single lane, lane 0 • 011 = Force single lane, lane 2 Other values are reserved. This bit does not affect the transfer of packets between the Switch and the Bridge.	R/W	0b000

(Continued)

Bits	Name	Description	Type	Reset Value
8	PORT_DIS	Port Disable 0 = Port receivers/drivers are enabled. 1 = Port receivers/drivers are disabled and are unable to receive/transmit to any packets or control symbols. When the port is disabled, there is no data flow to the output drivers. Transmit drivers of a disabled port transmits all zeros. Any data sent to this port sits in the Output Queue. This bit does affect the transfer of packets between the Switch and the Bridge. It also affects the PORT_OK/PORT_UNINIT bits (see "Link Maintenance Functions").	R/W	0
9	OUTPUT_EN	Output Port Transmit Enable 0 = Port is stopped. It is not able to issue any packets. It can only route and respond to maintenance packets. 1 = Port is enabled to issue any packets. This bit does affect the transfer of packets between the Switch and the Bridge (see "Link Maintenance Functions").	R/W	1
10	INPUT_EN	Inbound Port Enable 0 = Inbound port is stopped and only routes or responds to maintenance requests. Other packets generate packet-not-accepted control symbols to force an error condition on the sending device. 1 = Inbound port responds to any packet. This bit does affect the transfer of packets between the Switch and the Bridge (see "Link Maintenance Functions").	R/W	1
11	ERR_DIS	Error Checking Disable. (Physical layer CRC error only) 0 = Enable error checking and recovery 1 = Disable error checking and recovery When this bit is set, retransmission is suppressed for all packets. Note: If error checking is disabled, then corrupt maintenance packets may be accepted by the Tsi620. Even when error checking is disabled, a corrupt maintenance write request is ignored by the registers. If error checking is enabled, corrupt maintenance packets are not accepted. This bit controls whether or not the SREP performs CRC checks of the packets that it receives from the Switch (see "R2I CRC Error Event").	R/W	0
12	SPARE	This bit controls no functionality within SREP.	R/W	0
13	Reserved	N/A	R	0
14	ENUM_B	Enumeration boundary bit, used in system discovery algorithms. This bit does not control any functionality within the Tsi620.	R/W	0
	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
28	STOP_FAIL_EN	Stop on Port Failed Encountered Enable This bit does not control any functionality in the SREP (see "Link Maintenance Functions").	R/W	0
29	DROP_EN	Drop Packet Enable This bit does not control any functionality in the SREP (see "Link Maintenance Functions").	R/W	0
30	PORT_ LOCKOUT	When cleared, the packets that may be received and issued are controlled by the state of the Output Port Enable and Input Port Enable bits. When set, this port is stopped and is not enabled to issue or receive any packets. The input port can still send and respond to link-requests. According to the *RapidIO Interconnect Specification (Revision 1.3)*, all received packets return packet-not-accepted control symbols to force the sending device to signal an error condition. This is not possible with Port 8, so we set input/output error stopped conditions instead (see "Link Maintenance Functions").	R/W	0
31	PORT_TYPE	Port Type 1 = RapidIO port	R	1

26.6 RapidIO Error Management Extension Registers (0x00200–0x0026C)

This section describes the registers in the Error Management Extended Features block (EF_ID = 0x0007), which is defined in Part 8 of the RapidIO specification. This block allows an external processing element to manage the error status and reporting for a processing element.

When the SREP is powered down, the RapidIO Error Management Extension Registers are read only and return 0.

Reads to reserved register addresses return 0, while writes to reserved register addresses complete without error and do not affect the operation of the SREP.

Reserved bits should not be modified. Reserved register addresses should not be read or written.

All registers are 32-bits and aligned to a 32-bit boundary.

Table 169: Error Management Registers

Port	Offset	Description
All	0x200	General Error Management capability registers
Port	0x240	1x/4x Serial port

Table 170: Register Map for RapidIO Error Management Extension Registers

Offset	Register Name	See
0x200	SREP_RIO_ERR_RPT_BH	“SREP Error Reporting Block Header Register”
0x208	SREP_RIO_LIO_ERR_DET	“SREP Logical and Transport Layer Error Detect CSR”
0x20C	SREP_RIO_LIO_ERR_LOG_EN	“SREP Logical and Transport Layer Error Logging Enable CSR”
0x214	SREP_RIO_LIO_ERR_ADDR	“SREP Logical and Transport Layer Address Capture CSR”
0x218	SREP_RIO_LIO_ERR_DEVID	“SREP Logical and Transport Layer Device ID Capture CSR”
0x21C	SREP_RIO_LIO_ERR_CTRL_INFO	“SREP Logical and Transport Layer Control Capture CSR”
0x228	SREP_RIO_PW_DESTID	“SREP Port-Write Target Device ID CSR”
0x240	SREP_RIO_ERR_DET	“SREP Error Detect CSR”
0x244	SREP_RIO_ERR_RATE_EN	“SREP Error Rate Enable CSR”
0x248	SREP_RIO_ERR_ATTR_CAPT	“SREP Error Capture Attributes CSR”
0x24C	SREP_RIO_ERR_CAPT_0	“SREP Packet Error Capture CSR 0”

Table 170: Register Map for RapidIO Error Management Extension Registers

Offset	Register Name	See
0x250	SREP_RIO_ERR_CAPT_1	"SREP Packet Error Capture CSR 1"
0x254	SREP_RIO_ERR_CAPT_2	"SREP Packet Error Capture CSR 2"
0x258	SREP_RIO_ERR_CAPT_3	"SREP Packet Error Capture CSR 3"
0x268	SREP_RIO_ERR_RATE	"SREP Error Rate CSR"
0x26C	SREP_RIO_ERR_THRESH	"SREP Error Rate Threshold CSR"

26.6.1 SREP Error Reporting Block Header Register

The error reporting block header indicates the start of the Error Management Extensions registers in the Tsi620.

Register name: SREP_RIO_ERR_RPT_BH **Reset value: 0x0000_0007**	**Register offset: 200**

Bits	0	1	2	3	4	5	6	7
00:07	EF_PTR							
08:15	EF_PTR							
16:23	EF_ID							
24:31	EF_ID							

Bits	Name	Description	Type	Reset Value
0:15	EF_PTR	Extended Features Pointer Hard wired pointer to the next block in the data structure. 0000 = Last extended feature block	R	0x0000
16:31	EF_ID	Hard-wired Extended Features ID 0x0007 = EF ID for error management capability	R	0x0007

26.6.2 SREP Logical and Transport Layer Error Detect CSR

This register indicates the error that was detected by the Logical or Transport logic layer. Multiple bits may get set in the register when simultaneous errors are detected during the same clock cycle that the errors are logged.

For more information on the operation of this register, see "RapidIO Logical/Transport Error Information Registers".

Note that events must be enabled in the "SREP Logical and Transport Layer Error Logging Enable CSR" in order for information to be logged for that event, and for notification (either port-write or interrupt) to occur.

Reset values stated are for a powerup reset only. Non-powerup resets do not reset this register.

Register name: SREP_RIO_LIO_ERR_DET **Reset value: 0x0000_0000**	**Register offset: 208**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	L_ERR_RESP	L_DB_ERR_RESP	Reserved		L_ILL_TRANS	L_ILL_TARG	Reserved	L_RESP_TO
08:15	L_UNEXP_RESP	L_UNSUP_TRANS	Reserved					
16:23	Reserved							
24:31	STAT_EVENT	L_ISF_ERR	L_R2I_TTL	L_SPOOF	L_R2I_PERR	L_OOB	L_NO_WR	L_NO_RD

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	L_ERR_RESP	Error Response Bit is set when a response packet with an 'Error' status is received for an outstanding request.	R/WS	0
1	L_DB_ERR_RESP	Doorbell Error Response Bit is set when a response packet with an 'Error' status is received for an outstanding Doorbell request.	R/WS	0
2:3	Reserved	N/A	R	0
4	L_ILL_TRANS	Illegal Transaction Bit is set when a packet was received that has illegal field values for an otherwise supported transaction. This includes data length errors for RapidIO NWRITE, NWRITE_R and SWRITE request, and NREAD responses (see "Logical I/O Packet Events").	R/WS	0

(Continued)

Bits	Name	Description	Type	Reset Value
5	L_ILL_TARG	Illegal Target Decode Bit is set when a packet was received for a destination ID that is not supported by this endpoint (see "8/16-bit Destination ID Support").	R/WS	0
6	Reserved	N/A	R	0
7	L_RESP_TO	Response Timeout Bit is set when no response is received for a request (see "Logical I/O Packet Events").	R/WS	0
8	L_UNEXP_RESP	Unexpected Response Bit is set when a response was received that does not match an outstanding request, or has an amount of data that does not match the corresponding request.	R/WS	0
9	L_UNSUP_TRANS	Unsupported Transaction Bit is set when a packet was received that is an unsupported transaction (see "Logical I/O Packet Events").	R/WS	0
10:23	Reserved	N/A	R	0
24	STAT_EVENT	At least one bit is set in the "SREP R2I Event Status Register" (see "Logical I/O Packet Events"). Note that this bit is set if any bit is set in the "SREP R2I Event Status Register". This bit does not cause error information to be latched in the RapidIO Logical/Transport Error Information registers, and cannot cause a RIO_LOG event to be indicated in the "SREP Interrupt Status Register".	RS	0
25	L_ISF_ERR	An error is detected in the "SREP ISF Logical Error Detect CSR". This bit is cleared when no errors are asserted in the "SREP ISF Logical Error Detect CSR". Control of transmission of port-writes or assertion of interrupts for this error is performed through the ISF Logical Error related registers, starting with "SREP ISF Logical Error Detect CSR". This bit does not cause error information to be latched in the RapidIO Logical/Transport Error Information registers, and cannot cause a RIO_LOG event to be indicated in the "SREP Interrupt Status Register".	RS	0
26	L_R2I_TTL	A packet is buffered for longer than the Time-to-live period programmed in the "SREP R2I Transaction Time-To-Live Register" (see "Logical I/O Packet Events").	R/WS	0
27	L_SPOOF	A response is received whose source ID is not equal to the destination ID of the request (see "Logical I/O Packet Events").	R/WS	0

(Continued)

Bits	Name	Description	Type	Reset Value
28	L_R2I_PERR	A LUT or BAR has detected a parity error (see "Logical I/O Packet Events"). Note: Clearing this bit clears the all bits except LUT_IDX and BAR_IDX in the "SREP R2I BAR and LUT Parity Error Status Register". This bit has special behavior when the registers are locked (see "R2I LUT and BAR Parity Error Information Latching").	R/WS	0
29	L_OOB	A request was received for an address that does not appear in any of the enabled R2I windows (see "Logical I/O Packet Events").	R/WS	0
30	L_NO_WR	A write request was received for an area of memory that does not have write permissions enabled (see "Logical I/O Packet Events").	R/WS	0
31	L_NO_RD	A read request was received for an area of memory that does not have read permissions enabled (see "Logical I/O Packet Events").	R/WS	0

26.6.3 SREP Logical and Transport Layer Error Logging Enable CSR

This register contains the bits that control if an error condition locks the Logical/Transport Layer Error Detect and Capture registers, and is reported to the system host through an interrupt and/or a port-write.

Note that events must be enabled in the "SREP Logical and Transport Layer Error Logging Enable CSR" in order for information to be logged for that event, and for notification (either port-write or interrupt) to occur.

For more information on the operation of this register, see "RapidIO Logical/Transport Error Information Registers". Reset values stated are for a powerup reset only. Non-powerup resets do not reset this register

Register name: SREP_RIO_LIO_ERR_LOG_EN **Reset value: 0x0000_0000**	**Register offset: 20C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	ERR_RESP_EN	DB_ERR_RESP_EN	Reserved		ILL_TRANS_EN	ILL_TARG_EN	Reserved	RESP_TO_EN
08:15	UNEXP_RESP_EN	UNSUP_TRANS_EN	Reserved					
16:23	Reserved							
24:31	Reserved		R2I_TTL_EN	SPOOF_EN	R2I_PERR_EN	OOB_EN	NO_WR_EN	NO_RD_EN

Bits	**Name**	**Description**[a]	**Type**	**Reset Value**
0	ERR_RESP_EN	Error Response Received Logging Enable 0 = disable L_ERR_RESP 1 = enable L_ERR_RESP	R/WS	0
1	DB_ERR_RESP_EN	Doorbell Error Response Received Logging Enable 0 = disable L_DB_ERR_RESP 1 = enable L_DB_ERR_RESP	R/WS	0
2:3	Reserved	N/A	R	0
4	ILL_TRANS_EN	Illegal Transaction Decode Logging Enable 0 = disable L_ILL_TRANS 1 = enable L_ILL_TRANS	R/WS	0
5	ILL_TARG_EN	Illegal Target Decode Enable 0 = disable L_ILL_TARG 1 = enable L_ILL_TARG	R/WS	0
6	Reserved	N/A	R	0

(Continued)

Bits	Name	Description[a]	Type	Reset Value
7	RESP_TO_EN	Response Timeout Logging Enable 0 = disable L_RESP_TO 1 = enable L_RESP_TO	R/WS	0
8	UNEXP_RESP_EN	Unexpected Response Logging Enable 0 = disable L_UNEXP_RESP 1 = enable L_UNEXP_RESP	R/WS	0
9	UNSUP_TRANS_EN	Unsupported Transaction Logging Enable 0 = disable L_UNSUP_TRANS 1 = enable L_UNSUP_TRANS	R/WS	0
10:25	Reserved	N/A	R	0
26	R2I_TTL_EN	R2I Packet Time-to-Live Expired Logging Enable 0 = disable L_R2I_TTL 1 = enable L_R2I_TTL	R/WS	0
27	SPOOF_EN	Response ‘Spoofing’ Check Logging Enable 0 = disable L_SPOOF 1 = enable L_SPOOF	R/WS	0
28	R2I_PERR_EN	R2I BAR/LUT Parity Error Event Logging Enable 0 = disable L_R2I_PERR 1 = enable L_R2I_PERR Note: This bit only controls locking of the “SREP Logical and Transport Layer Error Detect CSR”, not the logging/locking of the “SREP R2I BAR and LUT Parity Error Status Register”.	R/WS	0
29	OOB_EN	Request Out of Bounds Logging Enable 0 = disable L_OOB 1 = enable L_OOB	R/WS	0
30	NO_WR_EN	No Write Permission Event Logging Enable 0 = disable L_NO_WR 1 = enable L_NO_WR	R/WS	0
31	NO_RD_EN	No Read Permission Event Logging Enable 0 = disable L_NO_RD 1 = enable L_NO_RD	R/WS	0

a. All 0 and 1 bits referenced in this register refer to bits enabled in “SREP Logical and Transport Layer Error Detect CSR”.

26.6.4 SREP Logical and Transport Layer Address Capture CSR

This register contains error information. It is locked when a Logical/Transport error is detected, with the exception of L_ISF_ERR, and the corresponding enable bit is set.

For the sourceID where the error originated, see “SREP Logical and Transport Layer Device ID Capture CSR”. For more information on the operation of this register, see “RapidIO Logical/Transport Error Information Registers”.

Reset values stated are for a powerup reset only. Non-powerup resets do not reset this register.

Register name: SREP_RIO_LIO_ERR_ADDR **Reset value: 0x0000_0000**	**Register offset: 214**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	ADDRESS							
08:15	ADDRESS							
16:23	ADDRESS							
24:31	ADDRESS					Reserved	EXT_ADDR	

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:28	ADDRESS	Least significant 29 address bits. For L_SPOOF errors: • Bits 0:15 contain the expected Source ID of the response, if it was a 16-bit Source ID. Bits 16:28 are undefined. • Bits 8:15 contain the expected Source ID of the response, if it was an 8-bit Source ID. Bits 0:7 are 0, and 16:28 are undefined. Note: When this field captures information for a Maintenance Request transaction, the config_offset[0:20] field is placed into ADDRESS[8:28].	R/WS	0
29	Reserved	N/A	R	0
30:31	EXT_ADDR	Extended address bits (most significant 2 bits of address field) Note: For L_SPOOF errors, this field contains the expected TT code of the response.	R/WS	0

26.6.5 SREP Logical and Transport Layer Device ID Capture CSR

This register contains error information, specifically the device ID field values for failed transactions. It is locked when a Logical/Transport error is detected and the corresponding enable bit is set. When the TT field of the erroneous message is not a defined value, the contents of this register are bytes 3 and 4 of the packet received.

For more information on the operation of this register, see "RapidIO Logical/Transport Error Information Registers". Reset values stated are for a powerup reset only. Non-powerup resets do not reset this register.

Register name: SREP_RIO_LIO_ERR_DEVID **Reset value: 0x0000_0000**	**Register offset: 218**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	DESTID_MSB							
08:15	DESTID							
16:23	SRCID_MSB							
24:31	SRCID							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:7	DESTID_MSB	Destination ID Most Significant Byte Most significant byte of the destination ID associated with the error (large transport systems only)	R/WS	0
8:15	DESTID	Destination ID The destination ID associated with the error. Note that this field contains Bridge ISF port information for I2I transactions whose TTL period expires in the R2I queue (see "RapidIO Logical/Transport Error Information Registers").	R/WS	0
16:23	SRCID_MSB	Source ID Most Significant Byte Most significant byte of the source ID associated with the error (large transport systems only)	R/WS	0
24:31	SRCID	Source ID The sourceID associated with the error.	R/WS	0

26.6.6 SREP Logical and Transport Layer Control Capture CSR

This register contains error information, specifically the format type and subtype field values for failed transactions. It is locked when a Logical/Transport error is detected and the corresponding enable bit is set.

For more information on the operation of this register, see "RapidIO Logical/Transport Error Information Registers". Reset values stated are for a powerup reset only. Non-powerup resets do not reset this register.

Register name: SREP_RIO_LIO_ERR_CTRL_INFO **Reset value: 0x0000_0000**	**Register offset: 21C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	FTYPE				TTYPE			
08:15	Reserved							
16:23	STAT_SIZE				WDPTR	CRF	PRIO	
24:31	TT		RESP_SIZE					

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:3	FTYPE	RapidIO Format type associated with the error	R/WS	0
4:7	TTYPE	RapidIO Transaction type associated with the error	R/WS	0
8:15	Reserved	N/A	R	0
16:19	STAT_SIZE	This field contains the Status value for a response packets, and Size field for Request packets. For unknown packets, this field contains the least significant 4 bits of the byte immediately following the source ID of the packet. Note that for Spoof Response events, STAT_SIZE and WDPTR indicate the amount of data expected/in the original request.	R/WS	0
20	WDPTR	This field contains the WDPTR value for request packets. It is undefined for Response packets. Note that for Spoof Response events, STAT_SIZE and WDPTR indicate the amount of data expected/in the original request.	R/WS	0
21	CRF	RapidIO Critical Request Flow bit for packet associated with the error.	R/WS	0
22:23	PRIO	RapidIO physical layer priority for packet associated with the error.	R/WS	0
24:25	TT	RapidIO TT Code for packet associated with the error.	R/WS	0

(Continued)

Bits	Name	Description	Type	Reset Value
26:31	RESP_SIZE	Response Size This field is useful for determining the cause of a RapidIO 'Spoof Response' event. The amount of data, in multiples of 8 bytes, which was received in a response. 0x00 = No data in the response 0x01 = 1 to 8 bytes of data in the response ... 0x20 = 256 bytes of data in the response 0x21 = More than 256 bytes of data was received 0x22 = 0x3F - Reserved For more information, see Table 51.	R/WS	0

26.6.7 SREP Port-Write Target Device ID CSR

This register contains the target device ID to be used when a device generates a Maintenance Port-write operation to report errors to a system host.

Register name: SREP_RIO_PW_DESTID Reset value: 0x0000_0000	Register offset: 228

Bits	0	1	2	3	4	5	6	7
00:07	DESTID_MSB							
08:15	DESTID_LSB							
16:23	LARGE_ DESTID	Reserved						
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0:7	DESTID_MSB	Most Significant Byte of Port-Write Target Device ID. Used only when LARGE_DESTID is 1.	R/W	0
8:15	DESTID_LSB	If LARGE_DESTID is 0, the DESTID_LSB field is the 8-bit DESTID used in Port-Write requests generated by the SREP. If LARGE_DESTID is 1, the DESTID_LSB field forms the least significant bits of a 16-bit DestID used in Port-Write requests generated by the SREP.	R/W	0
16	LARGE_DESTID	0 = Port-write transactions are generated with an 8-bit destination ID. 1 = Port-write transactions are generated with a 16-bit destination ID.	R/W	0
17:31	Reserved	N/A	R	0

26.6.8 SREP Error Detect CSR

This register is not used by the Tsi620. The SREP only detects a single implementation-specific physical layer error. For more information, see "Physical Layer Events".

Register name: SREP_RIO_ERR_DET Reset value: 0x0000_0000	Register offset: 240

Bits	0	1	2	3	4	5	6	7
00:07	IMP_ SPEC_ ERR	Reserved						
08:15	Reserved	CS_CRC_ ERR	CS_ILL_ID	CS_NOT_ ACC	PKT_ILL_A CKID	PKT_CRC_ ERR	PKT_ILL_ SIZE	Reserved
16:23	Reserved							
24:31	Reserved		LR_ACKID _ILL	PROT_ ERR	Reserved	DELIN_ ERR	CS_ACK_ ILL	LINK_TO

Bits	Name	Description	Type	Reset Value
0	IMP_SPEC_ERR	Implementation Specific Error The SREP has one implementation-specific event, LINK_INIT_NOTIFICATION.	R	0
1:8	Reserved	N/A	R	0
9	CS_CRC_ERR	Received a control symbol with a CRC error. This error does not exist for the SREP.	R	0
10	CS_ILL_ID	Received an acknowledge control symbol with an unexpected ackID (packet-accepted, packet-not accepted or packet_retry). The Capture register does not have valid information during this error detection. This error does not exist for the SREP.	R	0
11	CS_NOT_ACC	Received packet-not-accepted control symbol. This error does not exist for the SREP.	R	0
12	PKT_ILL_ACKID	Received packet with unexpected ackID This error does not exist for the SREP.	R	0
13	PKT_CRC_ERR	Received a packet with a CRC error This error does not exist for the SREP.	R	0
14	PKT_ILL_SIZE	Received packet exceeds 276 bytes.This error does not exist for the SREP.	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
15:25	Reserved	N/A	R	0
26	LR_ACKID_ILL	Link response received with an ackID that is not outstanding. The Capture register does not have valid information during this error detection. This error does not exist for the SREP.	R	0
27	PROT_ERR	Protocol Error Received control symbol is unexpected This error does not exist for the SREP.	R	0
28	Reserved	N/A	R	0
29	DELIN_ERR	Delineation Error Received unaligned /SC/ or /PD/, or undefined code-group. The Capture register does not capture information for this error. This error does not exist for the SREP.	R	0
30	CS_ACK_ILL	Received an unexpected acknowledge control symbol. This error does not exist for the SREP.	R	0
31	LINK_TO	An acknowledge or Link-response is not received within the specified timeout interval, see the "SREP Link Timeout Control CSR". The Capture register does not capture information for this error. This error does not exist for the SREP.	R	0

26.6.9 SREP Error Rate Enable CSR

This register controls no functionality in the Tsi620.

Register name: SREP_RIO_ERR_RATE_EN Reset value: 0x0000_0000	Register offset: 244

Bits	0	1	2	3	4	5	6	7
00:07	IMP_SPEC_ERR	Reserved						
08:15	Reserved	CS_CRC_ERR_EN	CS_ILL_ID_EN	CS_NOT_ACC_EN	PKT_ILL_ACKID_EN	PKT_CRC_ERR_EN	PKT_ILL_SIZE_EN	Reserved
16:23	Reserved							
24:31	Reserved		LR_ACKID_ILL_EN	PROT_ERR_EN	Reserved	DELIN_ERR_EN	CS_ACK_ILL_EN	LINK_TO_EN

Bits	Name	Description	Type	Reset Value
0	IMP_SPEC_ERR	Logical /Transport Error Enable For information on the errors applicable to this field, see "SREP Error Detect CSR".	R	0
1:8	Reserved	N/A	R	0
9	CS_CRC_ERR_EN	Enable error rate counting. Received Control Symbol with a CRC error. This error does not exist for the SREP.	R	0
10	CS_ILL_ID_EN	Enable error rate counting. Received an acknowledge control symbol with an unexpected ackID (packet-accepted or packet_retry). This error does not exist for the SREP.	R	0
11	CS_NOT_ACC_EN	Enable error rate counting. Received packet-not-accepted control symbol. This error does not exist for the SREP.	R	0
12	PKT_ILL_ACKID_EN	Enable error rate counting. Received packet with not unexpected ackID. This error does not exist for the SREP.	R	0
13	PKT_CRC_ERR_EN	Enable error rate counting. Received packet with a CRC error.This error does not exist for the SREP.	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
14	PKT_ILL_SIZE_EN	Enable error rate counting. Received packet exceeds 276 bytes. This error does not exist for the SREP.	R	0
15:25	Reserved	N/A	R	0
26	LR_ACKID_ILL_EN	Enable error rate counting. A received Link Response control symbol contains an ackID that is not outstanding. This error does not exist for the SREP.	R	0
27	PROT_ERR_EN	Enable error rate counting Protocol Error Received Control Symbol is unexpected. This error does not exist for the SREP.	R	0
28	Reserved	N/A	R	0
29	DELIN_ERR_EN	Enable error rate counting Delineation Error Received unaligned /SC/or/PD/ or undefined code-group. This error does not exist for the SREP.	R	0
30	CS_ACK_ILL_EN	Enable error rate counting An unexpected acknowledge control symbol was received. This error does not exist for the SREP.	R	0
31	LINK_TO_EN	Enable error rate counting An acknowledge or Link-response is not received within the specified timeout interval. This error does not exist for the SREP.	R	0

26.6.10 SREP Error Capture Attributes CSR

This register controls no functionality in the Tsi620.

Register name: SREP_RIO_ERR_ATTR_CAPT Reset value: 0x0000_0000	Register offset: 248

Bits	0	1	2	3	4	5	6	7
00:07	INFO_TYPE		Reserved	ERR_TYPE				
08:15	Reserved							
16:23	Reserved							
24:31	Reserved							VAL_CAPT

Bits	Name	Description	Type	Reset Value
0:1	INFO_TYPE	Type of information logged. • 00 = Packet • 01 = Control Symbol and unaligned /SC/or/PD/ or undefined code-group • 10 = Implementation specific (capture register contents are implementation-specific to report implementation-specific errors) • 11 = Reserved for RapidIO ports	R	0
2	Reserved	N/A	R	0
3:7	ERR_TYPE	Encoded 5-bit value of captured error bit in the "SREP Error Detect CSR". • 00000 = bit 0 (IMP_SPEC_ERR) • 00001 = bit 1 (reserved) • .... • 00111 = bit 8 (reserved) • 01000 = bit 9 (CS_CRC_ERR) • 01001 = bit 10 (CS_ILL_ID) • 01010 = bit 12 (CS_NOT_ACC) •	R	0
8:30	Reserved	N/A	R	0
31	VAL_CAPT	Capture Valid Information	R	0

26.6.11 SREP Packet Error Capture CSR 0

This register controls no functionality in the Tsi620.

Register name: SREP_RIO_ERR_CAPT_0 Reset value: 0x0000_0000	Register offset: 24C

Bits	0	1	2	3	4	5	6	7
00:7	CAPT_0[0:7]							
8:15	CAPT_0[8:15]							
16:23	CAPT_0[16:23]							
24:31	CAPT_0[24:31]							

Bits	Name	Description	Type	Reset Value
0:31	CAPT_0	No functionality	R	0

26.6.12 SREP Packet Error Capture CSR 1

This register controls no functionality in the Tsi620.

Register name: SREP_RIO_ERR_CAPT_1 Reset value: 0x0000_0000	Register offset: 250

Bits	0	1	2	3	4	5	6	7
00:7	CAPT_1[0:7]							
8:15	CAPT_1[8:15]							
16:23	CAPT_1[16:23]							
24:31	CAPT_1[24:31]							

Bits	Name	Description	Type	Reset Value
0:31	CAPT_1	Writeable by software for test purposes.	R	0

26.6.13 SREP Packet Error Capture CSR 2

This register controls no functionality in the Tsi620.

Register name: SREP_RIO_ERR_CAPT_2 Reset value: 0x0000_0000	Register offset: 254

Bits	0	1	2	3	4	5	6	7
0:7	CAPT_2[0:7]							
8:15	CAPT_2[8:15]							
16:23	CAPT_2[16:23]							
24:31	CAPT_2[24:31]							

Bits	Name	Description	Type	Reset Value
0:31	CAPT_2	Writeable by software for test purposes.	R	0

26.6.14 SREP Packet Error Capture CSR 3

This register controls no functionality in the Tsi620.

Register name: SREP_RIO_ERR_CAPT_3 Reset value: 0x0000_0000	Register offset: 258

Bits	0	1	2	3	4	5	6	7
0:7	CAPT_3[0:7]							
8:15	CAPT_3[8:15]							
16:23	CAPT_3[16:23]							
24:31	CAPT_3[24:31]							

Bits	Name	Description	Type	Reset Value
0:31	CAPT_3	Writeable by software for test purposes.	R	0

26.6.15 SREP Error Rate CSR

This register controls no functionality in the Tsi620.

Register name: SREP_RIO_ERR_RATE Reset value: 0x0000_0000	Register offset: 268

Bits	0	1	2	3	4	5	6	7
00:07	ERR_RB							
08:15	Reserved						ERR_RR	
16:23	PEAK							
24:31	ERR_RATE_CNT							

Bits	Name	Description	Type	Reset Value
0:7	ERR_RB	The Error Rate Bias value.	R	0x00
8:13	Reserved	N/A	R	0
14:15	ERR_RR	Error Rate Recovery	R	0
16:23	PEAK	The maximum value attained by the error rate counter.	R	0
24:31	ERR_RATE_CNT	Error Rate Counter	R	0

26.6.16 SREP Error Rate Threshold CSR

This register controls no functionality in the Tsi620.

Register name: SREP_RIO_ERR_THRESH Reset value: 0x0000_0000	Register offset: 26C

Bits	0	1	2	3	4	5	6	7
00:07	ERR_RFT							
08:15	ERR_RDT							
16:23	Reserved							
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0:7	ERR_RFT	Error Rate Failed Threshold	R	0x00
8:15	ERR_RDT	Error Rate Degraded Threshold	R	0x00
16:31	Reserved	N/A	R	0

26.7 RapidIO Register Block Description Registers

The registers in this block describe the IDT Implementation-specific register blocks in the remainder of the design. A value of ‘0’ for any of these registers means that they are not valid for this design. The register block description registers occur after the last RapidIO standard register in a device.

The purpose of these registers is to allow customers to define device drivers for subsections of the register space presented by the SREP. This allows customers to invest in software for IDT devices with the confidence that IDT will allow them to leverage that investment through register interface reuse in other IDT devices.

The registers are designed as a linked list of descriptors to blocks of IDT registers. The linked list starts with a pointer to the first IDT block of registers, with an indication of validity.

Table 171: Register Map for RapidIO Register Block Description Registers

Offset	Register Name	See
0x270	SREP_TUN_FIRST_BLOCK_ADDR	“SREP IDT First Register Block Address”
0x274	SREP_TUN_PHYS_BLK_HDR	“SREP IDT Physical Layer Registers Block”
0x278	SREP_TUN_TRANS_HDR	“SREP IDT Transport Layer Registers Block”
0x27C	SREP_TUN_LOG_HDR	“SREP IDT Logical Layer Registers Block”
0x280	SREP_TUN_STATS_HDR	“SREP IDT Statistics Registers Block”
0x284	SREP_TUN_EVENT_NOT_BLK	“SREP IDT Event Notification Registers Block”
0x288	SREP_TUN_RIO_MAC_HDR	“SREP IDT Physical Layer MAC Registers Block”

26.7.1 SREP IDT First Register Block Address

This register gives the offset in bytes, from the address of this register, at which the first IDT register block starts.

Register name: SREP_TUN_FIRST_BLOCK_ADDR Reset value: 0x8001_0090	Register offset: 270

Bits	0	1	2	3	4	5	6	7
00:07	VALID	OFFSET						
08:15	OFFSET							
16:23	OFFSET							
24:31	OFFSET							

Bits	Name	Description	Type	Reset Value
0	VALID	Indicates that this register is valid.	R	1
1:31	OFFSET	Offset, in bytes, from the address of this register, at which the first IDT Register block starts. Resulting address must be a multiple of 256 bytes. The first register block starts at address 0x10300. Note that this is the RapidIO Maintenance space offset.	R	0x010090

26.7.2 SREP IDT Physical Layer Registers Block

This register indicates the presence and type of RapidIO Physical Layer Registers in the device. The registers described by this block are located in "IDT Specific Registers – Physical Layer".

Register name: SREP_TUN_PHYS_BLK_HDR Reset value: 0x0100_0001	Register offset: 274

Bits	0	1	2	3	4	5	6	7
00:07	BLOCK_TYPE							
08:15	BLOCK_SIZE							
16:23	OFFSET_NEXT							
24:31	OFFSET_NEXT							

Bits	Name	Description	Type	Reset Value
0:7	BLOCK_TYPE	IDT specific block type 0x01 = Physical Layer Registers Block	R	0x01
8:15	BLOCK_SIZE	Describes the number of different instances of registers in a block-specific way. 0 = One, 4x/1x, RapidIO Interface	R	0x00
16:31	OFFSET_NEXT	The offset, in units of 0x100 bytes, from this register to the next IDT specific register header block. 0 is not a legal value.	R	0x01

26.7.3 SREP IDT Transport Layer Registers Block

This register indicates the presence and type of RapidIO Transport Layer Registers in the device. The registers described by this block are located in "IDT Specific Registers – Transport Layer".

Register name: SREP_TUN_TRANS_HDR **Reset value: 0x0200_0001**	**Register offset: 278**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	BLOCK_TYPE							
08:15	BLOCK_SIZE							
16:23	OFFSET_NEXT							
24:31	OFFSET_NEXT							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:7	BLOCK_TYPE	IDT specific block type 0x02 = RapidIO Transport Layer Registers	R	0x02
8:15	BLOCK_SIZE	Describes the number of different instances of registers in a block-specific way. 0 = One Bridge ISF Interface and one RapidIO Interface	R	0x00
16:31	OFFSET_NEXT	The offset, in units of 0x100 bytes, from this register to the next IDT specific register header block.	R	0x01

26.7.4 SREP IDT Logical Layer Registers Block

This register indicates the presence and type of RapidIO Logical Layer Registers in the device. The registers described by this block are located in "IDT Specific Registers – Logical Layer".

Register name: SREP_TUN_LOG_HDR **Reset value: 0x0300_0003**	**Register offset: 27C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	BLOCK_TYPE							
08:15	BLOCK_SIZE							
16:23	OFFSET_NEXT							
24:31	OFFSET_NEXT							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:7	BLOCK_TYPE	IDT specific block type 0x03 = IDT Logical Layer Registers	R	0x03
8:15	BLOCK_SIZE	Describes the number of different instances of registers in a block-specific way. 0 = One Bridge ISF Interface and one RapidIO Interface	R	0x00
16:31	OFFSET_NEXT	The offset, in units of 0x100 bytes, from this register to the next IDT specific register header block.	R	0x03

26.7.5 SREP IDT Statistics Registers Block

This register indicates the presence and type of RapidIO Statistics Gathering Registers in the device. The registers described by this block are located in “IDT Specific Registers – Statistics”

Register name: SREP_TUN_STATS_HDR **Reset value: 0x0400_0001**	**Register offset: 280**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	BLOCK_TYPE							
08:15	BLOCK_SIZE							
16:23	OFFSET_NEXT							
24:31	OFFSET_NEXT							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:7	BLOCK_TYPE	IDT specific block type 0x04 = RapidIO Statistics Gathering Registers	R	0x04
8:15	BLOCK_SIZE	Describes the number of different instances of registers in a block-specific way. 0 = One Bridge ISF Interface and one RapidIO Interface	R	0x00
16:31	OFFSET_NEXT	The offset, in units of 0x100 bytes, from the Statistics Gathering register block to the next IDT specific register header block.	R	0x01

26.7.6 SREP IDT Event Notification Registers Block

This register indicates the presence and type of RapidIO Event Notification Registers in the device. The registers described by this block are located in "IDT Specific Registers – Event Notification".

Register name: SREP_TUN_EVENT_NOT_BLK **Reset value: 0x0500_0001**	**Register offset: 284**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	BLOCK_TYPE							
08:15	BLOCK_SIZE							
16:23	OFFSET_NEXT							
24:31	OFFSET_NEXT							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:7	BLOCK_TYPE	IDT specific block type 3 = Event Notification Registers	R	0x05
8:15	BLOCK_SIZE	Describes the number of different instances of registers in a block-specific way. 0 = One Bridge ISF Interface and one RapidIO Interface	R	0x00
16:31	OFFSET_NEXT	The offset, in units of 0x100 bytes, from this register to the next IDT specific register header block.	R	0x01

26.7.7 SREP IDT Physical Layer MAC Registers Block

This register indicates the presence and type of RapidIO MAC registers in the device. The registers described by this block are located in "IDT Specific Registers – Electrical Layer".

Register name: SREP_TUN_RIO_MAC_HDR **Reset value: 0x0600_0001**	**Register offset: 288**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	BLOCK_TYPE							
08:15	BLOCK_SIZE							
16:23	OFFSET_NEXT							
24:31	OFFSET_NEXT							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:7	BLOCK_TYPE	IDT specific block type 6 = RapidIO MAC Header Registers	R	0x06
8:15	BLOCK_SIZE	Describes the number of different instances of registers in a block-specific way. 0 = One Bridge ISF Interface and one RapidIO Interface	R	0x00
16:31	OFFSET_NEXT	The offset, in units of 0x100 bytes, from this register to the next IDT specific register header block.	R	0x01

26.8 IDT Specific Registers – Physical Layer

The registers in this section are vendor specific. They are not defined in the *RapidIO Interconnect Specification (Revision 1.3)*. They are specific to IDT's design of a RapidIO endpoint.

Reserved bits should not be modified. Reserved register addresses should not be read or written.

These registers are reset when the SREP is reset. The registers are not accessible when the SREP is in reset or powered down.

Table 172: Register Map for SREP IDT Specific Registers – Physical Layer

Offset	Register Name	See
0x340	SREP_MODE	"SREP Mode CSR"
0x344	SREP_CTL_INDEP	"SREP Control Independent Register"
0x34C	SREP_SEND_MCS	"SREP Send Multicast-Event Control Symbol Register"
0x358	SREP_INT_STATUS	"SREP Interrupt Status Register"
0x35C	SREP_INT_GEN	"SREP Interrupt Generate Register"
0x6FC	SREP_B2S_BREL	"SREP B2S Buffer Release Control Register"
0x390	SREP_SCRATCH{0..1}	"SREP Scratch n Register"

26.8.1 SREP Mode CSR

This register defines whether receipt of a reset request from the link partner causes a reset or an interrupt for the SREP.

Register name: SREP_MODE Reset value: 0x0200_0000	Register offset: 340

Bits	0	1	2	3	4	5	6	7
00:07	Reserved						SELF_RST	Reserved
08:15	Reserved							
16:23	Reserved							
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0:5	Reserved	Reserved	R	0
6	SELF_RST	Self Reset Enable After four link-request reset control symbols are accepted, the device either resets itself or raises an interrupt, according to the value in this register field (see “Bridge Reset”). 0 = Disable: RST_IRQ_b signal is asserted 1 = Enable: Device is reset	R/W	1
7:31	Reserved	Reserved	R	0

26.8.2 SREP Control Independent Register

This register is used for error reporting control (see “Physical Layer Events”).

Register name: SREP_CTL_INDEP Reset value: 0x0000_0000	Register offset: 344

Bits	0	1	2	3	4	5	6	7
00:07	Reserved		SCRATCH	Reserved		FORCE_ REINIT	Reserved	
08:15	Reserved						LINK_INIT_ NOTIFICAT ION_EN	Reserved
16:23	Reserved							
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0:1	Reserved	N/A	R	0
2	SCRATCH	This bit controls no functionality. It is a read/write ‘scratch pad’ bit for software use.	R/W	0
3:4	Reserved	N/A	R	0
5	FORCE_ REINIT	Force link re-initialization process. This bit is active on write and automatically returns to “0”.	R/W1S	0
6:13	Reserved	N/A	R	0
14	LINK_INIT_NOTIF ICATION_EN	Enables interrupts and port writes for LINK_INIT_NOTIFICATION events. 0 = Event disabled 1 = Event enabled For more information, see “Link Maintenance Functions”.	R/W	0
15:31	Reserved	N/A	R	0

26.8.3 SREP Send Multicast-Event Control Symbol Register

When this register is written, it causes a Multicast-Event control symbol to be sent to the Internal Switch Port (see "Multicast-event Control Symbols").

Register name: SREP_SEND_MCS Reset value: 0x0000_0002	Register offset: 34C

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							
24:31	Reserved						DONE	SEND

Bits	Name	Description	Type	Reset Value
0:29	Reserved	N/A	R	0
30	DONE	The Tsi620 sets this field to 0 when system software sets SEND to 1. The Tsi620 sets this field to 1 once it has sent the Multicast-Event control symbol. Note that the ability to send and MECS is contingent upon the links ability to send control symbols (see "Multicast-event Control Symbols"). A value of 1 in this field indicates that the Tsi620 is ready to send another Multicast-Event control symbol.	R	1
31	SEND	Write 1 to send a multicast-event control symbol when DONE = 1.	R/W1S	0

26.8.4 SREP Interrupt Status Register

For more information on the use of this register, see "Physical Layer Events".

Register name: SREP_INT_STATUS Reset value: 0x0000_0000	Register offset: 358

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved						LINK_INIT_NOTIFICATION	Reserved
16:23	Reserved							
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0:13	Reserved	N/A Bit 13 = MC_TEA	R	0
14	LINK_INIT_NOTIFICATION	Link Initialization Notification Once set, the LINK_INIT_NOTIFICATION bit is cleared by writing 1 to it. When the PORT_LOCKOUT bit is set in "RapidIO Serial Port x Control CSR" RIO Serial Port x Control CSR, and a link has initialized according to the PORT_OK bit in "RapidIO Port x Error and Status CSR", the LINK_INIT_NOTIFICATION is set to 1. To stop the LINK_INIT_NOTIFICATION bit from getting set, PORT_LOCKOUT must be set to 0 and/or the link must no longer be in an initialized state.	R/W1C	0
15:31	Reserved	N/A	R	0

26.8.5 SREP Interrupt Generate Register

For more information on the use of this register, see “Physical Layer Events”.

Register name: SREP_INT_GEN Reset value: 0x0000_0000	Register offset: 35C

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved						LINK_INIT_NOTIFICATION_GEN	Reserved
16:23	Reserved							
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0:13	Reserved	N/A	R	0
14	LINK_INIT_NOTIFICATION_GEN	Force the LINK_INIT_NOTIFICATION bit in the“SREP Interrupt Status Register” to be set. Bit always reads as zero.	R/W1S	0
15:31	Reserved	N/A	R	0

26.8.6 SREP Scratch n Register

Where n is 0 or 1. The scratch registers are not linked to any functionality in the SREP. The scratch registers may be used for any purpose in a system.

Register name: SREP_SCRATCH{0..1} **Reset value: 0x0000_0000**	**Register offset: 390, 394**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	SCRATCH							
08:15	SCRATCH							
16:23	SCRATCH							
24:31	SCRATCH							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:31	SCRATCH	Scratch pad value.	R/W	0

26.9 IDT Specific Registers – Transport Layer

The registers in this section are vendor specific. They are not defined in the *RapidIO Interconnect Specification (Revision 1.3)*. They are specific to IDT's design of a RapidIO endpoint.

Reserved bits should not be modified. Reserved register addresses should not be read or written.

These registers are reset when the SREP is reset. The registers are not accessible when the RapidIO port is in reset or powered down.

Table 173: Register Map for SREP IDT Specific Registers – Transport Layer

Offset	Register Name	See
0x400	SREP_DESTID_CHK_CTL	"SREP Destination ID Checking Control Register"
0x404	SREP_DESTID_LG_CTL	"SREP Large Secondary Destination ID Checking Control Register"
0x408	SREP_REG_ACC_CTL	"SREP Register Access Source ID Checking Control Register"
0x40C	SREP_REG_ACC_SMCHK_CTL	"SREP Register Access Small Source ID Checking Control Register"
0x410	SREP_REG_ACC_LRGCHK_CTL	"SREP Large Register Access Source ID Checking Register"

26.9.1 SREP Destination ID Checking Control Register

This register controls whether or not the destination IDs of transactions received by the SREP are checked against the destination IDs programmed in the "SREP Base Device ID CSR".

Register name: SREP_DESTID_CHK_CTL **Reset value: 0x0000_0000**	**Register offset: 400**

Bits	0	1	2	3	4	5	6	7
00:07	SMC_MASK							
08:15	SMC_VALUE							
16:23	Reserved							
24:31	Reserved		LMC_EN	SMC_EN	LG_TT_EN		SM_TT_EN	

Bits	Name	Description	Type	Reset Value
0:7	SMC_MASK	Mask for 8-bit destination IDs that can be accepted (see "Secondary Destination ID Management"). When a bit is 1, the 8-bit destination ID of a received packet is compared with the corresponding bit in SMC_VALUE. When a bit is 0, the corresponding bit in SMC_VALUE is not checked.	R/W	0
8:15	SMC_VALUE	Value that an 8-bit destination ID must match in order for the packet to be accepted (see "Secondary Destination ID Management").	R/W	0
16:25	Reserved	N/A	R	0
26	LMC_EN	Enable checking of 16-bit destination IDs using the values in the "SREP Large Secondary Destination ID Checking Control Register" (see "Secondary Destination ID Management").	R/W	0
27	SMC_EN	Enable checking of 8-bit destination IDs using the values in SMC_MASK and SMC_VALUE (see "Secondary Destination ID Management").	R/W	0
28:29	LG_TT_EN	Enable checking of 16-bit destination IDs using the value in the LG_DESTID field of the "SREP Base Device ID CSR". 0b00 = Accept all packets with 16-bit destination IDs. Ignore the settings in "SREP Large Secondary Destination ID Checking Control Register" 0b01 = Do not accept any packets with 16-bit destination IDs. Ignore the settings in "SREP Large Secondary Destination ID Checking Control Register" 0b10 = Accept packets whose 16-bit destination ID matches LG_DEST_ID in "SREP Base Device ID CSR" 0b11 = Reserved, behaves as if value was 0b01.	R/W	0
30:31	SM_TT_EN	Enable checking of 8-bit destination IDs using the value in the DEST_ID field of the "SREP Base Device ID CSR". 0b00 = Accept all packets with 8-bit destination IDs. Ignore the values of SMC_MASK and SMC_VALUE. 0b01 = Do not accept any packets with 8-bit destination IDs. Ignore the values of SMC_MASK and SMC_VALUE. 0b10 = Accept packets whose 8-bit destination ID matches DEST_ID in "SREP Base Device ID CSR" 0b11 = Reserved (do not accept any packets with 8-bit destination IDs)	R/W	0

26.9.2 SREP Large Secondary Destination ID Checking Control Register

This register controls whether or not the destination IDs of transactions received by the SREP are checked against the destination IDs programmed in the "SREP Base Device ID CSR".

Register name: SREP_DESTID_LG_CTL Reset value: 0x0000_0000	Register offset: 404

Bits	0	1	2	3	4	5	6	7
00:07	LMC_MASK							
08:15	LMC_MASK							
16:23	LMC_VALUE							
24:31	LMC_VALUE							

Bits	Name	Description	Type	Reset Value
0:15	LMC_MASK	Mask for 16-bit destination IDs that can be accepted (see "Secondary Destination ID Management"). When a bit is 1, the 16-bit destination ID of a received packet is compared with the corresponding bit in LMC_VALUE. When a bit is 0, the corresponding bit in LMC_VALUE is not checked.	R/W	0
16:31	LMC_VALUE	Value that a 16-bit destination ID must match in order for the packet to be accepted (see "Secondary Destination ID Management").	R/W	0

26.9.3 SREP Register Access Source ID Checking Control Register

This register controls which 8 and/or 16-bit source IDs are accepted as valid sources of register write accesses. Note that this register is writable from any source.

Register name: SREP_REG_ACC_CTL Reset value: 0x4138_4136	Register offset: 408

Bits	0	1	2	3	4	5	6	7
00:07	SREG_CTL							
08:15	SREG_CTL							
16:23	LREG_CTL							
24:31	LREG_CTL							

Bits	Name	Description	Type	Reset Value
0:15	SREG_CTL	There are three valid values for SREG_CTL: ‘L8’ (0x4C38) - Lock out all 8-bit source IDs from writing to registers ‘A8’(0x4138) - All 8-bit source IDs can write to registers ‘S8’ (0x5338) - Some 8-bit source IDs are accepted based on the values of SREG_MASK and SREG_VAL in “SREP Register Access Small Source ID Checking Control Register”. Note: Writing any other value to this field generates a value of ‘L8’.	R/W	0x4138
16:31	LREG_CTL	There are three valid values for LREG_CTL: ‘L6’ (0x4C36) - Lock out all 16-bit source IDs from writing to registers ‘A6’(0x4136) - All 16-bit source IDs can write to registers ‘S6’ (0x5336) - Some 16-bit source IDs are accepted based on the values of LREG_MASK and LREG_VAL in “SREP Large Register Access Source ID Checking Register”. Note: Writing any other value to this field generates a value of ‘L6’.	R/W	0x4136

26.9.4 SREP Register Access Small Source ID Checking Control Register

This register determines which 8-bit source IDs of transactions received by the SREP are allowed to perform register write accesses. The interpretation/use of this register is modified by the "SREP Register Access Source ID Checking Control Register".

Register name: SREP_REG_ACC_SMCHK_CTL **Reset value: 0x0000_0000**	**Register offset: 40C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	SREG_MASK							
08:15	SREG_VALUE							
16:23	LAST_SRCID							
24:31	LAST_SRCID							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:7	SREG_MASK	Mask for 8-bit source IDs that can be accepted for register write accesses (see "Register Access Source ID Management").	R/W	0
8:15	SREG_VALUE	Value that an 8-bit source ID must match in order for the register write access packet to be accepted (see "Register Access Source ID Management").	R/W	0
16:31	LAST_SRCID	The source ID of the last transaction to write to the "SREP Register Access Source ID Checking Control Register". 8-bit destination IDs are zero extended to make a 16-bit value. If the "SREP Register Access Small Source ID Checking Control Register" is written from the register bus, this field is not affected.	R	0

26.9.5 SREP Large Register Access Source ID Checking Register

This register determines which 16-bit source IDs of transactions received by the SREP are allowed to perform register write accesses. The interpretation/use of this register is modified by the "SREP Register Access Source ID Checking Control Register"

Register name: SREP_REG_ACC_LRGCHK_CTL **Reset value: 0x0000_0000**	**Register offset: 410**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	LREG_MASK							
08:15	LREG_MASK							
16:23	LREG_VALUE							
24:31	LREG_VALUE							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:15	LREG_ MASK	Mask for 16-bit source IDs that can be accepted for register write accesses (see "Register Access Source ID Management").	R/W	0
16:31	LREG_ VALUE	Value that a 16-bit source ID must match in order for the register write access to be accepted (see "Register Access Source ID Management").	R/W	0

26.10 IDT Specific Registers – Logical Layer

The registers in this section are vendor specific. They are not defined in the *RapidIO Interconnect Specification (Revision 1.3)*. They are specific to IDT's design of a RapidIO endpoint.

Reserved bits should not be modified. Reserved register addresses should not be read or written.

These registers are reset when the SREP is reset. The registers are not accessible when the RapidIO port is in reset or powered down.

Table 174: Register Map for SREP IDT Specific Registers – Transport Layer

Offset	Register Name	See
0x500	SREP_R2I_BAR{0..7}_LUT_CSR	"SREP R2I Base Address Register x LUT Control CSR"
0x508	SREP_R2I_BAR{0..7}_LOWER	"SREP R2I Base Address Register x Lower"
0x590	SREP_R2I_LUT_PTY_CTL	"SREP R2I LUT and Parity Control Register"
0x5A0	SREP_R2I_LUT_UPPER	"SREP R2I Upper LUT Entry Translation Address Register"
0x5AC	SREP_R2I_LUT_LOWER	"SREP R2I Lower LUT Entry Translation Address Register"
0x5B0	SREP_R2I_ISF_REQ_PRIO_CSR	"SREP R2I ISF Request Priority Control Register"
0x5B4	SREP_R2I_ISF_RESP_PRIO_CSR	"SREP R2I ISF Response Priority Control Register"
0x5BC	SREP_R2I_RIO_MISC_CSR	"SREP R2I RapidIO Miscellaneous Control CSR"
0x5C0	SREP_R2I_WM	"SREP R2I Watermarks Register"
0x5C4	SREP_R2I_BREL	"SREP R2I Buffer Release Control Register"
0x5C8	SREP_R2I_ISF_WM	"SREP R2I ISF Watermarks Control Register"
0x5CC	SREP_R2I_ISF_BREL	"SREP R2I ISF Buffer Release Control Register"
0x5F0	SREP_R2I_TTL_CTL	"SREP R2I Transaction Time-To-Live Register"
0x600	SREP_I2R_BAR{0..7}_LUT_CSR	"SREP I2R Base Address Register x LUT Entry CSR"
0x604	SREP_I2R_BAR{0..7}_UPPER	"SREP I2R Base Address Register x Upper"
0x608	SREP_I2R_BAR{0..7}_LOWER	"SREP I2R Base Address Register x Lower"
0x690	SREP_I2R_DB_BAR_UPPER	"SREP I2R Doorbell BAR Upper"
0x694	SREP_I2R_DB_BAR_LOWER	"SREP I2R Doorbell BAR Lower"
0x6A0	SREP_I2R_LUT_PTY_CTL	"SREP I2R LUT and BAR Parity Control Register"
0x6A4	SREP_I2R_LUT_TA_UPPER	"SREP I2R Upper LUT Entry Translation Register"
0x6AC	SREP_I2R_LUT_TA_LOWER	"SREP I2R Lower LUT Entry Translation Address Register"
0x6BC	SREP_I2R_LUT_TA_RIO_PARMS	"SREP I2R LUT Translation Parameters Register"

Table 174: Register Map for SREP IDT Specific Registers – Transport Layer (Continued)

Offset	Register Name	See
0x6C0	SREP_I2R_MISC_CSR	"SREP I2R Miscellaneous CSR"
0x6C4	SREP_PW_TX	"SREP Port-Write Transmit Trigger Register"
0x6CC	SREP_I2R_TTL_CTL	"SREP I2R Transaction Time-To-Live Register"
0x6E4	SREP_I2R_BREL	"SREP I2R Buffer Release Control Register"
0x6F0	SREP_NWR_ERR_WM	"SREP R2R Queue Watermarks Control Register"
0x6F4	SREP_NWR_ERR_BREL	"SREP R2R Queue Buffer Release Control Register"
0x700	SREP_R2I_EVENT_STATUS	"SREP R2I Event Status Register"
0x704	SREP_R2I_EVENT_STATUS_LOG_EN	"SREP R2I Event Status Logging Enable Register"
0x708	SREP_R2I_ISF_LOG_ERR_CMD_ ATTR	"SREP R2I Error ISF Command Attributes Capture CSR"
0x70C	SREP_R2I_ISF_LOG_ERR_DC_ ATTR	"SREP R2I Error ISF Logical Error Decomposition Attributes Capture CSR"
0x71C	SREP_RIO_LIO_ERR_CLR	"SREP Logical and Transport Layer Error Clear CSR"
0x720	SREP_R2I_PERR_STAT	"SREP R2I BAR and LUT Parity Error Status Register"
0x724	SREP_PW_RX_STATUS	"SREP Port-Write Receive Status Register"
0x730	SREP_RIO_PW0_RX_BUFF{0..7}	"SREP Port-Write 0 Receive Buffer n Registers"
0x750	SREP_RIO_PW1_RX_BUFF{0..7}	"SREP Port-Write 1 Receive Buffer n Registers"
0x780	SREP_ISF_LIO_ERR_DET	"SREP ISF Logical Error Detect CSR"
0x784	SREP_ISF_ERR_DET_EN	"SREP ISF Logical Error Logging Enable CSR"
0x788	SREP_ISF_LOG_ERR_ATTR_U	"SREP ISF Logical Error Upper Attributes Capture CSR"
0x78C	SREP_ISF_LOG_ERR_ATTR_M	"SREP ISF Logical Error Middle Attributes Capture CSR"
0x790	SREP_ISF_LOG_ERR_ATTR_L	"SREP ISF Logical Error Lower Attributes Capture CSR"
0x7A0	SREP_ISF_LOG_ERR_RIO_RATTR	"SREP ISF Logical Error RapidIO Routing Attributes Capture CSR"
0x7A4	SREP_ISF_LOG_ERR_RIO_PATTR	"SREP ISF Logical Error RapidIO Physical Attributes Capture CSR"
0x7AC	SREP_ISF_LOG_ERR_RIO_ADDR	"SREP ISF Logical Error RapidIO Lower Address Capture CSR"
0x7B0	SREP_I2R_PERR_STAT	"SREP I2R BAR and LUT Parity Error Status Register"
0x7B4	SREP_ISF_ECC_STAT	"SREP ISF ECC Error Status Register"
0x7BC	SREP_ISF_LIO_ERR_GEN	"SREP ISF Logical Error Generate CSR"
0x7C0	SREP_ISF_RTO_CTL	"SREP ISF Response Timeout Register"
0x7C8	SREP_ISF_ECC_CTL	"SREP ISF ECC Control Register"

26.10.1 SREP R2I Base Address Register x LUT Control CSR

Where x ranges from 0 to 7. Each BAR can be mapped to a configurable number of LUT entries. This register selects the size of the BAR, the index of the first LUT entry in the 256 entry LUT, the number of LUT Entries into which the BAR is divided.

Note that the minimum size of each LUT Entry is 4 KB, and the maximum size is 1 GB.

For information on the use of these registers, see "Bridging Logical I/O Requests to the Bridge ISF". Be sure to follow the restrictions on the relationship between NUM_LUTS and BAR_SIZE, captured after the register description.

If these registers must be changed during normal operation of the part, follow the procedure described in "Synchronization of Data Path Control Register Changes".

Register name: SREP_R2I_BAR{0..7}_LUT_CSR **Reset value: 0x0000_0000**	**Register offset: 500, 510,..., 570**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	FIRST_LUT							
16:23	Reserved				BAR_SIZE			
24:31	Reserved					NUM_LUTS		

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:7	Reserved	N/A	R	0
8:15	FIRST_LUT	The index of the first entry in the 256 entry LUT to be used for this BAR.	R/W	0
16:18	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
19:23	BAR_SIZE	Value that indicates the size of the window 0 = 4 KB 1 = 8 KB 2 = 16 KB 3 = 32 KB 4 = 64 KB 5 = 128 KB 6 = 256 KB 7 = 512 KB 8 = 1 MB 9 = 2 MB 10 = 4 MB 11 = 8 MB 12 = 16 MB 13 = 32 MB 14 = 64 MB 15 = 128 MB 16 = 256 MB 17 = 512 MB 18 = 1 GB 19 = 2 GB 20 = 4 GB 21 = 8 GB 22 = 16 GB All other values are Reserved	R/W	0
24:27	Reserved	N/A	R	0
28:31	NUM_LUTS	The number of LUT entries to be used for this BAR. 0 = Use 1 LUT entry 1 = Use 2 LUT entries 2 = Use 4 LUT entries 3 = Use 8 LUT entries ... 8 = Use 256 LUT entries All other values are Reserved.	R/W	0

To meet the restrictions on RapidIO-to-ISF LUT size, the following must be true:

- BAR_SIZE >= NUM_LUTS (LUT size must be at least 4 KB)
- BAR_SIZE <= 18 + NUM_LUTS (LUT size must be less than or equal to 1 GB)
- NUM_LUTS <= 8 (cannot use more LUTs than exist)

26.10.2 SREP R2I Base Address Register x Lower

This register holds the least significant bits of a 34-bit RapidIO address. For information on using this register, see "Bridging Logical I/O Requests to the Bridge ISF".

If this register must be changed during normal operation of the part, follow the procedure described in "Synchronization of Data Path Control Register Changes".

Register name: SREP_R2I_BAR{0..7}_LOWER **Reset value: 0x0000_0000**	**Register offset: 508, 518,..., 578**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	ADDR							
08:15	ADDR							
16:23	ADDR				Reserved			
24:31	Reserved			BAR_EN	Reserved		MS_ADDR	

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:19	ADDR	Bits 2-21 of the 34 bit RapidIO address that this BAR should respond to. The smallest BAR size is 4 KB.	R/W	0
20:26	Reserved	N/A	R	0
27	BAR_EN	1 if the BAR is enabled, 0 if the BAR is disabled	R/W	0
28:29	Reserved	N/A	R	0
30:31	MS_ADDR	Bits 0-1 of the 34 bit RapidIO address that this BAR should respond to.	R/W	0

26.10.3 SREP R2I LUT and Parity Control Register

This register controls which LUT entry is accessible from the LUT entry registers, as well as parity error insertion and parity checking. For information on the use of these registers, see "Bridging Logical I/O Requests to the Bridge ISF".

Register name: SREP_R2I_LUT_PTY_CTL **Reset value: 0x0000_0000**	**Register offset: 590**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	AUTO_INC	Reserved					LUT_PAR_INV	LUT_PAR_DIS
08:15	Reserved		BAR_PAR_INV	BAR_PAR_DIS	Reserved			
16:23	Reserved							
24:31	LUT_IDX							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	AUTO_INC	Controls automatic incrementing of LUT_IDX when the "SREP R2I Lower LUT Entry Translation Address Register" is accessed.	R/W	0
1:5	Reserved	N/A	R	0
6	LUT_PAR_INV	LUT Parity Inversion Control 0 = Compute and store parity correctly when LUT entry registers are written 1 = Invert the parity computed when LUT registers are written.	R/W	0
7	LUT_PAR_DIS	LUT Parity Disable 0 = Check LUT parity and act on errors 1 = Do not check LUT parity	R/W	0
8:9	Reserved	N/A	R	0
10	BAR_PAR_INV	BAR Parity Inversion Control 0 = Compute and store parity correctly when BAR entry registers are written 1 = Invert the parity computed when BAR registers are written.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
11	BAR_PAR_DIS	BAR Parity Disable 0 = Check BAR parity and act on errors 1 = Do not check BAR parity This bit controls parity checking for the following registers: • "SREP R2I Base Address Register x LUT Control CSR" • "SREP R2I Base Address Register x Lower" • "SREP Local Configuration Space Base Address CSR"	R/W	0
12:23	Reserved	N/A	R	0
24:31	LUT_IDX	Index of the LUT entry accessible through SREP R2I LUT registers.	R/W	0

26.10.4 SREP R2I Upper LUT Entry Translation Address Register

There are a configurable number of LUT entries for each BAR. The LUT entry is selected through the LUT_IDX field of the "SREP R2I LUT and Parity Control Register". For information on the use of these registers, see "Bridging Logical I/O Requests to the Bridge ISF".

If this register must be changed during normal operation of the part, follow the procedure described in "Synchronization of Data Path Control Register Changes".

Register name: SREP_R2I_LUT_UPPER **Reset value: Undefined**	**Register offset: 5A0**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	T_ADDR							
08:15	T_ADDR							
16:23	T_ADDR							
24:31	T_ADDR							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:31	T_ADDR	The value of this field is replaced in the upper 64 bits of address of the Logical I/O packet address, and used as the OCN address for the transaction.	R/W	Undefined

26.10.5 SREP R2I Lower LUT Entry Translation Address Register

There are a configurable number of LUT entries for each BAR. The LUT entry is selected through the LUT_IDX field of the "SREP R2I LUT and Parity Control Register".

When "SREP R2I LUT and Parity Control Register".AUTO_INC is 1, reading or writing this register increments the "SREP R2I LUT and Parity Control Register".LUT_IDX after the read/write is completed. For information on the use of these registers, see "Bridging Logical I/O Requests to the Bridge ISF".

If this register must be changed during normal operation of the part, follow the procedure described in "Synchronization of Data Path Control Register Changes".

Register name: SREP_R2I_LUT_LOWER **Reset value: Undefined**	**Register offset: 5AC**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	T_ADDR							
08:15	T_ADDR							
16:23	T_ADDR				Reserved			
24:31	ISF_PORT				SWAP		WR_EN	RD_EN

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:19	T_ADDR	The value of this field is replaced in the least significant 4 bytes of the Logical I/O packet address, and used as the Bridge ISF address for the transaction.	R/W	Undefined
20:23	Reserved	N/A	R	Undefined
24:27	ISF_PORT	The Bridge ISF port to which transactions that hit in this LUT Entry should be routed. 0000 = SREP 0001 = PCI Interface Note: For information on how transactions that hit this LUT are handled, see "Bridge ISF".	R/W	Undefined
28:29	SWAP	These bits control no functionality within the SREP.	R/W	Undefined
30	WR_EN	This LUT Entry can be accessed with NWRITE, NWRITE_R and SWRITE transactions	R/W	Undefined
31	RD_EN	This LUT Entry can be accessed with NREAD transactions	R/W	Undefined

26.10.6 SREP R2I ISF Request Priority Control Register

This register controls the priority of Bridge ISF request transactions mastered by the SREP. If this register must be changed during normal operation of the part, follow the procedure described in "Synchronization of Data Path Control Register Changes".

Register name: SREP_R2I_ISF_REQ_PRIO_CSR **Reset value: 0x0000_0080**	**Register offset: 5B0**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							SYNC_REQ
24:31	MWR_BLK		MRD_BLK		Reserved			

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:21	Reserved	N/A	R	0
22:23	SYNC_REQ	This field controls no functionality in the Tsi620	R/W	0
24:25	MWR_BLK	ISF Priority of a Memory Write Block request transaction.	R/W	2
26:27	MRD_BLK	ISF Priority of a Memory Read Block request transaction.	R/W	0
28:31	Reserved	N/A	R	0

26.10.7 SREP R2I ISF Response Priority Control Register

This register controls the priority of Bridge ISF response transactions mastered by the SREP. If this register must be changed during normal operation of the part, follow the procedure described in "Synchronization of Data Path Control Register Changes".

Register name: SREP_R2I_ISF_RESP_PRIO_CSR Reset value: 0x1122_0000	Register offset: 5B4

Bits	0	1	2	3	4	5	6	7
00:07	Reserved		ERROR		Reserved		NORMAL	
08:15	Reserved		SYNC_ERROR		Reserved		SYNC_NORMAL	
16:23	Reserved							
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0:1	Reserved	N/A	R	0
2:3	ERROR	Error Response priority	R/W	1
4:5	Reserved	N/A	R	0
6:7	NORMAL	Normal Response priority	R/W	1
8:9	Reserved	N/A	R	0
10:11	SYNC_ERROR	This field controls no functionality in the Tsi620.	R/W	2
12:13	Reserved	N/A	R	0
14:15	SYNC_NORMAL	This field controls no functionality in the Tsi620.	R/W	2
16:31	Reserved	N/A	R	0

26.10.8 SREP R2I RapidIO Miscellaneous Control CSR

This register controls whether or not implementation-specific values (0b1100-0b1111) in response packets are accepted, and if they are accepted, how they are interpreted. It also controls whether or not RapidIO error responses are sent on ISF TEA and response timeout events. For information on using this register, see "RapidIO Unsupported Transaction Events".

If this register must be changed during normal operation of the part, follow the procedure described in "Synchronization of Data Path Control Register Changes".

Register name: SREP_R2I_RIO_MISC_CSR **Reset value: 0x0000_0020**	**Register offset: 5BC**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							
24:31	Reserved		LCS_DIS	ERESP_DIS	Reserved		RESP_CTL	

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:25	Reserved	N/A	R	0
26	LCS_DIS	Local Address Space Disable 0 = The "SREP Local Configuration Space Base Address CSR" is active. 1 = The "SREP Local Configuration Space Base Address CSR" is disabled. Note: This bit is cleared to 0 on any write to the "SREP Local Configuration Space Base Address CSR".	R/W	1
27	ERESP_DIS	Controls whether RapidIO Error Responses are sent on ISF Response Timeouts and TEAs. 0b0 = ISF Response Timeouts and TEAs result in RapidIO Error Responses 0b1 = RapidIO Error Responses are not sent on ISF Response Timeouts and TEAs.	R/W	0x0
28:29	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
30:31	RESP_CTL	Interpretation of implementation-specific values in Response packets. 00 = Implementation-specific values cause Unsupported Transaction events (see "RapidIO Unsupported Transaction Events") 01 = Implementation-specific values are interpreted as DONE 10 = Implementation-specific values are interpreted as ERROR 11 = Reserved	R/W	0x0

26.10.9 SREP R2I Watermarks Register

This register reserves R2I buffers for the assembly of responses to decomposed I2R transactions. It also controls the behavior of transfer of RapidIO packets into the R2I Data Buffers/Header queue based on RapidIO packet priority and the fill level of the R2I Request Queue and the Register Request Queue.

If this register must be changed during normal operation of the part, follow the procedure described in "Synchronization of Data Path Control Register Changes".

Register name: SREP_R2I_WM **Reset value: 0x1001_0203**	**Register offset: 5C0**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved			DECOMP				
08:15	Reserved			PRIO2WM				
16:23	Reserved			PRIO1WM				
24:31	Reserved			PRIO0WM				

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:2	Reserved	N/A	R	0
3:7	DECOMP	The number of buffers that can be reserved to assemble the response for I2R decomposed requests. Setting this value to be more than 28 (0x1C) results in 28 buffers being reserved. Note: Setting this value to 0 is a programming error. The minimum value for this field is 1.	R/W	0x10
8:10	Reserved	N/A	R	0
11:15	PRIO2WM	If this number or fewer R2I Queue entries are free, then RapidIO packets of priority 2 cannot be received. Note that the maximum possible value of 31 corresponds to the maximum number of buffers available for R2I transactions. The 32nd buffer is reserved for I2I transactions.	R/W	1
16:18	Reserved	N/A	R	0
19:23	PRIO1WM	If this number or fewer R2I Queue entries are free, then RapidIO requests of priority 1 cannot be received. Note that the maximum possible value of 31 corresponds to the maximum number of buffers available for R2I transactions. The 32nd buffer is reserved for I2I transactions.	R/W	2
24:26	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
27:31	PRIO0WM	If this number or fewer R2I Queue entries are free, then RapidIO packets of priority 0 cannot be received. Note that the maximum possible value of 31 corresponds to the maximum number of buffers available for R2I transactions. The 32nd buffer is reserved for I2I transactions.	R/W	3

26.10.10 SREP R2I Buffer Release Control Register

This register delays the reporting of free R2I buffers in SREP to the MAC. If this register must be changed during normal operation of the part, follow the procedure described in "Synchronization of Data Path Control Register Changes".

Register name: SREP_R2I_BREL **Reset value: 0x00FF_0000**	**Register offset: 5C4**

Bits	0	1	2	3	4	5	6	7
00:07	REL_MGMT_EN	REL_TO_MODE	Reserved		TO_CNT			
08:15	REL_MGMT_TO							
16:23	Reserved		REL_MGMT_RES					
24:31	Reserved		REL_MGMT_STOP					

Bits	Name	Description	Type	Reset Value
0	REL_MGMT_EN	Buffer Release Management Enable	R/W	0
1	REL_TO_MODE	Buffer Release Timeout Mode Control 0 = After a timeout, do not engage buffer release management until the number of free buffers reaches the REL_MGMT_RES level. 1 = After a timeout, engage buffer release management and restart the timeout if the number of free buffers ever reaches the REL_MGMT_STOP level.	R/W	0
2:3	Reserved	N/A	R	0
4:7	TO_CNT	Count of the number of times the release management timeout period has expired. This counter rolls over to 0 once it reaches its maximum value.	R	0
8:15	REL_MGMT_TO	Timeout value. 0 = TIMEOUT is disabled (this is not recommended). 1 = 0xFF - Maximum amount of time that can expire before resuming reporting of buffer fill levels. Each tick in this field is equivalent to 32 Reference Clock periods. It is recommended that this timeout period be set based on the transmission period for REL_MGMT_STOP-REL_MGMT_RES maximum sized packets.	R/W	0xFF
16:17	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
18:23	REL_MGMT_RES	Buffer Release Resume Level. The number of buffers that must be free before free buffer reporting can resume. REL_MGMT_RES must be greater than REL_MGMT_STOP. For other restrictions on the value of this register field, see "R2I Watermark and Buffer Release Management Register Value Restrictions".	R/W	0
24:25	Reserved	N/A	R	0
26:31	REL_MGMT_STOP	Buffer Release Stop Level. The number of buffers that must be free when free buffer reporting must stop. REL_MGMT_RES must be greater than REL_MGMT_STOP. For other restrictions on the value of this register field, see "R2I Watermark and Buffer Release Management Register Value Restrictions".	R/W	0

26.10.11 SREP R2I ISF Watermarks Control Register

This register controls the behavior of transfer of ISF requests from the R2I Data Buffers/Header queue to the R2I ISF Request queue based on ISF transaction priority and R2I ISF Request Queue fill level.

If this register must be changed during normal operation of the part, follow the procedure described in "Synchronization of Data Path Control Register Changes".

Register name: SREP_R2I_ISF_WM **Reset value: 0x0001_0203**	**Register offset: 5C8**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved				PRIO2WM			
16:23	Reserved				PRIO1WM			
24:31	Reserved				PRIO0WM			

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:11	Reserved	N/A	R	0
12:15	PRIO2WM	If this number or fewer R2I ISF Request Queue entries are free, then ISF requests of priority 2 cannot be received.	R/W	1
16:19	Reserved	N/A	R	0
20:23	PRIO1WM	If this number or fewer R2I ISF Request Queue entries are free, then ISF requests of priority 1 cannot be received.	R/W	2
24:27	Reserved	N/A	R	0
28:31	PRIO0WM	If this number or fewer R2I ISF Request Queue entries are free, then ISF requests of priority 0 cannot be received.	R/W	3

26.10.12 SREP R2I ISF Buffer Release Control Register

This register controls the behavior of reception of RapidIO Packets based on RapidIO transaction priority and R2I Buffer fill level.

If this register must be changed during normal operation of the part, follow the procedure described in "Synchronization of Data Path Control Register Changes".

Register name: SREP_R2I_ISF_BREL **Reset value: 0x00FF_0000**	**Register offset: 5CC**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	REL_MGMT_EN	REL_TO_MODE	Reserved		TO_CNT			
08:15	REL_MGMT_TO							
16:23	Reserved				REL_MGMT_RES			
24:31	Reserved				REL_MGMT_STOP			

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	REL_MGMT_EN	Buffer Release Management Enable	R/W	0
1	REL_TO_MODE	Buffer Release Timeout Mode Control 0 = After a timeout, do not engage buffer release management until the number of free buffers reaches the REL_MGMT_RES level. 1 = After a timeout, engage buffer release management and restart the timeout if the number of free buffers ever reaches the REL_MGMT_STOP level.	R/W	0
2:3	Reserved	N/A	R	0
4:7	TO_CNT	Count of the number of times the release management timeout period has expired. This counter rolls over to 0 once it reaches its maximum value.	R	0
8:15	REL_MGMT_TO	Timeout value 0 = TIMEOUT is disabled (this is not recommended). 1 = 0xFF. This is the maximum amount of time that can expire before resuming reporting of buffer fill levels. Each tick in this field is equivalent to 32 internal clock cycles periods. It is recommended that this timeout period be set based on the transmission period for REL_MGMT_STOP-REL_MGMT_RES maximum sized packets.	R/W	0xFF
16:19	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
20:23	REL_MGMT_RES	Buffer Release Resume Level. The number of buffers that must be free before free buffer reporting can resume. REL_MGMT_RES must be greater than REL_MGMT_STOP.	R/W	0
24:27	Reserved	N/A	R	0
28:31	REL_MGMT_STOP	Buffer Release Stop Level. The number of buffers that must be free when free buffer reporting must stop. REL_MGMT_RES must be greater than REL_MGMT_STOP.	R/W	0

Do not program this register in traffic; only program these registers after reset.

26.10.13 SREP R2I Transaction Time-To-Live Register

This is a IDT-specific feature. This register enforces an overall timeout on R2I transactions existence within the R2I Data Buffers/Header queues (see "RapidIO Logical Layer Time-to-Live Expired").

If this register must be changed during normal operation of the part, follow the procedure described in "Synchronization of Data Path Control Register Changes".

Register name: SREP_R2I_TTL_CTL **Reset value: 0x0000_0000**	**Register offset: 5F0**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	TVAL							
08:15	TVAL							
16:23	TVAL							
24:31	Reserved							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:23	TVAL	This timer runs on the ISF reference clock. The timeout period is computed by: TVAL * RIO_REF_CLK/32. For a clock frequency of 156.25 MHz, the maximum timeout period is 3.4 seconds and the granularity is 204.8 nsec. For a clock frequency of 133 MHz, the maximum timeout period is 4.0 seconds and the granularity is 240 nsec. A value of 0 disables the R2I Transaction Time-to-Live.	R/W	0x000000
24:31	Reserved	N/A	R	0

26.10.14 SREP I2R Base Address Register x LUT Entry CSR

Where x is 0 to 7. Each BAR can be mapped to a configurable number of LUT entries. This register selects the index of the first LUT entry in the 256 entry LUT, and the number of LUT entries into which the BAR is divided.

Note that the minimum size of each LUT Entry is 4 KB, and the maximum size is 1 GB. For information on using these registers, see “Bridging ISF Requests to RapidIO”.

If this register must be changed during normal operation of the part, follow the procedure described in “Synchronization of Data Path Control Register Changes”.

Register name: SREP_I2R_BAR{0..7}_LUT_CSR **Reset value: 0x0000_0000**	**Register offset: 600, 610, 620,..., 670**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	FIRST_LUT							
16:23	Reserved				BAR_SIZE			
24:31	Reserved					NUM_LUTS		

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:7	Reserved	N/A	R	0
8:15	FIRST_LUT	The index of the first entry in the 256 entry LUT to be used for this BAR.	R/W	0
16:18	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
19:23	BAR_SIZE	Value that indicates the size of the window 0 = 4 KB 1 = 8 KB 2 = 16 KB 3 = 32 KB 4 = 64 KB 5 = 128 KB 6 = 256 KB 7 = 512 KB 8 =1 MB 9 = 2 MB 10 = 4 MB 11 = 8 MB 12 = 16 MB 13 = 32 MB 14 = 64 MB 15 = 128 MB 16 = 256 MB 17 = 512 MB 18 = 1 GB 19 = 2 GB 20 = 4 GB 21 = 8 GB 22 = 16 GB All other values are Reserved	R/W	0
24:27	Reserved	N/A	R	0
28:31	NUM_LUTS	The number of LUT entries to be used for this BAR. 0 = Use 1 LUT entry 1 = Use 2 LUT entries 2 = Use 4 LUT entries 3 = Use 8 LUT entries ... 8 = Use 256 LUT entries All other values are Reserved.	R/W	0

26.10.15 SREP I2R Base Address Register x Upper

Where x is 0 to 7. This register holds the most significant 32 bits of a 64-bit Bridge ISF address at which this BAR should respond. For information on using this register, see “Bridging ISF Requests to RapidIO”.

If this register must be changed during normal operation of the part, follow the procedure described in “Synchronization of Data Path Control Register Changes”.

Register name: SREP_I2R_BAR{0..7}_UPPER **Reset value: 0x0000_0000**	**Register offset: 604, 614, 624,..., 674**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	ADDR							
08:15	ADDR							
16:23	ADDR							
24:31	ADDR							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:31	ADDR	Most significant bits of the 64-bit Bridge ISF address range used to translate ISF transactions to RapidIO transactions.	R/W	0

26.10.16 SREP I2R Base Address Register x Lower

Where x is 0 to 7. This register holds the least significant bits of a 64-bit Bridge ISF address at which this BAR should respond. For information on using this register, see “Bridging ISF Requests to RapidIO”.

If this register must be changed during normal operation of the part, follow the procedure described in “Synchronization of Data Path Control Register Changes”.

Register name: SREP_I2R_BAR{0..7}_LOWER **Reset value: 0x0000_0000**	**Register offset: 608, 618, 628,..., 678**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	ADDR							
08:15	ADDR							
16:23	ADDR				Reserved			
24:31	Reserved							BAR_EN

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:19	ADDR	Most significant 20 bits of the least significant 32 bits of the Bridge ISF address range used to translate Bridge ISF transactions to RapidIO logical I/O transactions.	R/W	0
20:30	Reserved	N/A	R	0
31	BAR_EN	1 if the BAR is enabled, 0 if the BAR is disabled	R/W	0

26.10.17 SREP I2R Doorbell BAR Upper

The "SREP I2R Doorbell BAR Upper" holds the most significant 32 bits of a 64-bit Bridge ISF address at which this BAR should respond. This BAR is 4 KB in size. All entries that hit this BAR are translated to RapidIO Doorbell transactions. For information on using this register, see "Bridging ISF Requests to RapidIO".

If this register must be changed during normal operation of the part, follow the procedure described in "Synchronization of Data Path Control Register Changes".

Register name: SREP_I2R_DB_BAR_UPPER **Reset value: 0x0000_0000**	**Register offset: 690**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	ADDR							
08:15	ADDR							
16:23	ADDR							
24:31	ADDR							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:31	ADDR	Most significant bits of the 64-bit internal address range used to translate Bridge ISF transactions to RapidIO Doorbell transactions.	R/W	0

26.10.18 SREP I2R Doorbell BAR Lower

This register holds the least significant bits of a 64-bit Bridge ISF address at which this BAR should respond. This BAR is 4 KB in size. All entries that hit in this BAR are translated to RapidIO Doorbell transactions. For information on using these registers, see “Bridging ISF Requests to RapidIO”.

If this register must be changed during normal operation of the part, follow the procedure described in “Synchronization of Data Path Control Register Changes”.

Register name: SREP_I2R_DB_BAR_LOWER **Reset value: 0x0000_0000**	**Register offset: 694**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	ADDR							
08:15	ADDR							
16:23	ADDR				Reserved			
24:31	Reserved							BAR_EN

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:19	ADDR	Most significant 20 bits of the least significant 32 bits of the internal address ranges used to translate Bridge ISF transactions to RapidIO Doorbell transactions.	R/W	0
20:30	Reserved	N/A	R	0
31	BAR_EN	1 if the BAR is enabled, 0 if the BAR is disabled	R/W	0

26.10.19 SREP I2R LUT and BAR Parity Control Register

This register controls selection of which BAR register and associated LUT registers are being accessed. For information on using this register, see "Bridging ISF Requests to RapidIO".

Register name: SREP_I2R_LUT_PTY_CTL **Reset value: 0x0000_0000**	**Register offset: 6A0**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	AUTO_INC	Reserved					LUT_PAR_INV	LUT_PAR_DIS
08:15	Reserved		BAR_PAR_INV	BAR_PAR_DIS	Reserved			
16:23	Reserved							
24:31	LUT_IDX							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	AUTO_INC	Controls automatic incrementing of LUT_IDX when the "SREP I2R LUT Translation Parameters Register" is read or written.	R/W	0
1:5	Reserved	N/A	R	0
6	LUT_PAR_INV	LUT Parity Inversion Control 0 = Compute and store parity correctly when LUT registers are written 1 = Invert the parity computed when LUT registers are written.	R/W	0
7	LUT_PAR_DIS	LUT Parity Disable 0 = Check LUT parity and act on errors 1 = Do no check LUT parity	R/W	0
8:9	Reserved	N/A	R	0
10	BAR_PAR_INV	BAR Parity Inversion Control 0 = Compute and store parity correctly when BAR registers are written 1 = Invert the parity computed when BAR registers are written.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
11	BAR_PAR_DIS	BAR Parity Disable 0 = Check BAR parity and act on errors 1 = Do no check BAR parity This bit controls parity checking for the following registers: • "SREP I2R Base Address Register x LUT Entry CSR" • "SREP I2R Base Address Register x Upper" • "SREP I2R Base Address Register x Lower" • "SREP I2R Doorbell BAR Upper" • "SREP I2R Doorbell BAR Lower"	R/W	0
12:23	Reserved	N/A	R	0
24:31	LUT_IDX	Index of the LUT entry accessible through SREP I2R LUT registers.	R/W	0

26.10.20 SREP I2R Upper LUT Entry Translation Register

There are a configurable number of LUT entries for each BAR. The LUT entry is selected through the LUT_IDX field of the "SREP I2R LUT and BAR Parity Control Register". For information on using this register, see "Bridging ISF Requests to RapidIO".

If this register must be changed during normal operation of the part, follow the procedure described in "Synchronization of Data Path Control Register Changes".

Register name: SREP_I2R_LUT_TA_UPPER **Reset value: Undefined**	**Register offset: 6A4**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	HOP_COUNT							
08:15	WR_EN	RD_EN	BST_2_BLK	PFTCH	Reserved		TT_CODE	
16:23	DEST_ID_M							
24:31	DEST_ID							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:7	HOP_COUNT	The hop count to be used if Maintenance Read or Maintenance Write transactions are originated from this LUT Entry. The hop count value sent in the Maintenance Read/Write requests is 1 less than the value in the HOP_COUNT field. Note that a HOP_COUNT value of 0 is invalid for the Tsi620.	R/W	Undefined
8	WR_EN	Control write access to this LUT Entry 0 = No Bridge ISF write transactions are accepted 1 = Bridge ISF write transactions are accepted	R/W	Undefined
9	RD_EN	Control read access to this LUT Entry 0 = No Bridge ISF read transactions are accepted 1 = Bridge ISF read transactions are accepted	R/W	Undefined
10	BST_2_BLK	Controls what SARing algorithm is used for Bridge ISF Write Burst transactions (see "Bridging ISF Requests to RapidIO"). 0 = Use the Bridge ISF Write Burst SARing Algorithm for Write Burst transactions 1 = Use the Bridge ISF Write Block SARing Algorithm for Write Burst transactions	R/W	Undefined

(Continued)

Bits	Name	Description	Type	Reset Value
11	PFTCH	Control use of 'prefetchable' SARing algorithm (see "Bridging ISF Requests to RapidIO"). 0 = Reads must access precisely the bytes requested from the Bridge ISF 1 = Reads may access more than the bytes requested from the Bridge ISF	R/W	Undefined
12:13	Reserved	N/A	R	0
14:15	TT_CODE	Encoded as per RapidIO specification: 0b00 = 8-bit dest ID 0b01 = 16-bit dest ID 0b1x = Reserved	R/W	Undefined
16:23	DEST_ID_M	Most significant 8 bits of the target 16-bit destination ID.	R/W	Undefined
24:31	DEST_ID	Least significant 8 bits of the target 16-bit destination ID, or only 8 bits of the 8-bit destination ID	R/W	Undefined

26.10.21 SREP I2R Lower LUT Entry Translation Address Register

There are a configurable number of LUT entries for each BAR. The LUT entry is selected through the LUT_IDX field of the "SREP I2R LUT and BAR Parity Control Register". For information on using this register, see "Bridging ISF Requests to RapidIO".

If this register must be changed during normal operation of the part, follow the procedure described in "Synchronization of Data Path Control Register Changes".

Register name: SREP_I2R_LUT_TA_LOWER **Reset value: Undefined**	**Register offset: 6AC**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	T_ADDR							
08:15	T_ADDR							
16:23	T_ADDR					Reserved	WR_CRF	RD_CRF
24:31	RD_PRIO		WR_PRIO			SWAP	MS_ADDR	

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:19	T_ADDR	Bits 2-21 of the 34-bit RapidIO address for packets generated from this LUT entry. The smallest LUT entry size is 4 KB.	R/W	Undefined
20:21	Reserved	N/A	R	0
22	WR_CRF	Critical Request Flow bit value for Write transactions	R/W	Undefined
23	RD_CRF	Critical Request Flow bit value for Read transactions	R/W	Undefined
24:25	RD_PRIO	Priority of packets issued for Bridge ISF Read transactions Note: • The maximum valid priority for reads is 2, to allow responses to be sent at priority 3. • RD_PRIO must be less than or equal to WR_PRIO, otherwise the logical layer ordering rules are violated.	R/W	Undefined
26:27	WR_PRIO	Priority of packets issued for Bridge ISF Write transactions Note: • The maximum valid priority for NWRITE and SWRITE is 3 • The maximum valid priority for NWRITE_R is 2, to allow responses to be sent at priority 3. • RD_PRIO must be less than or equal to WR_PRIO, otherwise the logical layer ordering rules are violated.	R/W	Undefined
28:29	SWAP	This value controls no functionality within the SREP.	R/W	Undefined
30:31	MS_ADDR	Most significant 2 bits of the RapidIO address that this window should respond to.	R/W	Undefined

26.10.22 SREP I2R LUT Translation Parameters Register

There are a configurable number of LUT entries for each BAR. The LUT entry is selected through the LUT_IDX field of the "SREP I2R LUT and BAR Parity Control Register".

When "SREP I2R LUT and BAR Parity Control Register".AUTO_INC is 1, reading or writing this register increments the "SREP I2R LUT and BAR Parity Control Register".LUT_IDX after the read/write is completed. For information on using this register, see "Bridging ISF Requests to RapidIO".

If this register must be changed during normal operation of the part, follow the procedure described in "Synchronization of Data Path Control Register Changes". LUTs cannot be used to send Doorbell packets. For information on how to master Doorbell (FType 10) packets, see "Bridge ISF Transaction Conversion to RapidIO Doorbells".

Register name: SREP_I2R_LUT_TA_RIO_PARMS **Reset value: Undefined**	**Register offset: 6BC**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved							
16:23	RD_FTYPE				RD_SUBTYPE			
24:31	WR_FTYPE				WR_SUBTYPE			

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:15	Reserved	N/A	R	0
16:19	RD_FTYPE	FTYPE of RapidIO Request issued for Bridge ISF Read transaction. 0, 1 = Reserved 2 = Request 3–7 = Reserved 8 = Maintenance 9 = Reserved 10–15 = Reserved Note: Programming the RD_FTYPE to a reserved value will cause undefined behavior, and may cause the SREP to stop accepting more ISF transactions.	R/W	Undefined

(Continued)

Bits	Name	Description	Type	Reset Value
20:23	RD_SUBTYPE	Subtype of the RD_FTYPE. RD_FTYPE = 2, RD SUBTYPE must be 4 (NREAD). All other values of RD_SUBTYPE are reserved. RD_FTYPE = 8, RD_SUBTYPE must be 0. All other values of RD_SUBTYPE are reserved. Note: Programming the RD_SUBTYPE to a reserved value will cause undefined behavior, and may cause the SREP to stop accepting more ISF transactions.	R/W	Undefined
24:27	WR_FTYPE	FTYPE of RapidIO Request issued for Bridge ISF Write transaction. 0–4 = Reserved 5 = Write 6 = Streaming write 7 = Reserved 8 = Maintenance 9 = Reserved 10–15 = Reserved Note: Programming the WR_FTYPE to a reserved value will cause undefined behavior, and may cause the SREP to stop accepting more ISF transactions.	R/W	Undefined
28:31	WR_SUBTYPE	Subtype of the WR_FTYPE. WR_FTYPE = 5 (Write) • WR_SUBTYPE value of 4 (NWRITE). • WR_SUBTYPE value is 5 (NWRITE_R). • All other values of WR_SUBTYPE are reserved. WR_FTYPE = 6 (Streaming write) • WR_SUBTYPE must be 0. • All other values of WR_SUBTYPE are reserved. WR_FTYPE = 8 • WR_SUBTYPE must be 1 (Write Request) • All other values of WR_SUBTYPE are reserved Note: Programming the WR_SUBTYPE to a reserved value will cause undefined behavior, and may cause the SREP to stop accepting more ISF transactions.	R/W	Undefined

26.10.23 SREP I2R Miscellaneous CSR

This register controls whether or not RapidIO response timeouts result in Bridge ISF Error responses. It also controls the transaction size translation for zero length Bridge ISF read requests (see "Debug Support").If this register must be changed during normal operation of the part, follow the procedure in "Synchronization of Data Path Control Register Changes".

Register name: SREP_I2R_MISC_CSR **Reset value: 0x2000_0008**	**Register offset: 6C0**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved		ERROR_RESP		Reserved			
08:15	Reserved				SYNC_REQ		Reserved	
16:23	Reserved							
24:31	Reserved			I_ERESP_DIS	RDSIZE_0RD			

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:1	Reserved	N/A	R	0
2:3	ERROR_RESP	RapidIO priority to use when prioritizing the processing of a Bridge ISF Error response in the I2R queues.	R/W	2
4:11	Reserved	N/A	R	0
12:13	SYNC_REQ	This field controls no functionality in the Tsi620.	R/W	0
14:26	Reserved	N/A	R	0
27	I_ERESP_DIS	Controls whether RapidIO response timeouts result in a Bridge ISF Error Response being sent. 0 = Send a Bridge ISF Error response when a RapidIO request times out waiting for a response 1 = Do not send a Bridge ISF Error response when a RapidIO request times out waiting for a response	R/W	0
28:31	RDSIZE_0RD	RDSIZE_0RD defines the byte lanes used in translating a zero-length Bridge ISF read to a RapidIO NREAD or Maintenance Read transaction. RDSIZE_0RD is defined in accordance with the RapidIO specification.	R/W	0b1000

26.10.24 SREP Port-Write Transmit Trigger Register

This register triggers the immediate transmission of a port-write. The data used in the implementation-specific fields of the port write is taken from this register (see "SREP Port-Write Transmit Trigger Register Transmission").

Register name: SREP_PW_TX Reset value: 0x0000_0000	Register offset: 6C4

Bits	0	1	2	3	4	5	6	7
00:07	PW_PEND	Reserved						
08:15	PW_DATA							
16:23	PW_DATA							
24:31	PW_DATA							

Bits	Name	Description	Type	Reset Value
0	PW_PEND	Indicates when a port-write is sent with the data in PW_DATA. 0 = No port-write is triggered from this register 1 = A port-write triggered from this register has not been sent yet This bit is cleared to 0 when the port-write packet is transmitted by the SREP.	R/W1S	0
1:7	Reserved	N/A	R	0
8:31	PW_DATA	The 24 bits of implementation-specific data to send in the port-write.	R/W	0

26.10.25 SREP I2R Transaction Time-To-Live Register

This is a IDT-specific feature. This register enforces an overall timeout on I2R transactions, register access responses, and R2R Queue transactions within the SREP (see "Transaction End-to-End Time-to-Live").

If this register must be changed during normal operation of the part, follow the procedure in "Synchronization of Data Path Control Register Changes".

Register name: SREP_I2R_TTL_CTL **Reset value: 0x0000_0000**	**Register offset: 6CC**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	TVAL							
08:15	TVAL							
16:23	TVAL							
24:31	Reserved							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:23	TVAL	This timer runs on the Bridge ISF reference clock. A value of 0 disables the I2R Time-to-Live. This timer runs on the Bridge ISF Clock. The timeout period is computed by: 7 * TVAL * (ISF Clock Interval * 4). For a clock frequency of 125 MHz, the maximum timeout period is 3.76 seconds and the granularity is 224 nsec. For a clock frequency of 156.25 MHz, the maximum timeout period is 3.00 seconds and the granularity is 179.2 nsec.	R/W	0x000000
24:31	Reserved	N/A	R	0

26.10.26 SREP I2R Buffer Release Control Register

This register controls when the Bridge ISF learns of free spots in the I2R Data Buffer/Header Queue. This delay can prevent starvation of low priority packets by high priority packets.

If this register must be changed during normal operation of the device, follow the procedure in "Synchronization of Data Path Control Register Changes".

Register name: SREP_I2R_BREL **Reset value: 0x00FF_0000**	**Register offset: 6E4**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	REL_MGMT_EN	REL_TO_MODE	Reserved		TO_CNT			
08:15	REL_MGMT_TO							
16:23	Reserved		REL_MGMT_RES					
24:31	Reserved		REL_MGMT_STOP					

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	REL_MGMT_EN	Buffer Release Management Enable	R/W	0
1	REL_TO_MODE	Buffer Release Timeout Mode Control 0 = After a timeout, do not engage buffer release management until the number of free buffers reaches the REL_MGMT_RES level. 1 = After a timeout, engage buffer release management and restart the timeout if the number of free buffers ever reaches the REL_MGMT_STOP level.	R/W	0
2:3	Reserved	N/A	R	0
4:7	TO_CNT	Count of the number of times the release management timeout period has expired. This counter rolls over to 0 once it reaches its maximum value.	R	0
8:15	REL_MGMT_TO	Timeout value. 0 = Timeout is disabled (this is not recommended). 1 = 0xFF. This is the maximum amount of time that can expire before resuming reporting of buffer fill levels. Each tick in this field is equivalent to 32 internal clock periods. It is recommended that this timeout period be set based on the transmission period for REL_MGMT_STOP-REL_MGMT_RES maximum sized packets.	R/W	0xFF
16:17	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
18:23	REL_MGMT_RES	Buffer Release Resume Level. The number of buffers that must be free before free buffer reporting can resume. REL_MGMT_RES must be greater than REL_MGMT_STOP. For other restrictions on the value of this register field, see "I2R Watermark and Buffer Release Management Register Value Restrictions".	R/W	0
24:25	Reserved	N/A	R	0
26:31	REL_MGMT_STOP	Buffer Release Stop Level. The number of buffers that must be free when free buffer reporting must stop. REL_MGMT_RES must be greater than REL_MGMT_STOP. For other restrictions on the value of this register field, see "I2R Watermark and Buffer Release Management Register Value Restrictions".	R/W	0

Do not program this register in traffic; only program these registers after reset.

26.10.27 SREP R2R Queue Watermarks Control Register

This register controls the behavior of reception of R2R Queue transactions based on RapidIO transaction priority and R2R Queue fill level.

If this register must be changed during normal operation of the part, follow the procedure in "Synchronization of Data Path Control Register Changes".

Register name: SREP_NWR_ERR_WM **Reset value: 0x0001_0203**	**Register offset: 6F0**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved		PRIO2WM					
16:23	Reserved		PRIO1WM					
24:31	Reserved		PRIO0WM					

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:9	Reserved	N/A	R	0
10:15	PRIO2WM	If this number or fewer register response buffers are free, then Register responses of priority 2 cannot be received.	R/W	1
16:17	Reserved	N/A	R	0
18:23	PRIO1WM	If this number or fewer register response buffers are free, then Register responses of priority 1 cannot be received.	R/W	2
24:25	Reserved	N/A	R	0
26:31	PRIO0WM	If this number or fewer register response buffers are free, then Register responses of priority 0 cannot be received.	R/W	3

26.10.28 SREP R2R Queue Buffer Release Control Register

This register controls the behavior of reception of R2R Queue transactions based on RapidIO transaction priority and R2R Queue fill level.

If this register must be changed during normal operation of the part, follow the procedure in "Synchronization of Data Path Control Register Changes".

Register name: SREP_NWR_ERR_BREL **Reset value: 0x00FF_0000**	**Register offset: 6F4**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	REL_MGMT_EN	REL_TO_MODE	Reserved		TO_CNT			
08:15	REL_MGMT_TO							
16:23	Reserved		REL_MGMT_RES					
24:31	Reserved		REL_MGMT_STOP					

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	REL_MGMT_EN	Buffer Release Management Enable	R/W	0
1	REL_TO_MODE	Buffer Release Timeout Mode Control 0 = After a timeout, do not engage buffer release management until the number of free buffers reaches the REL_MGMT_RES level. 1 = After a timeout, engage buffer release management and restart the timeout if the number of free buffers ever reaches the REL_MGMT_STOP level.	R/W	0
2:3	Reserved	N/A	R	0
4:7	TO_CNT	Count of the number of times the release management timeout period has expired. This counter rolls over to 0 once it reaches its maximum value.	R	0
8:15	REL_MGMT_TO	Timeout value 0 = TIMEOUT is disabled (this is not recommended). 1 = 0xFF - Maximum amount of time that can expire before resuming reporting of buffer fill levels. It is recommended that this timeout period be set based on the transmission period for REL_MGMT_STOP-REL_MGMT_RES maximum sized packets.	R/W	0xFF
16:17	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
18:23	REL_MGMT_RES	Buffer Release Resume Level. The number of buffers that must be free before free buffer reporting can resume. REL_MGMT_RES must be greater than REL_MGMT_STOP.	R/W	0
24:25	Reserved	N/A	R	0
26:31	REL_MGMT_STOP	Buffer Release Stop Level. The number of buffers that must be free when free buffer reporting must stop. REL_MGMT_RES must be greater than REL_MGMT_STOP.	R/W	0

Do not program this register in traffic; only program these registers after reset.

26.10.29 SREP B2S Buffer Release Control Register

This register controls when the SREP attempts to send packets to the Internal Switch Port see “RapidIO Physical Layer Flow Control”).

Register name: SREP_B2S_BREL **Reset value: 0x00FF_0000**	**Register offset: 6FC**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	REL_MGMT_EN	REL_TO_MODE	Reserved		TO_CNT			
08:15	REL_MGMT_TO							
16:23	Reserved				REL_MGMT_RES			
24:31	Reserved				REL_MGMT_STOP			

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	REL_MGMT_EN	Buffer Release Management Enable	R/W	0
1	REL_TO_MODE	Buffer Release Timeout Mode Control 0 = After a timeout, do not engage buffer release management until the buffer fill level reaches the REL_MGMT_RES level. 1 = After a timeout, engage buffer release management and restart the timeout if the buffer fill level ever increases before reaching the REL_MGMT_RES level and the fill level is >= REL_MGMT_STOP.	R/W	0
2:3	Reserved	N/A	R	0
4:7	TO_CNT	Count of the number of times the release management timeout period has expired. This counter rolls over to 0 once it reaches its maximum value.	R	0
8:15	REL_MGMT_TO	Timeout value. 0 = Timeout is disabled (this is not recommended). 1 = 0xFF - Maximum amount of time that can expire before resuming reporting of buffer fill levels. Each tick in this field is equivalent to 32 Reference Clock periods. It is recommended that this timeout period be set based on the transmission period for REL_MGMT_STOP-REL_MGMT_RES maximum sized packets.	R/W	0xFF
16:19	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
20:23	REL_MGMT_RES	Buffer Release Resume Level. REL_MGMT_RES must be greater than REL_MGMT_STOP. For other restrictions on the value of this register field, see "I2R Watermark and Buffer Release Management Register Value Restrictions".	R/W	0
24:27	Reserved	N/A	R	0
28:31	REL_MGMT_STOP	Buffer Release Stop Level. REL_MGMT_RES must be greater than REL_MGMT_STOP. For other restrictions on the value of this register field, see "I2R Watermark and Buffer Release Management Register Value Restrictions".	R/W	0

Do not program this register in traffic; only program these registers after reset.

26.10.30 SREP R2I Event Status Register

This register contains indications of R2I events. If any of the bit fields beginning with "L_" are set, the 'STAT_EVENT' bit in the "SREP Logical and Transport Layer Error Detect CSR" is set.

Reset values stated are for a powerup reset only. Non-powerup resets do not reset this register.

Register name: SREP_R2I_EVENT_STATUS **Reset value: 0x0400_0000**	**Register offset: 700**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	DB_NO_ACK	MNWR_NO_ACK	NO_DECOMP_BUFF	NO_RIO_TID	NO_BISF_TID	NUF_DECOMP	NUF_VALID	Reserved
08:15	Reserved							
16:23	Reserved							
24:31	Reserved				L_LUT_BND	L_BAD_REG_ACC	L_PKT_CRC	L_PKT_STOMP

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	DB_NO_ACK	This bit indicates if there are any unacknowledged I2R Doorbell requests outstanding. 0 = All doorbell requests are completed 1 = At least one doorbell request has not yet received a DONE or ERROR response Note: This bit does not cause R2I error information to be locked, nor does it cause a RIO_LOG event to occur in the "SREP Interrupt Status Register".	R	0
1	MNWR_NO_ACK	This bit indicates if there are any unacknowledged I2R NWRITE_R or Maintenance Write requests outstanding. 0 = All NWRITE_R and Maintenance Write requests are completed 1 = At least one NWRITE_R or Maintenance Write request has not yet received a DONE or ERROR response Note: This bit does not cause R2I error information to be locked, nor does it cause a RIO_LOG event to occur in the "SREP Interrupt Status Register".	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
2	NO_DECOMP_BUFF	This bit indicates if the decomposed transaction buffers are all in use. The number of decomposed transaction buffers is controlled by the DECOMP field of "SREP R2I Watermarks Register". 0 = At least one decomposed transaction buffer is available 1 = The decomposed transaction buffers are exhausted. Note: If this field is 1, then no transactions that require decomposition can be processed in the I2R direction (see "Bridge ISF-to-RapidIO Request Segmentation And Reassembly"). Note: This bit does not cause R2I error information to be locked, nor does it cause a RIO_LOG event to occur in the "SREP Interrupt Status Register".	R	0
3	NO_RIO_TID	This bit indicates if the 256 RapidIO transaction IDs are all in use. 0 = At least one RapidIO transaction ID is available 1 = All 256 RapidIO transaction IDs are exhausted Note: If this field is 1, then no transactions that require responses (Maintenance Reads/Writes, NREAD, NWRITE_R, Doorbell) can be processed in the I2R direction (see "Bridging ISF Requests to RapidIO"). This bit does not cause R2I error information to be locked, nor does it cause a RIO_LOG event to occur in the "SREP Interrupt Status Register".	R	0
4	NO_BISF_TID	This bit indicates if the 16 Bridge ISF transaction IDs are all in use. 0 = At least one transaction ID is available 1 = All 16 Bridge ISF transaction IDs are exhausted Note: If this field is 1, then no ISF requests that require Bridge ISF responses (Read) can be processed in the R2I direction (see "Bridging Logical I/O Requests to the Bridge ISF"). This bit does not cause R2I error information to be locked, nor does it cause a RIO_LOG event to occur in the "SREP Interrupt Status Register".	R	0
5	NUF_DECOMP	This bit indicates whether the transaction at the head of the I2R queue has enough DECOMP transaction buffers to proceed. 0 = There are not enough DECOMP transaction buffers 1 = There are enough DECOMP transaction buffers This bit is only valid when NUF_VALID is 1. Note: This bit does not cause R2I error information to be locked, nor does it cause a RIO_LOG event to occur in the "SREP Interrupt Status Register".	R	1

(Continued)

Bits	Name	Description	Type	Reset Value
6	NUF_VALID	This bit indicates when NUF_DECOMP is a valid indication. 0 = NUF_DECOMP does not apply to the transaction at the head of the I2R queue 1 = NUF_DECOMP does apply to the transaction at the head of the I2R queue Note: This bit does not cause R2I error information to be locked, nor does it cause a RIO_LOG event to occur in the "SREP Interrupt Status Register".	R	0
7:27	Reserved	N/A	R	0
28	L_LUT_BND	A request is received that crosses a LUT Entry boundary (see "Logical I/O Packet Events").	R/W1CS	0
29	L_BAD_REG_ACC	A RapidIO request greater than 4 bytes is mapped to a register read or write. Also detected when a write request was received from an illegal Source ID (see "RapidIO Illegal Register Access" and "Logical I/O Packet Events").	R/W1CS	0
30	L_PKT_CRC	A packet that was being received by the logical layer had a CRC error detected (see "R2I CRC Error Event").	R/W1CS	0
31	L_PKT_STOMP	A packet that was being received by the logical layer was stomped by the physical layer (see "R2I Stomped Packet Event").	R/W1CS	0

26.10.31 SREP R2I Event Status Logging Enable Register

This register controls which of the status events locks the Logical/Transport Layer Error Detect and Capture registers. Note that packet stomp and CRC events cannot lock the Logical/Transport Layer Error Detect and Capture registers.

Reset values stated are for a powerup reset only. Non-powerup resets do not reset this register.

Register name: SREP_R2I_EVENT_STATUS_LOG_EN **Reset value: 0x0000_0000**	**Register offset: 704**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							
24:31	Reserved				LUT_BND_EN	BAD_REG_ACC_EN	PKT_CRC_EN	PKT_STOMP_EN

Bits	**Name**	**Description**[a]	**Type**	**Reset Value**
0:27	Reserved	N/A	R	0
28	LUT_BND_EN	LUT Entry Boundary Crossing Event Logging Enable 0 = disable L_LUT_BND 1 = enable L_LUT_BND	R/WS	0
29	BAD_REG_ACC_EN	Register Access Error Event Logging Enable 0 = disable L_BAD_REG_ACC 1 = enable L_BAD_REG_ACC	R/WS	0
30	PKT_CRC_EN	Packet CRC Event Notification Enable 0 = disable L_PKT_CRC 1 = enable L_PKT_CRC	R/WS	0
31	PKT_STOMP_EN	Packet Stomp Event Notification Enable 0 = disable L_PKT_STOMP 1 = enable L_PKT_STOMP	R/WS	0

a. All 0 and 1 bits in this register enable bits in “SREP R2I Event Status Register”.

26.10.32 SREP R2I Error ISF Command Attributes Capture CSR

This register contains the Bridge ISF command attributes for the transaction in the "SREP Logical and Transport Layer Error Detect CSR" that caused the error.

This register is locked when a RapidIO Logical error is detected and the corresponding enable bit is set. The contents of this register are valid only for some errors detected in the "SREP Logical and Transport Layer Error Detect CSR" (see "Event Summary").

Reset values stated are for a powerup reset only. Non-powerup resets do not reset this register.

Register name: SREP_R2I_ISF_LOG_ERR_CMD_ATTR **Reset value: 0x0000_0000**	**Register offset: 708**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved				ISF_CMD			
16:23	Reserved					ISF_RSP_SIZE		
24:31	ISF_RSP_SIZE							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:11	Reserved	N/A	R	0
12:15	ISF_CMD	Bridge ISF Command code for R2I transaction that failed. For a list of ISF command codes, see "RapidIO Packet Mapping to Bridge ISF Transaction Types".	R/WS	0
16:20	Reserved	N/A	R	0
21:31	ISF_RSP_SIZE	The Bridge ISF RSP_SIZE has different meanings depending on the ISF CMD. • Mem read word: ISF_RSP_SIZE[10:8] = 3'b0, ISF_RSP_SIZE[7:0] = byte enable, active high • Mem read block that not cross 8 byte boundary: ISF_RSP_SIZE[10:8] = 3'b111, ISF_RSP_SIZE[7:0] = byte enable, active high • Mem read block that cross 8 byte boundary: the byte count of the response packet. Maximum is 0x400 (1024). When ISF_CMD is memory write block or memory write burst this field is not relevant.	R/WS	0

26.10.33 SREP R2I Error ISF Logical Error Decomposition Attributes Capture CSR

This register contains Bridge ISF request decomposition information for the RapidIO response in the "SREP Logical and Transport Layer Error Detect CSR" that caused the error.

This register is locked when a RapidIO Logical error is detected and the corresponding enable bit is set. The contents of this register are valid only for some errors detected in the "SREP Logical and Transport Layer Error Detect CSR" (see "Event Summary").

This register is useful in debugging partial failures of decomposed I2R transactions, when some of the decomposed transactions succeeded and some failed.

Reset values stated are for a powerup reset only. Non-powerup resets do not reset this register.

Register name: SREP_R2I_ISF_LOG_ERR_DC_ATTR **Reset value: 0x0000_0000**	**Register offset: 70C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	DECOMP	Reserved					DATA_OFFSET	
08:15	DATA_OFFSET							
16:23	Reserved							DECOMP_IDX
24:31	DECOMP_IDX							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	DECOMP	Indicates whether the response was part of a decomposed Bridge ISF request. 0 = Response was not a part of a decomposed request 1 = Response was a part of a decomposed request	R/WS	0
1:5	Reserved	N/A	R	0
6:15	DATA_OFFSET	The byte offset of the decomposed response data for the original Bridge ISF request.	R/WS	0
16:22	Reserved	N/A	R	0
23:31	DECOMP_IDX	The index of the decomposed response for the original Bridge ISF request.	R/WS	0

26.10.34 SREP Logical and Transport Layer Error Clear CSR

This register is a ‘Write 1 to Clear’ version of the “SREP Logical and Transport Layer Error Detect CSR”. It is created to allow a consistent programming model for all logical layer errors.

The status of bits in the “SREP Logical and Transport Layer Error Detect CSR” is indicated in this register. Reset values stated are for a powerup reset only. Non-powerup resets do not reset this register.

Register name: SREP_RIO_LIO_ERR_CLR **Reset value: 0x0000_0000**	**Register offset: 71C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	CL_ERR_RESP	CL_DB_ERR_RESP	Reserved		CL_ILL_TRANS	CL_ILL_TARG	Reserved	CL_RESP_TO
08:15	CL_UNEXP_RESP	CL_UNSUP_TRANS	Reserved					
16:23	Reserved							
24:31	STAT_EVENT	L_ISF_ERR	CL_R2I_TTL	CL_SPOOF	CL_R2I_PERR	CL_OOB	CL_NO_WR	CL_NO_RD

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	CL_ERR_RESP	Clear the L_ERR_RESP bit in the “SREP Logical and Transport Layer Error Detect CSR”.	R/W1CS	0
1	CL_DB_ERR_RESP	Clear the L_DB_ERR_RESP bit in the “SREP Logical and Transport Layer Error Detect CSR”.	R/W1CS	0
2:3	Reserved	N/A	R	0
4	CL_ILL_TRANS	Clear the L_ILL_TRANS bit in the “SREP Logical and Transport Layer Error Detect CSR”.	R/W1CS	0
5	CL_ILL_TARG	Clear the L_ILL_TARG bit in the “SREP Logical and Transport Layer Error Detect CSR”.	R/W1CS	0
6	Reserved	N/A	R	0
7	CL_RESP_TO	Clear the L_RESP_TO bit in the “SREP Logical and Transport Layer Error Detect CSR”.	R/W1CS	0
8	CL_UNEXP_RESP	Clear the L_UNEXP_RESP bit in the “SREP Logical and Transport Layer Error Detect CSR”.	R/W1CS	0
9	CL_UNSUP_TRANS	Clear the L_UNSUP_TRANS bit in the “SREP Logical and Transport Layer Error Detect CSR”.	R/W1CS	0
10:23	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
24	STAT_EVENT	An enabled status event has occurred in the "SREP R2I Event Status Register" (see "Logical I/O Packet Events").	RS	0
25	L_ISF_ERR	An error is detected in the "SREP ISF Logical Error Detect CSR". This bit is cleared when no errors are asserted in the "SREP ISF Logical Error Detect CSR". Control of transmission of port-writes or assertion of interrupts for this error is performed through the ISF Logical Error related registers, starting with "SREP ISF Logical Error Detect CSR".	RS	0
26	CL_R2I_TTL	Clear the L_R2I_TTL bit in the "SREP Logical and Transport Layer Error Detect CSR".	R/W1CS	0
27	CL_SPOOF	Clear the L_SPOOF bit in the "SREP Logical and Transport Layer Error Detect CSR".	R/W1CS	0
28	CL_R2I_PERR	Clear the L_R2I_PERR bit in the "SREP Logical and Transport Layer Error Detect CSR".	R/W1CS	0
29	CL_OOB	Clear the L_OOB bit in the "SREP Logical and Transport Layer Error Detect CSR".	R/W1CS	0
30	CL_NO_WR	Clear the L_NO_WR bit in the "SREP Logical and Transport Layer Error Detect CSR".	R/W1CS	0
31	CL_NO_RD	Clear the L_NO_RD bit in the "SREP Logical and Transport Layer Error Detect CSR".	R/W1CS	0

26.10.35 SREP R2I BAR and LUT Parity Error Status Register

This register gives the status of BAR and LUT parity errors. All bits in this register are sticky - they are preserved over all resets except power-up. For information on the use of these registers, see "Bridging Logical I/O Requests to the Bridge ISF".

Register name: SREP_R2I_PERR_STAT Reset value: 0x0000_0000	Register offset: 720

Bits	0	1	2	3	4	5	6	7
00:07	BAR_PERR_RD	BAR_PERR	LUT_PERR_RD	LUT_PERR	LCS_PERR	SYNC_PERR	Reserved	
08:15	Reserved					BAR_IDX		
16:23	Reserved							
24:31	LUT_IDX							

Bits	Name	Description	Type	Reset Value
0	BAR_PERR_RD	BAR Parity Error on Register Read 0 = No BAR parity error, or BAR parity error was detected as a result of packet bridging to the Bridge ISF 1 = BAR parity error detected during a register read of the BAR registers. To clear this bit, clear the L_R2I_PERR bit in the "SREP Logical and Transport Layer Error Detect CSR".	RS	0
1	BAR_PERR	BAR Parity Error 0 = No BAR parity error 1 = BAR parity error detected during a register read of the BAR registers or during packet bridging Check LCS_PERR to see if the parity error was detected in the "SREP Local Configuration Space Base Address CSR". If LCS_PERR is 0, this bit indicates that a parity error occurred in either "SREP R2I Base Address Register x LUT Control CSR" or "SREP R2I Base Address Register x Lower". To clear this bit, clear the L_R2I_PERR bit in the "SREP Logical and Transport Layer Error Detect CSR".	RS	0
2	LUT_PERR_RD	LUT Parity Error on Register Read 0 = No LUT parity error, or LUT parity error was detected as a result of packet bridging to the Bridge ISF 1 = LUT parity error detected during a register read of the LUT registers. To clear this bit, clear the L_R2I_PERR bit in the "SREP Logical and Transport Layer Error Detect CSR".	RS	0

(Continued)

Bits	Name	Description	Type	Reset Value
3	LUT_PERR	LUT Parity Error 0 = No LUT parity error, 1 = LUT parity error detected during a register read of the LUT registers or during packet bridging to the Bridge ISF To clear this bit, clear the L_R2I_PERR bit in the "SREP Logical and Transport Layer Error Detect CSR".	RS	0
4	LCS_PERR	When LCS_PERR is 1, it indicates that the "SREP Local Configuration Space Base Address CSR" has a parity error. To clear this bit, clear the L_R2I_PERR bit in the "SREP Logical and Transport Layer Error Detect CSR".	RS	0
5	SYNC_PERR	This field controls no functionality in the Tsi620.	RS	0
6:12	Reserved	N/A	R	0
13:15	BAR_IDX	When BAR_PERR is 1 and LCS_PERR is 0, this is the index of the R2I BAR entry in which the BAR parity error occurred. Otherwise this field contains 0.	RS	0
16:23	Reserved	N/A	R	0
24:31	LUT_IDX	When LUT_PERR is 1, this is the index of the LUT entry that the LUT parity error occurred in. Otherwise this field contains 0.	RS	0

26.10.36 SREP Port-Write Receive Status Register

This register indicates the status of port-write transactions received (see "Receiving Port-Writes").

Register name: SREP_PW_RX_STATUS Reset value: 0x0000_0000	Register offset: 724

Bits	0	1	2	3	4	5	6	7
00:07	Reserved					PW1_SIZE		
08:15	PW1_FAKE	Reserved					PW1_ERR	PW1_RXD
16:23	Reserved					PW0_SIZE		
24:31	PW0_FAKE	PW_DISC	Reserved				PW0_ERR	PW0_RXD

Bits	Name	Description	Type	Reset Value
0:4	Reserved	N/A	R	0
5:7	PW1_SIZE	This is the size of the port-write, as received by the SREP, encoded as below. Only valid when PW1_RXD is 1. 0x0 = 4 bytes 0x1 = 8 bytes 0x2 = 16 bytes 0x3 = 24 bytes 0x4-0x7 = 32 bytes Note: If the port-write received has more than 32 bytes of data, the first 32 bytes are captured and PW1_SIZE is set to 0x4. Setting PW1_SIZE to 0x5-0x7 may be completed by software, but does not affect the operation of the port-write receive logic.	R/WS	0
8	PW1_FAKE	Set the PW1_RXD bit to test software reception of a port-write.	R/W1S	0
9:13	Reserved	N/A	R	0
14	PW1_ERR	The port-write received was not 4 bytes or a multiple of 8 bytes.	RS	0
15	PW1_RXD	A port-write is received in the Port-Write 1 buffer. The port-write payload is captured in the "SREP Port-Write 1 Receive Buffer n Registers". Note: Clearing PW_RXD also clears PW_DISC and PW1_ERR.	R/W1CS	0
16:20	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
21:23	PW0_SIZE	This is the size of the port-write, as received by the SREP, encoded as below. Only valid when PW0_RXD is 1. 0x0 = 4 bytes 0x1 = 8 bytes 0x2 = 16 bytes 0x3 = 24 bytes 0x4-0x7 = 32 bytes Note: If the port-write received has more than 32 bytes of data, the first 32 bytes are captured and PW0_SIZE is set to 0x4. Setting PW0_SIZE to 0x5-0x7 may be completed by software, but does not affect the operation of the port-write receive logic.	R/WS	0
24	PW0_FAKE	Set the PW0_RXD bit to test software reception of a port-write.	R/W1S	0
25	PW_DISC	A port-write was received and discarded because the port-write buffer was occupied.	RS	0
26:29	Reserved	N/A	R	0
30	PW0_ERR	The port-write received into the port-write 0 buffer was not 4 bytes or a multiple of 8 bytes.	RS	0
31	PW0_RXD	A port-write is received in the port-write 0 buffer. The port-write payload is captured in the “SREP Port-Write 0 Receive Buffer n Registers”. Note: Clearing PW0_RXD also clears PW_DISC and PW0_ERR.	R/W1CS	0

26.10.37 SREP Port-Write 0 Receive Buffer n Registers

There are 8 "SREP Port-Write 0 Receive Buffer n Registers", allowing access to the 32 byte payload of a port-write packet received by the SREP. These registers are locked once a Port-Write is received.

Register name: SREP_RIO_PW0_RX_BUFF{0..7} Reset value: 0x0000_0000	Register offset: 730, 734, 738,..., 74C

Bits	0	1	2	3	4	5	6	7
00:07	PW_DATA							
08:15	PW_DATA							
16:23	PW_DATA							
24:31	PW_DATA							

Bits	Name	Description	Type	Reset Value
0:31	PW_DATA	Port-write payload data	R/WS	0

26.10.38 SREP Port-Write 1 Receive Buffer n Registers

There are 8 "SREP Port-Write 1 Receive Buffer n Registers", allowing access to the 32 byte payload of a port-write packet received by the SREP. These registers are locked once a Port-Write is received.

Register name: SREP_RIO_PW1_RX_BUFF{0..7} Reset value: 0x0000_0000	Register offset: 750, 754, 758,..., 76C

Bits	0	1	2	3	4	5	6	7
00:07	PW_DATA							
08:15	PW_DATA							
16:23	PW_DATA							
24:31	PW_DATA							

Bits	Name	Description	Type	Reset Value
0:31	PW_DATA	Port-write payload data	R/WS	0

26.10.39 SREP ISF Logical Error Detect CSR

This register indicates the error that was detected in/for a transaction received from the Bridge ISF. Multiple bits may get set in the register if simultaneous errors are detected during the same clock cycle that the errors are logged.

Note that events must be enabled in the "SREP ISF Logical Error Logging Enable CSR" in order for information to be logged for that event, and for notification (either port-write or interrupt) to occur.

Reset values stated are for a powerup reset only. Non-powerup resets do not reset this register..

Register name: SREP_ISF_LIO_ERR_DET **Reset value: 0x0000_0000**	**Register offset: 780**

Bits	0	1	2	3	4	5	6	7
00:07	I_ERR_RESP	Reserved						I_RESP_TO
08:15	I_UNEXP_RESP	I_UNSUP_TRANS	Reserved					
16:23	I_BE_DISCONT	Reserved		I_ECC_ERR	I_R2R_TTL	I_REG_TTL	I_I2R_TTL	I_TEA
24:31	I_LUT_BND	I_I2R_PERR	Reserved		I_UNSAR_REQ	I_OOB	I_NO_WR	I_NO_RD

Bits	Name	Description	Type	Reset Value
0	I_ERR_RESP	Error Response Bit is set when a Bridge ISF Error response is received for an outstanding request.	R/W1CS	0
1:6	Reserved	N/A	R	0
7	I_RESP_TO	Response Timeout Bit is set when no response is received for a request within a specified timeout period (see "Bridge ISF Error Conditions").	R/W1CS	0
8	I_UNEXP_RESP	Unexpected Response Bit is set when a Bridge ISF response was received that does not match an outstanding request.	R/W1CS	0
9	I_UNSUP_TRANS	Unsupported Transaction Bit is set when a packet was received that is an unsupported transaction (see "Bridge ISF Error Conditions").	R/W1CS	0
10:15	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
16	I_BE_DISCONT	Byte Enables Discontiguous Bit is set when a Bridge ISF Write Burst transaction with discontiguous byte enables is being SARed as an ISF Write Block transaction (see "Bridge ISF Error Conditions").	R/W1CS	0
17:18	Reserved	N/A	R	0
19	I_ECC_ERR	An ECC error is detected in the data path (see "ECC Error in Data Path"). Clearing this bit also clears the UNCORR, ISF_REQ, ECC_INB, ECC_OUTB, and ECC_ISF bits in the "SREP ISF ECC Error Status Register".	R/W1CS	0
20	I_R2R_TTL	An R2R Queue transaction could not be transmitted from the R2R queues to the physical layer before the transactions Time-to-Live period, as programmed in the "SREP I2R Transaction Time-To-Live Register", has expired.	R/W1CS	0
21	I_REG_TTL	A register response transaction could not be transmitted from the Register Access Response queues to the physical layer before the transactions Time-to-Live period, as programmed in the "SREP I2R Transaction Time-To-Live Register", has expired.	R/W1CS	0
22	I_I2R_TTL	A transaction could not be transmitted from the I2R Data Buffers/Header queues to the physical layer before the transactions Time-to-Live period, as programmed in the "SREP I2R Transaction Time-To-Live Register", has expired.	R/W1CS	0
23	I_TEA	The Bridge ISF completed a transaction with an error acknowledge (see "Bridge ISF Error Conditions").	R/W1CS	0
24	I_LUT_BND	A request is received that crosses a Bridge ISF LUT Entry boundary (see "Bridge ISF Error Conditions").	R/W1CS	0
25	I_I2R_PERR	A Parity Error is detected in the Bridge ISF-to-RapidIO BARs and/or LUTs (see "Bridge ISF Error Conditions"). Clearing this bit clears all bits except LUT_IDX and BAR_IDX in the "SREP I2R BAR and LUT Parity Error Status Register". This bit has special behavior when the registers are locked (see "R2I LUT and BAR Parity Error Information Latching").	R/W1CS	0
26:27	Reserved	N/A	R	0
28	I_UNSAR_REQ	A Bridge ISF request is mapped to a Maintenance read or write or a Doorbell request is not 4 bytes or at a 4-byte aligned address boundary. Also detected when a Bridge ISF transaction hits in the Doorbell BAR but cannot be translated to a RapidIO Doorbell request (see "Bridge ISF Error Conditions").	R/W1CS	0
29	I_OOB	A request was received for an address that does not appear in any of the enabled I2R BARs or the Doorbell BAR (see "Bridge ISF Error Conditions").	R/W1CS	0

(Continued)

Bits	Name	Description	Type	Reset Value
30	I_NO_WR	A write request was received for an area of memory that does not have write permissions enabled (see "Bridge ISF Error Conditions").	R/W1CS	0
31	I_NO_RD	A read request was received for an area of memory that does not have read permissions enabled (see "Bridge ISF Error Conditions").	R/W1CS	0

26.10.40 SREP ISF Logical Error Logging Enable CSR

This register controls whether or not an event locks the ISF Logical Error Detect and Capture registers. Note that events must be enabled in the "SREP ISF Logical Error Logging Enable CSR" in order for information to be logged for that event, and for notification (either port-write or interrupt) to occur. Reset values stated are for a powerup reset only. Non-powerup resets do not reset this register.

Register name: SREP_ISF_ERR_DET_EN **Reset value: 0x0000_0000**	**Register offset: 784**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	ERR_RESP_EN	Reserved						RESP_TO_EN
08:15	UNEXP_RESP_EN	UNSUP_TRANS_EN	Reserved					
16:23	BE_DISCONT_EN	Reserved		ECC_ERR_EN	R2R_TTL_EN	REG_TTL_EN	I2R_TTL_EN	I_TEA_EN
24:31	LUT_BND_EN	I2R_PERR_EN	Reserved		UNSAR_REQ_EN	OOB_EN	NO_WR_EN	NO_RD_EN

Bits	**Name**	**Description**[a]	**Type**	**Reset Value**
0	ERR_RESP_EN	Error Response Received Logging Enable 0 = disable I_ERR_RESP 1 = enable I_ERR_RESP	R/WS	0
1:6	Reserved	N/A	R	0
7	RESP_TO_EN	Response Timeout Logging Enable 0 = disable I_RESP_TO 1 = enable I_RESP_TO	R/WS	0
8	UNEXP_RESP_EN	Unexpected Response Logging Enable 0 = disable I_UNEXP_RESP 1 = enable I_UNEXP_RESP	R/WS	0
9	UNSUP_TRANS_EN	Unsupported Transaction Logging Enable 0 = disable I_UNSUP_TRANS 1 = enable I_UNSUP_TRANS	R/WS	0
10:15	Reserved	N/A	R	0
16	BE_DISCONT_EN	Byte Enables Discontiguous Logging Enable 0 = disable I_BE_DISCONT 1 = enable I_BE_DISCONT	R/WS	0
17:18	Reserved	N/A	R	0

(Continued)

Bits	Name	Description[a]	Type	Reset Value
19	ECC_ERR_EN	Data Path ECC Error Logging Enable 0 = disable information latching for I_ECC_ERR 1 = enable information latching for I_ECC_ERR	R/WS	0
20	R2R_TTL_EN	R2R Queue Time-to-Live Expired Logging Enable 0 = disable information latching for I_R2R_TTL 1 = enable information latching for I_R2R_TTL	R/WS	0
21	REG_TTL_EN	Register Response Time-to-Live Expired Logging Enable 0 = disable information latching for I_REG_TTL 1 = enable information latching for I_REG_TTL	R/WS	0
22	I2R_TTL_EN	I2R Transaction Time-to-Live Expired Logging Enable 0 = disable information latching for I_I2R_TTL 1 = enable information latching for I_I2R_TTL	R/WS	0
23	I_TEA_EN	ISF Transmission Error Acknowledge Logging Enable 0 = disable information latching for I_TEA 1 = enable information latching for I_TEA	R/WS	0
24	LUT_BND_EN	LUT Boundary Crossing Event Logging Enable 0 = disable I_LUT_BND 1 = enable I_LUT_BND	R/WS	0
25	I2R_PERR_EN	I2R LUT/BAR Parity Error Event Logging Enable 0 = disable I_I2R_PERR 1 = enable I_I2R_PERR	R/WS	0
26:27	Reserved	N/A	R	0
28	UNSAR_REQ_EN	Unsegmentable Request Logging Enable 0 = disable I_UNSAR_REQ 1 = enable I_UNSAR_REQ	R/WS	0
29	OOB_EN	Request Out of Bounds Logging Enable 0 = disable I_OOB 1 = enable I_OOB	R/WS	0
30	NO_WR_EN	No Write Permission Event Logging Enable 0 = disable I_NO_WR 1 = enable I_NO_WR	R/WS	0
31	NO_RD_EN	No Read Permission Event Logging Enable 0 = disable I_NO_RD 1 = enable I_NO_RD	R/WS	0

a. All 0 and 1 bits in this register enable bits in “SREP ISF Logical Error Detect CSR”.

26.10.41 SREP ISF Logical Error Upper Attributes Capture CSR

This register contains the most significant 24 bits of the Bridge ISF transaction header for the transaction that caused the error. It is locked when an ISF Logical error is detected and the corresponding enable bit is set. Reset values stated are for a powerup reset only. Non-powerup resets do not reset this register.

Register name: SREP_ISF_LOG_ERR_ATTR_U **Reset value: 0x0000_0000**	**Register offset: 788**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	ATTR_U							
16:23	ATTR_U							
24:31	ATTR_U							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:7	Reserved	N/A	R	0
8:31	ATTR_U	Most significant 24 bits of the Bridge ISF request transaction for which the error was detected (see “Bridging ISF Requests to RapidIO”).	R/WS	0

26.10.42 SREP ISF Logical Error Middle Attributes Capture CSR

This register contains bits 24 to 55 of the 88 bits of the Bridge ISF transaction header for the transaction that caused the error. It is locked when an ISF Logical error is detected and the corresponding enable bit is set.

Reset values stated are for a powerup reset only. Non-powerup resets do not reset this register.

Register name: SREP_ISF_LOG_ERR_ATTR_M **Reset value: 0x0000_0000**	**Register offset: 78C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	ATTR_M							
08:15	ATTR_M							
16:23	ATTR_M							
24:31	ATTR_M							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:31	ATTR_M	Bits 24 to 55 of the Bridge ISF transaction header for the transaction for which the error was detected (see "Bridging ISF Requests to RapidIO").	R/WS	0

26.10.43 SREP ISF Logical Error Lower Attributes Capture CSR

This register contains the least significant 32 bits of the Bridge ISF transaction header for the transaction that caused the error. It is locked when an ISF Logical error is detected and the corresponding enable bit is set.

Reset values stated are for a powerup reset only. Non-powerup resets do not reset this register.

Register name: SREP_ISF_LOG_ERR_ATTR_L **Reset value: 0x0000_0000**	**Register offset: 790**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	ATTR_L							
08:15	ATTR_L							
16:23	ATTR_L							
24:31	ATTR_L							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:31	ATTR_L	Least significant 32 bits of the Bridge ISF transaction header for which the error was detected (see “Bridging ISF Requests to RapidIO”).	R/WS	0

26.10.44 SREP ISF Logical Error RapidIO Routing Attributes Capture CSR

This register contains the RapidIO routing information associated with the Bridge ISF transaction that caused the error. It is locked when an ISF Logical error is detected and the corresponding enable bit is set.

To determine whether the SRC_ID and DEST_ID are 8 or 16 bits in size, see the TT field in the “SREP ISF Logical Error RapidIO Physical Attributes Capture CSR”.

This register is not valid for all ISF Logical errors (see “Event Summary”). Reset values stated are for a powerup reset only. Non-powerup resets do not reset this register.

Register name: SREP_ISF_LOG_ERR_RIO_RATTR **Reset value: 0x0000_0000**	**Register offset: 7A0**

Bits	0	1	2	3	4	5	6	7
00:07	SRC_ID							
08:15	SRC_ID							
16:23	DEST_ID							
24:31	DEST_ID							

Bits	Name	Description	Type	Reset Value
0:15	SRC_ID	Source ID of the RapidIO transaction associated with this error.	R/WS	0
16:31	DEST_ID	Destination ID to which the RapidIO transaction associated with this error was sent	R/WS	0

26.10.45 SREP ISF Logical Error RapidIO Physical Attributes Capture CSR

This register contains the RapidIO physical layer information associated with the Bridge ISF transaction that caused the error. It is locked when an ISF Logical error is detected and the corresponding enable bit is set.

This register is not valid for all ISF Logical errors (see "Event Summary"). Reset values stated are for a powerup reset only. Non-powerup resets do not reset this register.

Register name: SREP_ISF_LOG_ERR_RIO_PATTR **Reset value: 0x0000_0000**	**Register offset: 7A4**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	FTYPE				TTYPE			
08:15	Reserved							
16:23	Reserved							
24:31	Reserved		TT		Reserved	CRF	PRIO	

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:3	FTYPE	RapidIO Format type associated with the error	R/WS	0
4:7	TTYPE	RapidIO Transaction type associated with the error	R/WS	0
8:25	Reserved	N/A	R	0
26:27	TT	RapidIO TT code, which determines the size of the source/destination IDs (8 or 16 bit) captured in the "SREP ISF Logical Error RapidIO Routing Attributes Capture CSR".	R/WS	0
28	Reserved	N/A	R	0
29	CRF	The CRF bit setting of the RapidIO packet associated with the Bridge ISF transaction.	R/WS	0
30:31	PRIO	The Priority bit setting of the RapidIO packet associated with the Bridge ISF transaction.	R/WS	0

26.10.46 SREP ISF Logical Error RapidIO Lower Address Capture CSR

This register contains the RapidIO address associated with a Bridge ISF transaction that has had an error detected. Reset values stated are for a powerup reset only. Non-powerup resets do not reset this register.

Register name: SREP_ISF_LOG_ERR_RIO_ADDR **Reset value: 0x0000_0000**	**Register offset: 7AC**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	ADDRESS							
08:15	ADDRESS							
16:23	ADDRESS							
24:31	ADDRESS					Reserved	EXT_ADDR	

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:28	ADDRESS	Least significant 29 address bits.	R/WS	0
29	Reserved	N/A	R	0
30:31	EXT_ADDR	Extended address bits (most significant 2 bits of address field)	R/WS	0

26.10.47 SREP I2R BAR and LUT Parity Error Status Register

This register gives the status of BAR and LUT parity errors. All bits in this register are sticky - they are preserved over all resets except power-up. For information on the use of this register, see "Bridge ISF-to-RapidIO Parity Error".

Register name: SREP_I2R_PERR_STAT Reset value: 0x0000_0000	Register offset: 7B0

Bits	0	1	2	3	4	5	6	7
00:07	BAR_PERR_RD	BAR_PERR	LUT_PERR_RD	LUT_PERR	DB_BAR	Reserved		
08:15	Reserved					BAR_IDX		
16:23	Reserved							
24:31	LUT_IDX							

Bits	Name	Description	Type	Reset Value
0	BAR_PERR_RD	BAR Parity Error on Register Read 0 = No BAR parity error, or BAR parity error was detected as a result of transaction bridging to RapidIO 1 = BAR parity error detected during a register read of the BAR register. To clear this bit, clear the I_I2R_PERR bit in the "SREP ISF Logical Error Detect CSR".	RS	0
1	BAR_PERR	BAR Parity Error Detected 0 = No BAR parity error 1 = BAR parity error was detected either during packet transfer or during a register read of the BAR registers Check the DB_BAR bit to see if the Doorbell BAR incurred a parity error. If DB_BAR is 0, this bit indicates that there was a parity error in at least one of "SREP I2R Base Address Register x LUT Entry CSR", "SREP I2R Base Address Register x Upper", or "SREP I2R Base Address Register x Lower". To clear this bit, clear the I_I2R_PERR bit in the "SREP ISF Logical Error Detect CSR".	RS	0
2	LUT_PERR_RD	LUT Parity Error on Register Read 0 = No LUT parity error, or LUT parity error was detected as a result of transaction bridging to RapidIO 1 = LUT parity error detected during a register read of the LUT registers. To clear this bit, clear the I_I2R_PERR bit in the "SREP ISF Logical Error Detect CSR".	RS	0

(Continued)

Bits	Name	Description	Type	Reset Value
3	LUT_PERR	LUT Parity Error Detected 0 = No LUT parity error 1 = LUT parity error was detected either during transaction bridging or during a register read of the LUT registers To clear this bit, clear the I_I2R_PERR bit in the "SREP ISF Logical Error Detect CSR".	RS	0
4	DB_BAR	When BAR_PERR is 1, this indicates that either the "SREP I2R Doorbell BAR Upper" or "SREP I2R Doorbell BAR Lower" had a parity error. 0 = No parity error in the "SREP I2R Doorbell BAR Upper" or "SREP I2R Doorbell BAR Lower" 1 = Parity error detected in the "SREP I2R Doorbell BAR Upper" or "SREP I2R Doorbell BAR Lower" To clear this bit, clear the I_I2R_PERR bit in the "SREP ISF Logical Error Detect CSR".	RS	0
5:12	Reserved	N/A	R	0
13:15	BAR_IDX	When BAR_PERR is 1 and DB_BAR is 0, this is the index of the I2R BAR entry in which the parity error occurred. Otherwise, this field is 0.	RS	0
16:23	Reserved	N/A	R	0
24:31	LUT_IDX	When LUT_PERR is 1, this is the index of the LUT entry that the LUT parity error occurred in. Otherwise, this field is 0.	RS	0

26.10.48 SREP ISF ECC Error Status Register

This register contains status and error information for ECC errors detected in the data path. Reset values stated are for a powerup reset only. Non-powerup resets do not reset this register.

If ECC errors occur in consecutive datums, an indeterminate mix of information from both errors will be latched in this register.

Register name: SREP_ISF_ECC_STAT **Reset value: 0x0000_0000**	**Register offset: 7B4**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	ECC_SYN_M							
08:15	ECC_SYN_L							
16:23	UNCORR	ISF_REQ	Reserved			ECC_INB	ECC_OUTB	ECC_ISF
24:31	Reserved			BUFF_IDX				

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:7	ECC_SYN_M	The ECC syndrome associated with the ECC error detected in the Bridge ISF Header, or in unused ISF bits during transfer of the data of a transaction. For correctable errors, the ECC syndrome indicates which bit was faulty. Note: The ECC syndrome of an uncorrectable error cannot be used to determine which bits are faulty.	RS	0
8:15	ECC_SYN_L	The ECC syndrome associated with the ECC error detected in the data. For correctable errors, the ECC syndrome indicates which bit was faulty. Note: The ECC syndrome of an uncorrectable error cannot be used to determine which bits are faulty	RS	0
16	UNCORR	Indicates if the error was correctable or uncorrectable 0 = Correctable ECC error 1 = Uncorrectable ECC error Note: This bit is cleared when the I_ECC_ERR bit is cleared in the "SREP ISF Logical Error Detect CSR".	RS	0

(Continued)

Bits	Name	Description	Type	Reset Value
17	ISF_REQ	Indicates if that the error was detected in the header information (upper bits of a response and/or address bits of a read/write request) of a Bridge ISF request. If this bit is set, only the information about the request is latched. 0 = ECC error was not detected in the header information (upper bits of a response and/or address bits of a read/write request) of a Bridge ISF transaction. 1 = ECC error was detected in the header information (upper bits of a response and/or address bits of a read/write request) of a Bridge ISF request. Note: This bit is cleared when the I_ECC_ERR bit is cleared in the "SREP ISF Logical Error Detect CSR".	RS	0
18:20	Reserved	N/A	R	0
21	ECC_INB	An ECC error was detected for data that was read out of the R2I buffers. Only one of ECC_ISF, ECC_OUTB or ECC_INB is set at one time. To clear this bit, write 1 to the I_ECC_ERR bit in "SREP ISF Logical Error Detect CSR".	RS	0
22	ECC_OUTB	An ECC error was detected for data that was read out of the I2R buffers. Only one of ECC_ISF, ECC_OUTB or ECC_INB is set at one time. To clear this bit, write 1 to the I_ECC_ERR bit in "SREP ISF Logical Error Detect CSR".	RS	0
23	ECC_ISF	An ECC error was detected for data that was received from the Bridge ISF. Only one of ECC_ISF, ECC_OUTB or ECC_INB is set at one time. To clear this bit, write 1 to the I_ECC_ERR bit in "SREP ISF Logical Error Detect CSR". If the data corrupted was the Bridge ISF Request Header, then the ISF_REQ bit is set when ECC_ISF is set.	RS	0
24:26	Reserved	N/A	R	0
27:31	BUFF_IDX	The Index of the I2R or R2I buffer that contains the datum which has bad ECC. This field is valid only when one of ECC_OUTB or ECC_INB is set.	RS	0

26.10.49 SREP ISF Logical Error Generate CSR

This register allows Bridge ISF events to be generated for purposes of software testing (see "Debug Support"

Register name: SREP_ISF_LIO_ERR_GEN Reset value: 0x0000_0000	Register offset: 7BC

Bits	0	1	2	3	4	5	6	7
00:07	I_ERR_RESP	Reserved						I_RESP_TO
08:15	I_UNEXP_RESP	I_UNSUP_TRANS	Reserved					
16:23	I_BE_DISCONT	Reserved		I_ECC_ERR	I_R2R_TTL	I_REG_TTL	I_I2R_TTL	I_TEA
24:31	I_LUT_BND	I_I2R_PERR	Reserved		I_UNSAR_REQ	I_OOB	I_NO_WR	I_NO_RD

Bits	Name	Description	Type	Reset Value
0	I_ERR_RESP	Assert I_ERR_RESP in "SREP ISF Logical Error Detect CSR".	R/W1S	0
1:6	Reserved	N/A	R	0
7	I_RESP_TO	Assert I_RESP_TO in "SREP ISF Logical Error Detect CSR".	R/W1S	0
8	I_UNEXP_RESP	Assert I_UNEXP_RESP in "SREP ISF Logical Error Detect CSR".	R/W1S	0
9	I_UNSUP_TRANS	Assert I_UNSUP_TRANS in "SREP ISF Logical Error Detect CSR".	R/W1S	0
10:15	Reserved	N/A	R	0
16	I_BE_DISCONT	Assert I_BE_DISCONT in "SREP ISF Logical Error Detect CSR".	R/W1S	0
17:18	Reserved	N/A	R	0
19	I_ECC_ERR	Assert I_ECC_ERR in "SREP ISF Logical Error Detect CSR".	R/W1S	0
20	I_R2R_TTL	Assert I_R2R_TTL in "SREP ISF Logical Error Detect CSR".	R/W1S	0
21	I_REG_TTL	Assert I_REG_TTL in "SREP ISF Logical Error Detect CSR".	R/W1S	0
22	I_I2R_TTL	Assert I_I2R_TTL in "SREP ISF Logical Error Detect CSR".	R/W1S	0
23	I_TEA	Assert I_TEA in "SREP ISF Logical Error Detect CSR".	R/W1S	0
24	I_LUT_BND	Assert I_LUT_BND in "SREP ISF Logical Error Detect CSR".	R/W1S	0
25	I_I2R_PERR	Assert I_I2R_PERR in "SREP ISF Logical Error Detect CSR".	R/W1S	0

(Continued)

Bits	Name	Description	Type	Reset Value
26:27	Reserved	N/A	R	0
28	I_UNSAR_REQ	Assert I_UNSAR_REQ in “SREP ISF Logical Error Detect CSR”.	R/W1S	0
29	I_OOB	Assert I_OOB in “SREP ISF Logical Error Detect CSR”.	R/W1S	0
30	I_NO_WR	Assert I_NO_WR in “SREP ISF Logical Error Detect CSR”.	R/W1S	0
31	I_NO_RD	Assert I_NO_RD in “SREP ISF Logical Error Detect CSR”.	R/W1S	0

26.10.50 SREP ISF Response Timeout Register

This is a IDT-specific feature. If a Bridge ISF target does not return a response within the period programmed in this timer, the request is dropped. If the ERESP_DIS bit in the "SREP R2I RapidIO Miscellaneous Control CSR" is set to 0, then a RapidIO error response is sent for the request.

The response timeout error is reported through a port-write and/or an interrupt. If this register must be changed during normal operation of the part, follow the procedure described in "Synchronization of Data Path Control Register Changes".

Register name: SREP_ISF_RTO_CTL **Reset value: 0xFFFF_FF00**	**Register offset: 7C0**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	TVAL							
08:15	TVAL							
16:23	TVAL							
24:31	Reserved							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:23	TVAL	This timer runs on the RapidIO reference clock, which is either 156.25 or 125 MHz. When the RapidIO reference clock frequency is 156.25 MHz or 125 MHz, the timeout period is computed by: TVAL * RIO_REF_CLK/32. For a clock frequency of 156.25 MHz, the maximum timeout period is 3.4 seconds and the granularity is 204.8 nsec. For a clock frequency of 125 MHz, the maximum timeout period is 4.2 seconds and the granularity is 256 nsec. Note: If TVAL is programmed to 0, Bridge ISF response timeouts are disabled. Bridge ISF requests will wait forever for responses.	R/W	0xFFFFFF
24:31	Reserved	N/A	R	0

26.10.51 SREP ISF ECC Control Register

This register defines control functions for generation and check of ECC, the error detection/correction code accompanying each 8 bytes of payload data handled by the SREP.

Register name: SREP_ISF_ECC_CTL **Reset value: 0x0000_0000**	**Register offset: 7C8**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved							
16:23	Reserved			ECC_INV_TYPE	ECC_INV_EN	INB_ECC_CHK_EN	OUTB_ECC_CHK_EN	ISF_ECC_CHK_EN
24:31	ECC_INV_DLY							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:18	Reserved	N/A	R	0
19	ECC_INV_TYPE	Type of ECC corruption to insert: 0 = Invert the entire ECC code (uncorrectable error) 1 = Invert the least significant bit of the ECC code (correctable error)	R/W	0
20	ECC_INV_EN	Invert ECC after ECC_INV_DELAY datums are received from RapidIO. 0 = No ECC inversion operation is pending 1 = ECC inversion is pending This bit clears to zero after ECC_INV_DLY+1 datums are received from the Bridge ISF, and ECC is inverted.	R/W	0
21	INB_ECC_CHK_EN	Enable checking of ECC for data transferred from the R2I Data buffers into the Bridge ISF. 0 = Do not check ECC for the R2I data buffers 1 = Check ECC for the R2I data buffers.	R/W	0
22	OUTB_ECC_CHK_EN	Enable checking of ECC for data transferred from the I2R Data buffers into RapidIO. 0 = Do not check ECC for the R2I data buffers 1 = Check ECC for the R2I data buffers.	R/W	0
23	ISF_ECC_CHK_EN	Enable checking of ECC for data received from the Bridge ISF. 0 = Do not check ECC for the ISF request 1 = Check ECC for the ISF request.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
24:31	ECC_INV_DLY	Number of datums to wait before changing the ECC generated for 1 datum in the manner controlled by ECC_INV_TYPE. Note: An ECC error cannot be inserted in the first datum of an R2I ISF Response Packet (see "ECC Error in Data Path").	R/W	0

26.11 IDT Specific Registers – Doorbell Receive

The registers in this section are vendor specific. They are not defined in the *RapidIO Interconnect Specification (Revision 1.3)*. They are specific to IDT's design of a RapidIO endpoint.

Reserved bits should not be modified. Reserved register addresses should not be read or written.

These registers are reset when the SREP is reset. The registers are not accessible when the RapidIO port is in reset or powered down.

Table 175: Register Map for SREP IDT Specific Registers – Doorbell Receive

Offset	Register Name	See
0x800	SREP_RIO_RX_DB_CTRL	"SREP Doorbell Receive Control Register"
0x810	SREP_RIO_RX_DB_DATA1	"SREP Doorbell Receive Data 1 Register"
0x814	SREP_RIO_RX_DB_DATA2	"SREP Doorbell Receive Data 2 Register"

26.11.1 SREP Doorbell Receive Control Register

This register controls whether or not doorbell requests are accepted from RapidIO.

Register name: SREP_RIO_RX_DB_CTRL Reset value: 0x0000_0000	Register offset: 800

Bits	0	1	2	3	4	5	6	7
00:7	Reserved							
8:15	Reserved							
16:23	Reserved							
24:31	Reserved							DB_RX_EN

Bits	Name	Description	Type	Reset Value
0:30	Reserved	N/A	R	0
31	DB_RX_EN	Enable reception of doorbell requests 0 = No doorbell requests are accepted. Doorbell requests receive a response with an 'Error' status 1 = Doorbell requests are accepted.	R/W	0

26.11.2 SREP Doorbell Receive Data 1 Register

This register contains data about the doorbell request received. Interrupt handlers should read this register, and then the "SREP Doorbell Receive Data 2 Register", until the "SREP Doorbell Receive Data 1 Register".DB_VALID field is 0.

For more information on the use of this register, see "RapidIO Doorbell Request Handling".

Register name: SREP_RIO_RX_DB_DATA1 **Reset value: 0x0000_0000**	**Register offset: 810**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	DB_DATA							
8:15	DB_DATA							
16:23	Reserved							
24:31	DB_RTRD	DB_RIO_CRF	DB_RIO_PRIO		DB_RIO_TT		DB_FULL	DB_VALID

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:15	DB_DATA	This field contains the 16 bits of data in the RapidIO doorbell request.	R	0
16:23	Reserved	N/A	R	0
24	DB_RTRD	Doorbell Retried At least one Doorbell Request received by the SREP was retried since the last time a doorbell request was read from this register.	RC	0
25	DB_RIO_CRF	This field contains the CRF bit of the RapidIO doorbell request.	R	0
26:27	DB_RIO_PRIO	This field contains the RapidIO priority of the doorbell request. 0 = Priority 0 (Lowest) 1 = Priority 1 2 = Priority 2 3 = Reserved (not possible to send a Doorbell request at priority 3)	R	0
28:29	DB_RIO_TT	This field contains the RapidIO TT code of the doorbell request 0b00 = 8-bit destination IDs 0b01 = 16-bit destination IDs 0b10, 0b11 = Reserved	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
30	DB_FULL	Doorbell Queue is Full 0 = The Doorbell queue has less than the maximum number of doorbells in it. 1 = The Doorbell queue as the maximum number of doorbell requests captured. Check the DB_RTRY bit to see if doorbells are being retried. Note that this bit is cleared after a read of the "SREP Doorbell Receive Data 2 Register". It may be set again if a doorbell message is received between the time the bit is cleared and the next time that the "SREP Doorbell Receive Data 1 Register" is read.	R	0
31	DB_VALID	This field indicates if the doorbell entry is valid. 0 = Doorbell entry is not valid. There are no doorbell entries in the doorbell queue. 1 = Doorbell queue entry is valid. There may be more doorbell entries in the doorbell queue	R	0

26.11.3 SREP Doorbell Receive Data 2 Register

This register contains data about the doorbell request received. Interrupt handlers should read "SREP Doorbell Receive Data 1 Register", and then this register, until the "SREP Doorbell Receive Data 1 Register".DB_VALID field is 0.

For more information on the use of this register, see "RapidIO Doorbell Request Handling".

Register name: SREP_RIO_RX_DB_DATA2 **Reset value: 0x0000_0000**	**Register offset: 814**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	DB_RIO_SRC							
8:15	DB_RIO_SRC							
16:23	DB_RIO_DEST							
24:31	DB_RIO_DEST							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:15	DB_RIO_SRC	The source ID of the RapidIO entity that sent the doorbell request. The size of the sourceID is determined by the "SREP Doorbell Receive Data 1 Register".DB_RIO_TT field. Note that 8-bit source IDs are zero-extended in this field.	R	0
16:31	DB_RIO_DEST	The destination ID of the doorbell request. The size of the destination ID is determined by the "SREP Doorbell Receive Data 1 Register".DB_RIO_TT field. Note that 8 bits destination IDs are zero-extended in this field.	R	0

26.12 IDT Specific Registers – Statistics

The registers in this section are vendor specific. They are not defined in the *RapidIO Interconnect Specification (Revision 1.3)*. They are specific to IDT's design of a RapidIO endpoint.

Reserved bits should not be modified. Reserved register addresses should not be read or written.

These registers are reset when the SREP is reset. The registers are not accessible when the RapidIO port is in reset or powered down.

Table 176: Register Map for SREP IDT Specific Registers – Statistics

Offset	Register Name	See
0x940	SREP_ISF_PSC0n1_CTRL	"SREP ISF Performance Statistics Counter 0 and 1 Control Register"
0x944	SREP_ISF_PSC2n3_CTRL	"SREP ISF Performance Statistics Counter 2 and 3 Control Register"
0x948	SREP_ISF_PSC4n5_CTRL	"SREP ISF Performance Statistics Counter 4 and 5 Control Register"
0x950	SREP_ISF_PSC0	"SREP ISF Performance Statistics Counter 0 Register"
0x954	SREP_ISF_PSC1	"SREP ISF Performance Statistics Counter 1 Register"
0x958	SREP_ISF_PSC2	"SREP ISF Performance Statistics Counter 2 Register"
0x95C	SREP_ISF_PSC3	"SREP ISF Performance Statistics Counter 3 Register"
0x960	SREP_ISF_PSC4	"SREP ISF Performance Statistics Counter 4 Register"
0x964	SREP_ISF_PSC5	"SREP ISF Performance Statistics Counter 5 Register"

26.12.1 SREP ISF Performance Statistics Counter 0 and 1 Control Register

This register controls the PS0 and PS1 performance statistics counter registers. The PS0 and PS1 performance statistics counter registers have the following configuration values selected using this register: direction, type and priority.

The PSy_DIR field determines the performance statistics I2R versus R2I direction application. The PSy_TYPE field assigns the type of statistics collection (for example, all transactions, request, and response) to be accumulated for a specific counter register, starting with "SREP ISF Performance Statistics Counter 0 Register".

The PSy_PRIO[0.3] fields determine the Bridge ISF priority of the transactions that contribute to the performance statistics. "SREP ISF Performance Statistics Counter 0 Register" and "SREP ISF Performance Statistics Counter 1 Register" can be disabled by selecting PSy_PRIO[0.3] to be set to 0. Setting PSy_PRIO[0.3] to all ones collects performance statistics for all transactions.

Register name: SREP_ISF_PSC0n1_CTRL **Reset value: 0x0000_0000**	**Register offset: 940**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	PS0_PRIO 3	PS0_PRIO 2	PS0_PRIO 1	PS0_PRIO 0	Reserved			PS0_DIR
8:15	Reserved					PS0_TYPE		
16:23	PS1_PRIO 3	PS1_PRIO 2	PS1_PRIO 1	PS1_PRIO 0	Reserved			PS1_DIR
24:31	Reserved					PS1_TYPE		

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	PS0_PRIO3	Performance Stats Reg PS0 Priority 3 Selection This value controls whether or not transactions of ISF priority 3 contribute to the performance stats accumulated in the "SREP ISF Performance Statistics Counter 0 Register". These are Bridge ISF transactions that have a translation error (that is, Illegal transaction, transaction OOB). 0 = Do not count priority 3 transactions 1 = Count priority 3 transactions	R/W	0
1	PS0_PRIO2	Performance Stats Reg PS0 Priority 2 Selection This value controls whether or not transactions of ISF priority 2 contribute to the performance stats accumulated in the "SREP ISF Performance Statistics Counter 0 Register". These are ISF Write Block and Write Burst transactions. 0 = Do not count priority 2 transactions. 1 = Count priority 2 transactions.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
2	PS0_PRIO1	Performance Stats Reg PS0 Priority 1 Selection This value controls whether or not transactions of ISF priority 1 contribute to the performance stats accumulated in the "SREP ISF Performance Statistics Counter 0 Register". These are ISF Response transactions. ISF Responses may have DONE or ERROR status. 0 = Do not count priority 1 transactions. 1 = Count priority 1 transactions.	R/W	0
3	PS0_PRIO0	Performance Stats Reg PS0 Priority 0 Selection This value controls whether or not transactions of ISF priority 0 contribute to the performance stats accumulated in the "SREP ISF Performance Statistics Counter 0 Register". These are Bridge ISF Read Word or Read Block transactions. 0 = Do not count priority 0 transactions. 1 = Count priority 0 transactions.	R/W	0
4:6	Reserved	N/A	R	0
7	PS0_DIR	Performance Stats Reg PS0 Direction Selection This value selects the direction (I2R or R2I) for the "SREP ISF Performance Statistics Counter 0 Register". 0 = I2R Stats Counter Register 1 = R2I Stats Counter Register	R/W	0
8:12	Reserved	N/A	R	0
13:15	PS0_TYPE	Performance Stats Reg PS0 Type Selection This value determines the type of performance statistics collected in the "SREP ISF Performance Statistics Counter 0 Register". • 000 = Count all ISF transactions • 001–011 = Reserved • 100 = Count datums of ISF transactions. This counter counts all datums for all transactions (including header). • 101–111 = Reserved	R/W	0
16	PS1_PRIO3	Performance Stats Reg PS1 Priority 3 Selection This value controls whether or not transactions of ISF priority 3 contribute to the performance stats accumulated in the "SREP ISF Performance Statistics Counter 1 Register". These are ISF transactions that have a translation error (that is, Illegal transaction, transaction OOB). • 0 = Do not count priority 3 transactions. • 1 = Count priority 3 transactions.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
17	PS1_PRIO2	Performance Stats Reg PS1 Priority 2 Selection This value controls whether or not transactions of ISF priority 2 contribute to the performance stats accumulated in the "SREP ISF Performance Statistics Counter 1 Register". These are ISF Write Block and Write Burst transactions. • 0 = Do not count priority 2 transactions. • 1 = Count priority 2 transactions.	R/W	0
18	PS1_PRIO1	Performance Stats Reg PS1 Priority 1 Selection This value controls whether or not transactions of ISF priority 1 contribute to the performance stats accumulated in the "SREP ISF Performance Statistics Counter 1 Register". These are ISF Response transactions. ISF Responses may have DONE or ERROR status. • 0 = Do not count priority 1 transactions. • 1 = Count priority 1 transactions.	R/W	0
19	PS1_PRIO0	Performance Stats Reg PS1 Priority 0 Selection This value controls whether or not transactions of ISF priority 0 contribute to the performance stats accumulated in the "SREP ISF Performance Statistics Counter 1 Register". These are Bridge ISF Read Word or Read Block transactions. • 0 = Do not count priority 0 transactions. • 1 = Count priority 0 transactions.	R/W	0
20:22	Reserved	N/A	R	0
23	PS1_DIR	Performance Stats Reg PS1 Direction Selection This value selects the direction (I2R or R2I) for the "SREP ISF Performance Statistics Counter 1 Register". 0 = I2R Stats Counter Register 1 =R2I Stats Counter Register	R/W	0
24:28	Reserved	N/A	R	0
29:31	PS1_TYPE	Performance Stats Reg PS1 Type Selection This value determines the type of performance statistics collected in the "SREP ISF Performance Statistics Counter 1 Register". • 000 = Count all ISF transactions • 001–011 = Reserved • 100 = Count datums of ISF transactions. This counter counts all datums for all transactions (including header). • 101–111 = Reserved	R/W	0

26.12.2 SREP ISF Performance Statistics Counter 2 and 3 Control Register

This register controls the PS2 and PS3 performance statistics counter registers. The PS4 and PS5 performance statistics counter registers have the following configuration values selected using this register: direction, type and priority.

The PSy_DIR field determines the performance statistics I2R versus R2I direction application.

The PSy_TYPE field assigns the type of statistics collection (for example, all transactions, request, and response) to be accumulated for a specific counter register, starting with "SREP ISF Performance Statistics Counter 0 Register".

The PSy_PRIO[0.3] fields determine the Bridge ISF priority of the transactions that contribute to the performance statistics. "SREP ISF Performance Statistics Counter 2 Register" and "SREP ISF Performance Statistics Counter 3 Register" can be disabled by selecting PSy_PRIO[0.3] to be set to 0. Setting PSy_PRIO[0.3] to all ones collects performance statistics for all transactions.

Register name: SREP_ISF_PSC2n3_CTRL **Reset value: 0x0000_0000**	**Register offset: 944**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	PS2_PRIO 3	PS2_PRIO 2	PS2_PRIO 1	PS2_PRIO 0	Reserved			PS2_DIR
8:15	Reserved					PS2_TYPE		
16:23	PS3_PRIO 3	PS3_PRIO 2	PS3_PRIO 1	PS3_PRIO 0	Reserved			PS3_DIR
24:31	Reserved					PS3_TYPE		

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	PS2_PRIO3	Performance Stats Reg PS2 Priority 3 Selection This value controls whether or not transactions of ISF priority 3 contribute to the performance stats accumulated in the "SREP ISF Performance Statistics Counter 2 Register". These are ISF transactions that have a translation error (that is, Illegal transaction, transaction OOB). • 0 = Do not count priority 3 transactions. • 1 = Count priority 3 transactions.	R/W	0
1	PS2_PRIO2	Performance Stats Reg PS2 Priority 2 Selection This value controls whether or not transactions of ISF priority 2 contribute to the performance stats accumulated in the "SREP ISF Performance Statistics Counter 2 Register". These are ISF Write Block and Write Burst transactions. • 0 = Do not count priority 2 transactions. • 1 = Count priority 2 transactions.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
2	PS2_PRIO1	Performance Stats Reg PS2 Priority 1 Selection This value controls whether or not transactions of ISF priority 1 contribute to the performance stats accumulated in the "SREP ISF Performance Statistics Counter 2 Register". These are ISF Response transactions. ISF Responses may have DONE or ERROR status. • 0 = Do not count priority 1 transactions. • 1 = Count priority 1 transactions.	R/W	0
3	PS2_PRIO0	Performance Stats Reg PS2 Priority 0 Selection This value controls whether or not transactions of ISF priority 0 contribute to the performance stats accumulated in the "SREP ISF Performance Statistics Counter 2 Register". These are Bridge ISF Read Word or Read Block transactions. • 0 = Do not count priority 0 transactions. • 1 = Count priority 0 transactions.	R/W	0
4:6	Reserved	N/A	R	0
7	PS2_DIR	Performance Stats Reg PS2 Direction Selection This value selects the direction (I2R or R2I) for the "SREP ISF Performance Statistics Counter 2 Register". • 0 = I2R Stats Counter Register • 1 =R2I Stats Counter Register	R/W	0
8:12	Reserved	N/A	R	0
13:15	PS2_TYPE	Performance Stats Reg PS2 Type Selection This value determines the type of performance statistics collected in the "SREP ISF Performance Statistics Counter 2 Register". • 000 = Count all ISF transactions • 001–011 = Reserved • 100 = Count datums of ISF transactions. This counter counts all datums for all transactions (including header). • 101–111 = Reserved	R/W	0
16	PS3_PRIO3	Performance Stats Reg PS3 Priority 3 Selection This value controls whether or not transactions of ISF priority 3 contribute to the performance stats accumulated in the "SREP ISF Performance Statistics Counter 3 Register". These are ISF transactions that have a translation error (that is, Illegal transaction, transaction OOB). • 0 = Do not count priority 3 transactions. • 1 = Count priority 3 transactions.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
17	PS3_PRIO2	Performance Stats Reg PS3 Priority 2 Selection This value controls whether or not transactions of ISF priority 2 contribute to the performance stats accumulated in the "SREP ISF Performance Statistics Counter 3 Register". These are ISF Write Block and Write Burst transactions. • 0 = Do not count priority 2 transactions. • 1 = Count priority 2 transactions.	R/W	0
18	PS3_PRIO1	Performance Stats Reg PS3 Priority 1 Selection This value controls whether or not transactions of ISF priority 1 contribute to the performance stats accumulated in the "SREP ISF Performance Statistics Counter 3 Register". These are ISF Response transactions. ISF Responses may have DONE or ERROR status. • 0 = Do not count priority 1 transactions. • 1 = Count priority 1 transactions.	R/W	0
19	PS3_PRIO0	Performance Stats Reg PS3 Priority 0 Selection This value controls whether or not transactions of ISF priority 0 contribute to the performance stats accumulated in the "SREP ISF Performance Statistics Counter 3 Register". These are Bridge ISF Read Word or Read Block transactions. • 0 = Do not count priority 0 transactions. • 1 = Count priority 0 transactions.	R/W	0
20:22	Reserved	N/A	R	0
23	PS3_DIR	Performance Stats Reg PS3 Direction Selection This value selects the direction (I2R or R2I) for the "SREP ISF Performance Statistics Counter 3 Register". 0 = I2R Stats Counter Register 1 =R2I Stats Counter Register	R/W	0
24:28	Reserved	N/A	R	0
29:31	PS3_TYPE	Performance Stats Reg PS3 Type Selection This value determines the type of performance statistics collected in the "SREP ISF Performance Statistics Counter 3 Register". • 000 = Count all ISF transactions • 001–011 = Reserved • 100 = Count datums of ISF transactions. This counter counts all datums for all transactions (including header). • 101–111 = Reserved	R/W	0

26.12.3 SREP ISF Performance Statistics Counter 4 and 5 Control Register

This register controls the PS4 and PS5 performance statistics counter registers. The PS4 and PS5 performance statistics counter registers have the following configuration values selected using this register: direction, type and priority.

The PSy_DIR field determines the performance statistics I2R versus R2I direction application.

The PSy_TYPE field assigns the type of statistics collection (for example, all transactions, request, and response) to be accumulated for a specific counter register, starting with "SREP ISF Performance Statistics Counter 0 Register".

The PSy_PRIO[0.3] fields determine the Bridge ISF priority of the transactions that contribute to the performance statistics. "SREP ISF Performance Statistics Counter 3 Register" and "SREP ISF Performance Statistics Counter 5 Register" can be disabled by selecting PSy_PRIO[0.3] to be set to 0. Setting PSy_PRIO[0.3] to all ones collects performance statistics for all transactions.

Register name: SREP_ISF_PSC4n5_CTRL **Reset value: 0x0000_0000**	**Register offset: 948**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	PS4_PRIO 3	PS4_PRIO 2	PS4_PRIO 1	PS4_PRIO 0	Reserved			PS4_DIR
8:15	Reserved					PS4_TYPE		
16:23	PS5_PRIO 3	PS5_PRIO 2	PS5_PRIO 1	PS5_PRIO 0	Reserved			PS5_DIR
24:31	Reserved					PS5_TYPE		

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	PS4_PRIO3	Performance Stats Reg PS4 Priority 3 Selection This value controls whether or not transactions of ISF priority 3 contribute to the performance stats accumulated in the "SREP ISF Performance Statistics Counter 2 Register". These are ISF transactions that have a translation error (that is, Illegal transaction, transaction OOB). • 0 = Do not count priority 3 transactions. • 1 = Count priority 3 transactions.	R/W	0
1	PS4_PRIO2	Performance Stats Reg PS4 Priority 2 Selection This value controls whether or not transactions of ISF priority 2 contribute to the performance stats accumulated in the "SREP ISF Performance Statistics Counter 2 Register". These are ISF Write Block and Write Burst transactions. • 0 = Do not count priority 2 transactions. • 1 = Count priority 2 transactions.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
2	PS4_PRIO1	Performance Stats Reg PS4 Priority 1 Selection This value controls whether or not transactions of ISF priority 1 contribute to the performance stats accumulated in the "SREP ISF Performance Statistics Counter 2 Register". These are ISF Response transactions. ISF Responses may have DONE or ERROR status. • 0 = Do not count priority 1 transactions. • 1 = Count priority 1 transactions.	R/W	0
3	PS4_PRIO0	Performance Stats Reg PS4 Priority 0 Selection This value controls whether or not transactions of ISF priority 0 contribute to the performance stats accumulated in the "SREP ISF Performance Statistics Counter 2 Register". These are Bridge ISF Read Word or Read Block transactions. • 0 = Do not count priority 0 transactions. • 1 = Count priority 0 transactions.	R/W	0
4:6	Reserved	N/A	R	0
7	PS4_DIR	Performance Stats Reg PS4 Direction Selection This value selects the direction (I2R or R2I) for the "SREP ISF Performance Statistics Counter 2 Register". 0 = I2R Stats Counter Register 1 =R2I Stats Counter Register	R/W	0
8:12	Reserved	N/A	R	0
13:15	PS4_TYPE	Performance Stats Reg PS4 Type Selection This value determines the type of performance statistics collected in the "SREP ISF Performance Statistics Counter 2 Register". • 000 = Count all ISF transactions • 001–011 = Reserved • 100 = Count datums of ISF transactions. This counter counts all datums for all transactions (including header). • 101–111 = Reserved	R/W	0
16	PS5_PRIO3	Performance Stats Reg PS5 Priority 3 Selection This value controls whether or not transactions of ISF priority 3 contribute to the performance stats accumulated in the "SREP ISF Performance Statistics Counter 5 Register". These are ISF transactions that have a translation error (that is, Illegal transaction, transaction OOB). • 0 = Do not count priority 3 transactions. • 1 = Count priority 3 transactions.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
17	PS5_PRIO2	Performance Stats Reg PS5 Priority 2 Selection This value controls whether or not transactions of ISF priority 2 contribute to the performance stats accumulated in the "SREP ISF Performance Statistics Counter 5 Register". These are ISF Write Block and Write Burst transactions. • 0 = Do not count priority 2 transactions. • 1 = Count priority 2 transactions.	R/W	0
18	PS5_PRIO1	Performance Stats Reg PS5 Priority 1 Selection This value controls whether or not transactions of ISF priority 1 contribute to the performance stats accumulated in the "SREP ISF Performance Statistics Counter 5 Register". These are ISF Response transactions. ISF Responses may have DONE or ERROR status. • 0 = Do not count priority 1 transactions. • 1 = Count priority 1 transactions.	R/W	0
19	PS5_PRIO0	Performance Stats Reg PS5 Priority 0 Selection This value controls whether or not transactions of ISF priority 0 contribute to the performance stats accumulated in the "SREP ISF Performance Statistics Counter 5 Register". These are Bridge ISF Read Word or Read Block transactions. • 0 = Do not count priority 0 transactions. • 1 = Count priority 0 transactions.	R/W	0
20:22	Reserved	N/A	R	0
23	PS5_DIR	Performance Stats Reg PS5 Direction Selection This value selects the direction (I2R or R2I) for the "SREP ISF Performance Statistics Counter 5 Register". 0 = I2R Stats Counter Register 1 = R2I Stats Counter Register	R/W	0
24:28	Reserved	N/A	R	0
29:31	PS5_TYPE	Performance Stats Reg PS5 Type Selection This value determines the type of performance statistics collected in the "SREP ISF Performance Statistics Counter 5 Register". • 000 = Count all ISF transactions • 001–011 = Reserved • 100 = Count datums of ISF transactions. This counter counts all datums for all transactions (including header). • 101–111 = Reserved	R/W	0

26.12.4 SREP ISF Performance Statistics Counter 0 Register

This register collects performance statistics. These counters provide the means of accumulating statistics for the purposes of throughput and latency performance monitoring.

The counter collects performance statistics information based on the configuration fields in the "SREP ISF Performance Statistics Counter 0 and 1 Control Register".

The PS0_CTR counter value is writable for testing purposes.This counter fills when it reaches its maximum value of 0xFFFFFFFF, and is cleared to 0 when read. The PS0_CTR is enabled when any PS0_PRIO[0..3] value in the "SREP ISF Performance Statistics Counter 0 and 1 Control Register" is configured to a value other than 0.

Register name: SREP_ISF_PSC0 **Reset value: 0x0000_0000**	**Register offset: 950**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	PS0_CTR							
8:15	PS0_CTR							
16:23	PS0_CTR							
24:31	PS0_CTR							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:31	PS0_CTR	This counter collects performance statistics based on the configurations specified using the "SREP ISF Performance Statistics Counter 0 and 1 Control Register". A read clears this register.	R/W	0

26.12.5 SREP ISF Performance Statistics Counter 1 Register

This register collects performance statistics. These counters provide the means of accumulating statistics for the purposes of throughput and latency performance monitoring.

The counter collects performance statistics information based on the configuration fields in the "SREP ISF Performance Statistics Counter 0 and 1 Control Register".

The PS1_CTR counter value is writable for testing purposes.This counter fills when it reaches its maximum value of 0xFFFFFFFF, and is cleared to 0 when read. The PS1_CTR is enabled when any PS1_PRIO[0..3] value in the "SREP ISF Performance Statistics Counter 0 and 1 Control Register" is configured to a value other than 0.

Register name: SREP_ISF_PSC1 **Reset value: 0x0000_0000**	**Register offset: 954**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	PS1_CTR							
8:15	PS1_CTR							
16:23	PS1_CTR							
24:31	PS1_CTR							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:31	PS1_CTR	This counter collects performance statistics based on the configurations specified using the "SREP ISF Performance Statistics Counter 0 and 1 Control Register". A read clears this register.	R/W	0

26.12.6 SREP ISF Performance Statistics Counter 2 Register

This register collects performance statistics. These counters provide the means of accumulating statistics for the purposes of throughput and latency performance monitoring.

The counter collects performance statistics information based on the configuration fields in the "SREP ISF Performance Statistics Counter 2 and 3 Control Register".

The PS2_CTR counter value is writable for testing purposes.This counter fills when it reaches its maximum value of 0xFFFFFFFF, and is cleared to 0 when read. The PS2_CTR is enabled when any PS2_PRIO[0..3] value in the "SREP ISF Performance Statistics Counter 2 and 3 Control Register" is configured to a value other than 0.

Register name: SREP_ISF_PSC2 **Reset value: 0x0000_0000**	**Register offset: 958**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	PS2_CTR							
8:15	PS2_CTR							
16:23	PS2_CTR							
24:31	PS2_CTR							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:31	PS2_CTR	This counter collects performance statistics based on the configurations specified using the "SREP ISF Performance Statistics Counter 2 and 3 Control Register". A read clears this register.	R/W	0

26.12.7 SREP ISF Performance Statistics Counter 3 Register

This register collects performance statistics. These counters provide the means of accumulating statistics for the purposes of throughput and latency performance monitoring.

The counter collects performance statistics information based on the configuration fields in the "SREP ISF Performance Statistics Counter 2 and 3 Control Register".

The PS3_CTR counter value is writable for testing purposes.This counter fills when it reaches its maximum value of 0xFFFFFFFF, and is cleared to 0 when read. The PS3_CTR is enabled when any PS3_PRIO[0..3] value in the "SREP ISF Performance Statistics Counter 2 and 3 Control Register" is configured to a value other than 0.

Register name: SREP_ISF_PSC3 **Reset value: 0x0000_0000**	**Register offset: 95C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	PS3_CTR							
8:15	PS3_CTR							
16:23	PS3_CTR							
24:31	PS3_CTR							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:31	PS3_CTR	This counter collects performance statistics based on the configurations specified using the "SREP ISF Performance Statistics Counter 2 and 3 Control Register". A read clears this register.	R/W	0

26.12.8 SREP ISF Performance Statistics Counter 4 Register

This register collects performance statistics. These counters provide the means of accumulating statistics for the purposes of throughput and latency performance monitoring.

The counter collects performance statistics information based on the configuration fields in the "SREP ISF Performance Statistics Counter 4 and 5 Control Register".

The PS4_CTR counter value is writable for testing purposes.This counter fills when it reaches its maximum value of 0xFFFFFFFF, and is cleared to 0 when read. The PS4_CTR is enabled when any PS4_PRIO[0..3] value in the "SREP ISF Performance Statistics Counter 4 and 5 Control Register" is configured to a value other than 0.

Register name: SREP_ISF_PSC4 **Reset value: 0x0000_0000**	**Register offset: 960**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	PS4_CTR							
8:15	PS4_CTR							
16:23	PS4_CTR							
24:31	PS4_CTR							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:31	PS4_CTR	This counter collects performance statistics based on the configurations specified using the "SREP ISF Performance Statistics Counter 4 and 5 Control Register". A read clears this register.	R/W	0

26.12.9 SREP ISF Performance Statistics Counter 5 Register

This register collects performance statistics. These counters provide the means of accumulating statistics for the purposes of throughput and latency performance monitoring.

The counter collects performance statistics information based on the configuration fields in the "SREP ISF Performance Statistics Counter 4 and 5 Control Register".

The PS5_CTR counter value is writable for testing purposes.This counter fills when it reaches its maximum value of 0xFFFFFFFF, and is cleared to 0 when read. The PS5_CTR is enabled when any PS5_PRIO[0..3] value in the "SREP ISF Performance Statistics Counter 4 and 5 Control Register" is configured to a value other than 0.

Register name: SREP_ISF_PSC5 **Reset value: 0x0000_0000**	**Register offset: 964**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	PS5_CTR							
8:15	PS5_CTR							
16:23	PS5_CTR							
24:31	PS5_CTR							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:31	PS5_CTR	This counter collects performance statistics based on the configurations specified using the "SREP ISF Performance Statistics Counter 4 and 5 Control Register". A read clears this register.	R/W	0

26.13 IDT Specific Registers – Event Notification

The registers in this section are vendor specific. They are not defined in the *RapidIO Interconnect Specification (Revision 1.3)*. They are specific to IDT's design of a RapidIO endpoint.

Reserved bits should not be modified. Reserved register addresses should not be read or written.

Most of these registers are sticky - they are only reset when the SREP receives a power-up reset. The registers are not accessible when the SREP is in reset or powered down.

There are a number of addresses reserved for future use, when the SREP is integrated with a MAC that can detect physical layer errors. These addresses are used for registers that control when events cause port writes or interrupt notifications.

Table 177: Register Map for SREP IDT Specific Registers – Statistics

Offset	Register Name	See
0xA00	SREP_INT_STAT	"SREP Interrupt Status Register"
0xA04	SREP_INT_ENABLE	"SREP Interrupt Event Enable Register"
0xA08	SREP_PW_STAT	"SREP Port-Write Status Register"
0xA0C	SREP_PW_ENABLE	"SREP Port-Write Event Enable Register"
0xA18	SREP_PW_TX_LOG_CTL	"SREP Port-Write Transmit on Logical/Transport Event Control Register"
0xA1C	SREP_PW_TX_LOG_STAT_CTL	"SREP Port-Write Transmit on Logical/Transport Status Event Control Register"
0xA20	SREP_PW_TX_ISF_CTL	"SREP Port-Write Transmit on ISF Event Control Register"
0xA24	SREP_RIO_PW_PARMS	"SREP Port-Write Parameters Register"
0xA44	SREP_INT_LOG_CTL	"SREP Interrupt on Logical/Transport Event Control Register"
0xA48	SREP_INT_LOG_STAT_CTL	"SREP Interrupt on Logical/Transport Status Event Control Register"
0xA4C	SREP_INT_ISF_CTL	"SREP Interrupt on ISF Event Control Register"

26.13.1 SREP Interrupt Status Register

This register indicates which function within the SREP has generated an interrupt. For information on handling interrupts, see "Event Notification and Register Hierarchy".

Reset values stated are for a powerup reset only. Non-powerup resets do not reset this register.

Register name: SREP_INT_STAT **Reset value: 0x0000_0000**	**Register offset: A00**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved				RCS	MECS	Reserved	
08:15	Reserved		DB_RX	ISF_LOG	PW_RX	RIO_LOG	Reserved	
16:23	Reserved							
24:31	Reserved						IMPL_PHY_ERR	STD_PHY_ERR

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:3	Reserved	N/A	R	0
4	RCS	A valid reset request was received by the RapidIO Port	R/W1CS	0
5	MECS	Multicast-Event Control Symbol Received Event status	R/W1CS	0
6:9	Reserved	N/A	R	0
10	DB_RX	At least one Doorbell packet is present in the doorbell request queue.	R	0
11	ISF_LOG	A Bridge ISF logical error has occurred (see "Bridge ISF Error Conditions"	RS	0
12	PW_RX	A port-write packet is received	RS	0
13	RIO_LOG	A logical layer error has occurred (see "Logical I/O Packet Events").	RS	0
14:29	Reserved	N/A	R	0
30	IMPL_PHY_ERR	Implementation Specific Physical Layer Error For information on how to enable this event, see the MAC documentation.	R	0
31	STD_PHY_ERR	Standard Physical Layer Error This bit is never asserted in the Tsi620.	R	0

26.13.2 SREP Interrupt Event Enable Register

This register controls whether SREP events cause an event in the "Block Event Status Register", which can be routed to different event notification methods using the "Event Routing Register 1". For more information on control of SREP events, see "Event Notification and Register Hierarchy".

Register name: SREP_INT_ENABLE Reset value: 0x0000_0000	Register offset: A04

Bits	0	1	2	3	4	5	6	7
00:07	Reserved				RCS_EN	MECS_EN	Reserved	
08:15	Reserved		DB_EN	ISF_LOG_EN	PW_RX_EN	RIO_LOG_EN	Reserved	
16:23	Reserved							
24:31	Reserved						IMPL_PHY_ERR_EN	STD_PHY_ERR_EN

Bits	Name	Description	Type	Reset Value
0:3	Reserved	N/A	R	0
4	RCS_EN	Assert interrupt (RST_IRQ_b) when the MAC signals a valid reset request.	R/W	0
5	MECS_EN	Multicast Event Control Symbol Interrupt Enable Connects to the SREP_MCS_RX bit in the "Block Event Status Register".	R/W	0
6:9	Reserved	N/A	R	0
10	DB_EN	Doorbell Reception Interrupt Enable Connects to the SREP_DB_RX bit in the "Block Event Status Register".	R/W	0
11	ISF_LOG_EN	ISF Logical Error Interrupt Enable Connects to the SREP_ERR bit in the "Block Event Status Register".	R/W	0
12	PW_RX_EN	Port-Write Reception Interrupt Enable Connects to the SREP_ERR bit in the "Block Event Status Register".	R/W	0
13	RIO_LOG_EN	RapidIO Logical Layer Error Interrupt Enable Connects to the SREP_ERR bit in the "Block Event Status Register".	R/W	0
14:29	Reserved	N/A	R	0
30	IMPL_PHY_ERR_EN	Implementation Specific Physical Layer Error Port Write Enable Connects to the SREP_ERR bit in the "Block Event Status Register".	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
31	STD_PHY_ERR_EN	No events are associated with this bit.	R/W	0

26.13.3 SREP Port-Write Status Register

This register indicates which function within the SREP has generated a port-write. Reset values stated are for a power-up reset only. Non-powerup resets do not reset this register.

Register name: SREP_PW_STAT Reset value: 0x0000_0000	Register offset: A08

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved			ISF_LOG	Reserved	RIO_LOG	Reserved	
16:23	Reserved							
24:31	Reserved						IMPL_PHY_ERR	STD_PHY_ERR

Bits	Name	Description	Type	Reset Value
0:10	Reserved	N/A	R	0
11	ISF_LOG	A Bridge ISF logical error has occurred (see “Bridge ISF Error Conditions”).	RS	0
12	Reserved	N/A	R	0
13	RIO_LOG	A logical layer error has occurred (see “Logical I/O Packet Events”.	RS	0
14:29	Reserved	N/A	R	0
30	IMPL_PHY_ERR	Implementation Specific Physical Layer Error For more information on this event, see “Physical Layer Events”.	RS	0
31	STD_PHY_ERR	Standard Physical Layer Error This event will never be asserted in the Tsi620.	RS	0

26.13.4 SREP Port-Write Event Enable Register

This register allows an internal event to cause a port-write to be transmitted.

Note: Addresses A10 and A14 are reserved for registers to enable port-write event notification for individual implementation-specific and standard physical layer events.

Register name: SREP_PW_ENABLE **Reset value: 0x0000_0000**	**Register offset: A0C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved			ISF_LOG_EN	Reserved	RIO_LOG_EN	Reserved	
16:23	Reserved							
24:31	Reserved						IMPL_PHY_ERR_EN	STD_PHY_ERR_EN

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:10	Reserved	N/A	R	0
11	ISF_LOG_EN	Bridge ISF Logical Error Port Write Enable	R/W	0
12	Reserved	N/A	R	0
13	RIO_LOG_EN	RapidIO Logical Layer Error Port Write Enable	R/W	0
14:29	Reserved	N/A	R	0
30	IMPL_PHY_ERR_EN	Implementation Specific Physical Layer Error Port Write Enable	R/W	0
31	STD_PHY_ERR_EN	Standard Physical Layer Error Port Write Enable	R/W	0

26.13.5 SREP Port-Write Transmit on Logical/Transport Event Control Register

This register controls which RapidIO logical/transport layer events use the port-write notification function.

Register name: SREP_PW_TX_LOG_CTL Reset value: 0x0000_0000	Register offset: A18

Bits	0	1	2	3	4	5	6	7
00:07	PW_ERR_RESP_EN	PW_DB_ERR_RESP_EN	Reserved		PW_ILL_TRANS_EN	PW_ILL_TARG_EN	Reserved	PW_RESP_TO_EN
08:15	PW_UNEXP_RESP_EN	PW_UNSUP_TRANS_EN	Reserved					
16:23	Reserved							
24:31	Reserved		PW_R2I_TTL_EN	PW_SPOOF_EN	PW_R2I_PERR_EN	PW_OOB_EN	PW_NO_WR_EN	PW_NO_RD_EN

Bits	Name	Description	Type	Reset Value
0	PW_ERR_RESP_EN	Port-Write on Error Response Enable Enables port-write transmission when a response packet with an ‘Error’ status is received for an outstanding request.	R/WS	0
1	PW_DB_ERR_RESP_EN	Port-Write on Doorbell Error Response Enable Enables port-write transmission when a response packet with an ‘Error’ status is received for an outstanding Doorbell request.	R/WS	0
2:3	Reserved	N/A	R	0
4	PW_ILL_TRANS_EN	Port-Write on Illegal Transaction Enable Enables port-write transmission when a packet was received that has illegal field values for an otherwise supported transaction (see “Logical I/O Packet Events”).	R/WS	0
5	PW_ILL_TARG_EN	Port-Write on Illegal Target Decode Enable Enables port-write transmission when a packet was received for a destination ID that is not supported by this endpoint (see “8/16-bit Destination ID Support”).	R/WS	0
6	Reserved	N/A	R	0
7	PW_RESP_TO_EN	Port-Write on Response Timeout Enable Enables port-write transmission when no response is received for a request (see “Logical I/O Packet Events”).	R/WS	0

(Continued)

Bits	Name	Description	Type	Reset Value
8	PW_UNEXP_RESP_EN	Port-Write on Unexpected Response Enable Enables port-write transmission when a response was received that does not match an outstanding request.	R/WS	0
9	PW_UNSUP_TRANS_EN	Port-Write on Unsupported Transaction Enable Enables port-write transmission when a packet was received that is an unsupported request (see "Logical I/O Packet Events").	R/WS	0
10:25	Reserved	N/A	R	0
26	PW_R2I_TTL_EN	Port-Write on R2I Packet Time-to-Live Expired Event Enable Enables an interrupt when a packet is buffered in the R2I Data Buffers/Header Queue for longer than the R2I TTL period programmed in the "SREP R2I Transaction Time-To-Live Register" (see "Logical I/O Packet Events").	R/WS	0
27	PW_SPOOF_EN	Port-Write on Spoof Response Enable Enables port-write transmission when a response is received from a source that does not match the destination ID of the request (see "Logical I/O Packet Events").	R/WS	0
28	PW_R2I_PERR_EN	Port-Write on R2I Parity Error Enables port-write transmission when an R2I BAR or LUT detects a parity error (see "Logical I/O Packet Events").	R/WS	0
29	PW_OOB_EN	Port-Write on Out Of Bounds Enable Enables port-write transmission when a request was received for an address that does not appear in any of the enabled R2I windows (see "Logical I/O Packet Events").	R/WS	0
30	PW_NO_WR_EN	Port-Write on No Write Permission Enable Enables port-write transmission when a write request was received for an area of memory that does not have write permissions enabled (see "Logical I/O Packet Events").	R/WS	0
31	PW_NO_RD_EN	Port-Write on No Read Permission Enable Enables port-write transmission when a read request was received for an area of memory that does not have read permissions enabled (see "Logical I/O Packet Events").	R/WS	0

26.13.6 SREP Port-Write Transmit on Logical/Transport Status Event Control Register

This register controls which RapidIO logical/transport layer status events, as captured in the "SREP R2I Event Status Register", use the port-write notification function.

Register name: SREP_PW_TX_LOG_STAT_CTL Reset value: 0x0000_0000	Register offset: A1C

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							
24:31	Reserved				PW_LUT_BND_EN	PW_BAD_REG_ACC_EN	Reserved	

Bits	Name	Description	Type	Reset Value
0:27	Reserved	N/A	R	0
28	PW_LUT_BND_EN	Port-Write on LUT Entry Boundary Crossing Enables port-write transmission when a request is received that crosses a LUT Entry boundary (see "Logical I/O Packet Events").	R/WS	0
29	PW_BAD_REG_ACC_EN	Port-Write on Bad Register Access Enables port-write transmission when a Maintenance Read, Maintenance Write or a memory-mapped access to registers is greater than 4 bytes in size. Also occurs when a register write transaction is received from a sourceID that is not allowed to write to registers (see "RapidIO Illegal Register Access" and "Logical I/O Packet Events").	R/WS	0
30:31	Reserved	N/A	R	0

26.13.7 SREP Port-Write Transmit on ISF Event Control Register

This register controls which Bridge ISF logical events use the port-write notification function.

Register name: SREP_PW_TX_ISF_CTL Reset value: 0x0000_0000	Register offset: A20

Bits	0	1	2	3	4	5	6	7
00:07	PW_ERR_RESP_EN	Reserved						PW_RESP_TO_EN
08:15	PW_UNEXP_RESP_EN	PW_UNSUP_TRANS_EN	Reserved					
16:23	PW_BE_DISCONT_EN	Reserved		PW_ECC_ERR_EN	PW_R2R_TTL_EN	PW_REG_TTL_EN	PW_I2R_TTL_EN	PW_TEA_EN
24:31	PW_LUT_BND_EN	PW_I2R_PERR_EN	Reserved		PW_UNSAR_REQ_EN	PW_OOB_EN	PW_NO_WR_EN	PW_NO_RD_EN

Bits	Name	Description	Type	Reset Value
0	PW_ERR_RESP_EN	Port-Write on Error Response Enable Enables port-write transmission when a response packet with an 'Error' status is received for an outstanding request.	R/WS	0
1:6	Reserved	N/A	R	0
7	PW_RESP_TO_EN	Port-Write on Response Timeout Enable Enables port-write transmission when no response is received for a request (see "Bridge ISF Error Conditions").	R/WS	0
8	PW_UNEXP_RESP_EN	Port-Write on Unexpected Response Enable Enables port-write transmission when a response was received that does not match an outstanding request.	R/WS	0
9	PW_UNSUP_TRANS_EN	Port-Write on Unsupported Transaction Enable Enables port-write transmission when a packet was received that is an unsupported transaction (see "Bridge ISF Error Conditions").	R/WS	0
10:15	Reserved	N/A	R	0
16	PW_BE_DISCONT_EN	Port-Write on Byte Enables Discontiguous Enable Enables port-write transmission when a Bridge ISF Write Burst transaction was received that causes a Byte Enables Discontiguous event (see "Bridge ISF Error Conditions").	R/WS	0
17:18	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
19	PW_ECC_ERR_EN	Port-Write on ECC Error Event Enables port-write transmission when an ECC error is detected in the data path (see "Bridge ISF Error Conditions").	R/WS	0
20	PW_R2R_TTL_EN	Port-Write on R2R Time-to-Live Expired Event Enables Port-Write transmission when a RapidIO packet is buffered in the R2R Queue for a period longer than that programmed in the "SREP I2R Transaction Time-To-Live Register" (see "Bridge ISF Error Conditions").	R/WS	0
21	PW_REG_TTL_EN	Port-Write on Register Response Time-to-Live Expired Event Enables Port-Write transmission when a Register Response is buffered in the Register Response Data Buffers/ Queue for a period longer than that programmed in the "SREP I2R Transaction Time-To-Live Register" (see "Bridge ISF Error Conditions").	R/WS	0
22	PW_I2R_TTL_EN	Port-Write on I2R Transaction Time-to-Live Expired Event Enables Port-Write transmission when a Bridge ISF transaction is buffered in the I2R Data Buffers/Header Queue for a period longer than that programmed in the "SREP I2R Transaction Time-To-Live Register" (see "Bridge ISF Error Conditions").	R/WS	0
23	PW_TEA_EN	Port-Write on Bridge ISF TEA Event Enables port-write transmission when a Bridge ISF transaction receives an error acknowledge (see "Bridge ISF Error Conditions").	R/WS	0
24	PW_LUT_BND_EN	Port-Write on Bridge ISF LUT Boundary Crossing Event Enables port-write transmission when a Bridge ISF transaction crosses a LUT entry boundary (see "Bridge ISF Error Conditions").	R/WS	0
25	PW_I2R_PERR_EN	Port-Write on I2R Parity Error Enables port-write transmission when an I2R LUT/BAR parity error is detected (see "Bridge ISF Error Conditions").	R/WS	0
26:27	Reserved	N/A	R	0
28	PW_UNSAR_REQ_EN	Port-Write on Unsegmentable Request Enables port-write transmission when a request that is mapped to a Maintenance read or write is either not 4 bytes, or is not aligned to a 4-byte boundary, or both. Also detected when a Bridge ISF transaction hits in the Doorbell BAR but cannot be translated to a RapidIO Doorbell request (see "Bridge ISF Error Conditions").	R/WS	0
29	PW_OOB_EN	Port-Write on Out Of Bounds Enable Enables port-write transmission when a request was received for an address that does not appear in any of the enabled I2R windows (see "Bridge ISF Error Conditions").	R/WS	0

(Continued)

Bits	Name	Description	Type	Reset Value
30	PW_NO_WR_EN	Port-Write on No Write Permission Enable Enables port-write transmission when a write request was received for an area of memory that does not have write permissions enabled (see "Bridge ISF Error Conditions").	R/WS	0
31	PW_NO_RD_EN	Port-Write on No Read Permission Enable Enables port-write transmission when a read request was received for an area of memory that does not have read permissions enabled (see "Bridge ISF Error Conditions").	R/WS	0

26.13.8 SREP Port-Write Parameters Register

This register defines port-write timeout value, and other parameters for the transmission of port-writes. When a port-write is pending, this timer begins counting. When this timer expires and the port-write has not yet been cleared, another port-write is sent and the timer begins counting again.

Note that when software triggers transmission of a port-write, this timer does not trigger retransmission (see "SREP Port-Write Transmit Trigger Register Transmission").

Register name: SREP_RIO_PW_PARMS **Reset value: 0x00C0_0000**	**Register offset: A24**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	PW_TIMER				Reserved			
08:15	PW_PRIORITY		Reserved					
16:23	Reserved							
24:31	Reserved							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:3	PW_TIMER	Port-Write Timer This field defines the time period to repeat sending an error reporting Port-Write request for software assistance. The timer is stopped by software writing to the error detect registers (see "Event Notification and Register Hierarchy"). The timeout value is computed by = {[167772160 nsec x pw_timer_value (in decimal)] + 2x(CLK nsec)}, where CLK nsec = clock cycle period of the register bus clock. Register bus frequency - 78.125 MHz • 0000: Disabled. Port-Write is sent once only per event. • 0001: 214.75 ms • 0010: 429.5 ms • 0100: 859 ms • 1000: 1.718 s • 1111: 2.05 us (Debug only) Other values are reserved. Note: For register bus frequency of 62.5 MHz, these intervals increase by (78.125)/(62.5).	R/W	0
4:7	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
8:9	PW_PRIORITY	Port-Write packet priority This field sets the priority of a Port-Write packet. The priority can be set from 0 to 3, where 0b00 = priority 0 0b01 = priority 1 0b10 = priority 2 0b11 = priority 3	R/W	0b11
10:31	Reserved	N/A	R	0

26.13.9 SREP Interrupt on Logical/Transport Event Control Register

This register controls which logical/transport layer events an interrupt is asserted for.

Register name: SREP_INT_LOG_CTL Reset value: 0xCDC0_003F	Register offset: A44

Bits	0	1	2	3	4	5	6	7
00:07	INT_ERR_RESP_EN	INT_DB_ERR_RESP_EN	Reserved		INT_ILL_TRANS_EN	INT_ILL_TARG_EN	Reserved	INT_RESP_TO_EN
08:15	INT_UNEXP_RESP_EN	INT_UNSUP_TRANS_EN	Reserved					
16:23	Reserved							
24:31	Reserved		INT_R2I_TTL_EN	INT_SPOOF_EN	INT_R2I_PERR_EN	INT_OOB_EN	INT_NO_WR_EN	INT_NO_RD_EN

Bits	Name	Description	Type	Reset Value
0	INT_ERR_RESP_EN	Interrupt on Error Response Enable Enables an interrupt when a response packet with an ‘Error’ status is received for an outstanding request.	R/WS	1
1	INT_DB_ERR_RESP_EN	Interrupt on Doorbell Error Response Enable Enables an interrupt when a response packet with an ‘Error’ status is received for an outstanding Doorbell request.	R/WS	1
2:3	Reserved	N/A	R	0
4	INT_ILL_TRANS_EN	Interrupt on Illegal Transaction Enable Enables an interrupt when a packet was received that has illegal field values for an otherwise supported transaction (see “Logical I/O Packet Events”).	R/WS	1
5	INT_ILL_TARG_EN	Interrupt on Illegal Target Decode Enable Enables an interrupt when a packet was received for a destination ID that is not supported by this endpoint (see “8/16-bit Destination ID Support”).	R/WS	1
6	Reserved	N/A	R	0
7	INT_RESP_TO_EN	Interrupt on Response Timeout Enable Enables an interrupt when no response is received for a request (see “Logical I/O Packet Events”).	R/WS	1

(Continued)

Bits	Name	Description	Type	Reset Value
8	INT_UNEXP_RESP_EN	Interrupt on Unexpected Response Enable Enables an interrupt a response was received that does not match an outstanding request.	R/WS	1
9	INT_UNSUP_TRANS_EN	Interrupt on Unsupported Transaction Enable Enables an interrupt when a packet was received that is an unsupported transaction (see "Logical I/O Packet Events").	R/WS	1
10:25	Reserved	N/A	R	0
26	INT_R2I_TTL_EN	Interrupt on R2I Packet Time-to-Live Expired Event Enables an interrupt when a packet is buffered in the R2I Data Buffers/Header Queue for longer than the R2I TTL period programmed in the "SREP R2I Transaction Time-To-Live Register" (see "Logical I/O Packet Events").	R/WS	1
27	INT_SPOOF_EN	Interrupt on Spoof Response Event Enables an interrupt when a response is received from a source ID that does not match the destination ID of the request (see "Logical I/O Packet Events").	R/WS	1
28	INT_R2I_PERR_EN	Interrupt on R2I Parity Error Enables an interrupt when an R2I BAR or LUT detects a parity error (see "Logical I/O Packet Events").	R/WS	1
29	INT_OOB_EN	Interrupt on Out Of Bounds Enable Enables an interrupt when a request was received for an address that does not appear in any of the enabled R2I windows (see "Logical I/O Packet Events").	R/WS	1
30	INT_NO_WR_EN	Interrupt on No Write Permission Enable Enables an interrupt when a write request was received for an area of memory that does not have write permissions enabled (see "Logical I/O Packet Events").	R/WS	1
31	INT_NO_RD_EN	Interrupt on No Read Permission Enable Enables an interrupt when a read request was received for an area of memory that does not have read permissions enabled (see "Logical I/O Packet Events").	R/WS	1

26.13.10 SREP Interrupt on Logical/Transport Status Event Control Register

This register controls which RapidIO logical/transport layer status events, as captured in the "SREP R2I Event Status Register", use the interrupt notification function.

Register name: SREP_INT_LOG_STAT_CTL **Reset value: 0x0000_000F**	**Register offset: A48**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							
24:31	Reserved				INT_LUT_BND_EN	INT_BAD_REG_ACC_EN	INT_PKT_CRC_EN	INT_PKT_STOMP_EN

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:27	Reserved	N/A	R	0
28	INT_LUT_BND_EN	Interrupt on LUT Entry Boundary Crossing Enables an interrupt when a request is received that crosses a LUT Entry boundary (see "Logical I/O Packet Events").	R/WS	1
29	INT_BAD_REG_ACC_EN	Interrupt on Bad Register Request Enables an interrupt when a Maintenance Read, Maintenance Write or a memory-mapped access to registers is greater than 4 bytes. Also occurs when a register write transaction is received from a sourceID that is not allowed to write to registers (see "RapidIO Illegal Register Access" and "Logical I/O Packet Events").	R/WS	1
30	INT_PKT_CRC_EN	Interrupt on Packet CRC Event Enables an interrupt when a packet has bad CRC detected (see "Logical I/O Packet Events").	R/WS	1
31	INT_PKT_STOMP_EN	Interrupt on Packet Stomp Event Enables an interrupt when a packet is stomped while being received from the physical layer (see "Logical I/O Packet Events").	R/WS	1

26.13.11 SREP Interrupt on ISF Event Control Register

This register controls which Bridge ISF logical events cause an interrupt.

Register name: SREP_INT_ISF_CTL Reset value: 0x81C0_9FCF	Register offset: A4C

Bits	0	1	2	3	4	5	6	7
00:07	INT_ERR_RESP_EN	Reserved						INT_RESP_TO_EN
08:15	INT_UNEXP_RESP_EN	INT_UNSUP_TRANS_EN	Reserved					
16:23	INT_BE_DISCONT_EN	Reserved		INT_ECC_ERR_EN	INT_R2R_TTL_EN	INT_REG_TTL_EN	INT_I2R_TTL_EN	INT_TEA_EN
24:31	INT_LUT_BND_EN	INT_I2R_PERR_EN	Reserved		INT_UNSAR_REQ_EN	INT_OOB_EN	INT_NO_WR_EN	INT_NO_RD_EN

Bits	Name	Description	Type	Reset Value
0	INT_ERR_RESP_EN	Interrupt on Error Response Enable Enables interrupt transmission when a response packet with an 'Error' status is received for an outstanding request.	R/WS	1
1:6	Reserved	N/A	R	0
7	INT_RESP_TO_EN	Interrupt on Response Timeout Enable Enables interrupt transmission when no response is received for a request (see "Bridge ISF Error Conditions").	R/WS	1
8	INT_UNEXP_RESP_EN	Interrupt on Unexpected Response Enable Enables interrupt transmission when a response was received that does not match an outstanding request.	R/WS	1
9	INT_UNSUP_TRANS_EN	Interrupt on Unsupported Transaction Enable Enables interrupt transmission when a packet was received that is an unsupported transaction (see "Bridge ISF Error Conditions").	R/WS	1
10:15	Reserved	N/A	R	0
16	INT_BE_DISCONT_EN	Interrupt on Byte Enables Discontiguous Enable Enables interrupt transmission when a Bridge ISF Write Burst transaction was received that causes a Byte Enables Discontiguous event (see "Bridge ISF Error Conditions").	R/WS	1
17:18	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
19	INT_ECC_ERR_EN	Interrupt on ECC Error Event Enables Interrupt transmission when an ECC error is detected in the data path (see "Bridge ISF Error Conditions").	R/WS	1
20	INT_R2R_TTL_EN	Interrupt on R2R Queue Time-to-Live Expired Event Enables interrupt transmission when an R2R transaction is buffered in the R2R Queue for a period longer than that programmed in the "SREP I2R Transaction Time-To-Live Register" (see "Bridge ISF Error Conditions").	R/WS	1
21	INT_REG_TTL_EN	Interrupt on Register Response Time-to-Live Expired Event Enables interrupt transmission when a Register Response is buffered in the Register Response Data Buffers/ Queue for a period longer than that programmed in the "SREP I2R Transaction Time-To-Live Register" (see "Bridge ISF Error Conditions").	R/WS	1
22	INT_I2R_TTL_EN	Interrupt on I2R Transaction Time-to-Live Expired Event Enables interrupt transmission when a Bridge ISF transaction is buffered in the I2R Data Buffers/Header Queue for a period longer than that programmed in the "SREP I2R Transaction Time-To-Live Register" (see "Bridge ISF Error Conditions").	R/WS	1
23	INT_TEA_EN	Interrupt on Bridge ISF TEA Event Enables interrupt transmission when a Bridge ISF transaction receives an error acknowledge (see "Bridge ISF Error Conditions").	R/WS	1
24	INT_LUT_BND_EN	Interrupt on ISF LUT Boundary Crossing Event Enables interrupt transmission when a Bridge ISF transaction crosses a LUT entry boundary (see "Bridge ISF Error Conditions").	R/WS	1
25	INT_I2R_PERR_EN	Interrupt on I2R Parity Error Enables interrupt transmission when a BAR/LUT parity error is detected (see "Bridge ISF Error Conditions").	R/WS	1
26:27	Reserved	N/A	R	0
28	INT_UNSAR_REQ_EN	Interrupt on UnSARable Request Enables interrupt transmission when a Bridge ISF request that is mapped to a Maintenance read or write is either not 4 bytes, or is not aligned to a 4-byte boundary, or both. Also detected when a Bridge ISF transaction hits in the Doorbell BAR but cannot be translated to a RapidIO Doorbell request (see "Bridge ISF Error Conditions").	R/WS	1
29	INT_OOB_EN	Interrupt on Out Of Bounds Enable Enables interrupt transmission when a request was received for an address that does not appear in any of the enabled I2R windows (see "Bridge ISF Error Conditions").	R/WS	1

(Continued)

Bits	Name	Description	Type	Reset Value
30	INT_NO_WR_EN	interrupt on No Write Permission Enable Enables interrupt transmission when a write request was received for an area of memory that does not have write permissions enabled (see “Bridge ISF Error Conditions”).	R/WS	1
31	INT_NO_RD_EN	interrupt on No Read Permission Enable Enables interrupt transmission when a read request was received for an area of memory that does not have read permissions enabled (see “Bridge ISF Error Conditions”).	R/WS	1

26.14 IDT Specific Registers – Electrical Layer

The registers in this section are vendor specific. They are not defined in the *RapidIO Interconnect Specification (Revision 1.3)*. They are specific to IDT's design of a RapidIO endpoint.

Reserved bits should not be modified. Reserved register addresses should not be read or written.

26.14.1 SREP Digital Loopback and Clock Selection Register

For more information on the use of this register, see "Link Maintenance Functions".

Register name: SREP_MAC_DLOOP_CLK_SEL Reset value: 0x7FFF_0000	Register offset: B18

Bits	0	1	2	3	4	5	6	7
00:07	SCRATCH							
08:15	SCRATCH							
16:23	Reserved							
24:31	Reserved					PWDN_X4	Reserved	

Bits	Name	Description	Type	Reset Value
0:15	SCRATCH	Scratch bits. These bits control no functionality.	R/W	0x7FFF
16:28	Reserved	N/A	R	0
29	PWDN_X4	Power-down control for the SREP. When this bit is set, this register is still accessible. All other registers in the SREP are no longer accessible, and are reset. For more information on the use of this field, see "Link Maintenance Functions" and "Bridge Shutdown".	R/W	0
30:31	Reserved	N/A	R	0

Table 178: SREP_RIO_I2R_DATA0 when STAT = 1

Bits	Name	Description	Type	Reset Value
0:15	Reserved	N/A	R	Undefined
16:23	ISF_DATA_LOW_ECC	The ECC code for the data captured in SREP_RIO_I2R_DATA0 and 1.	R	Undefined
24:31	ISF_DATA_LOW_BE	Byte enable indicates for the data captured in SREP_RIO_I2R_DATA0 and 1. Byte enable indications are active high.	R	Undefined

Table 179: SREP_RIO_I2R_DATA2 when STAT = 1

Bits	Name	Description	Type	Reset Value
0:7	Reserved	N/A	R	Undefined
8:15	ISF_DATA_UP_BE_ECC	This is the combined ECC code for the byte enables for SREP_RIO_I2R_DATA0-3.	R	Undefined
16:23	ISF_DATA_UP_ECC	The ECC code for the data captured in SREP_RIO_I2R_DATA2 and 3.	R	Undefined
24:31	ISF_DATA_UP_BE	Byte enable indicates for the data captured in SREP_RIO_I2R_DATA2 and 3. The byte enables are active high.	R	Undefined

Table 180: SREP_RIO_R2I_PKT_DATAy{0, 2, 4,...,62} When CAPT_EN = 0 and STAT = 1

Bits	Name	Description	Type	Reset Value
00:01	Reserved	N/A	R	Undefined
2	DECOMP_EOP	Indicates the last datum in the current decomposed response buffer. For example, assume a decomposed transaction requiring 520 bytes. This requires 3 decomposed buffers, with 256 bytes in the first two and 8 bytes in the last. DECOMP_EOP will be set for the last datum of the first two buffers. DECOMP_EOP is not set for the only datum of the last buffer. EOP is set for the only datum of the last buffer.	R	Undefined
3	EOP	End of packet indication 0 = Packet continues 1 = End of Packet	R	Undefined

Table 180: SREP_RIO_R2I_PKT_DATAy{0, 2, 4,...,62} When CAPT_EN = 0 and STAT = 1 (Continued)

Bits	Name	Description	Type	Reset Value
4:11	ECC_CODE	ECC code for the 64 bits of data	R	Undefined
12:31	Reserved	N/A	R	Undefined

Table 181: SREP_RIO_R2I_PKT_DATAy{0, 2, 4,...,62} When CAPT_EN = 1 and STAT = 1

Bits	Name	Description	Type	Reset Value
00:04	Reserved	N/A	R	Undefined
5:7	TTL	Time to live value of the packet received from the switch. This is only valid for SREP_RIO_R2I_PKT_DATA0.	R	Undefined
8	VALID	Indicates that this datum is a valid datum 0 = Datum is not valid 1 = Datum is valid This bit should always be set.	R	Undefined
9:10	DATUM	Indicates the validity of SREP_RIO_R2I_PKT_DATAx and SREP_RIO_R2I_PKT_DATAx+1, encoded as follows: 0b00 = The packet is STOMPed 0b01 = Half the datum is valid. Only SREP_RIO_R2I_PKT_DATAx has valid data. No data is in SREP_RIO_R2I_PKT_DATAx+1. 0b10 = Undefined 0b11 = Full datum is valid. Both SREP_RIO_R2I_PKT_DATAx and SREP_RIO_R2I_PKT_DATAx+1 have good data.	R	Undefined
11	EOP	Indicates that this datum is the last datum in a packet 0 = Packet continues after this datum 1 = This is the last datum in the packet Note: EOP should always be 1 when DATUM is 0b01.	R	Undefined
12:31	Reserved	N/A	R	Undefined

27. Bridge ISF Registers

Topics discussed include the following:

27.1 Overview

The Bridge ISF registers consist of two types:

- Registers associated with the entire Bridge ISF
- Registers associated with a specific Bridge ISF port

The Bridge ISF registers are divided into multiple address spaces to allow for ports to be added or removed without impacting the register map for other ports. The register map for the Bridge ISF (two ports numbered 0–1) is shown in Table 182. The address range for each port contains 32 registers, and is separated by 0x80. Unspecified offsets are considered reserved.

Table 182: Bridge ISF Base/Offset Register Map

Address Start	Address End	Bridge ISF Port Number	Port
0x2_1000	0x2_107F	0	SREP
0x2_1080	0x2_10FF	1	PCI Interface
0x2_1100	0x21F7F	Reserved	N/A
0x2_1F80	0x21FFF	Global	N/A

27.2 Register Map

The following table lists the register map for the Bridge ISF.

Table 183: Bridge ISF Register Map

Offset	Register Name	See
Bridge ISF Port {0..1} Registers		
0x000, 0x080	BISF_P{0..1}_CAP	“Bridge ISF Port Capability Register”
0x004, 0x084	Reserved	
0x008,0x088	BISF_P{0..1}_ECC_CTRL	“Bridge ISF Port ECC Control Register”
0x00C, 0x08C	BISF_P{0..1}_TEA_TIMEOUT	“Bridge ISF Port TEA Timeout Register”
0x010, 0x090	BISF_P{0..1}_ERR_STAT	“Bridge ISF Port Error Status Register”
0x014–07F	Reserved	
Bridge ISF Global Registers		
0xF84	BISF_CTRL	“Bridge ISF Control Register”
0xF90	BISF_TEA_INT_STAT	“Bridge ISF TEA Interrupt Status Register”
0xF94	BISF_TEA_INT_EN	“Bridge ISF TEA Interrupt Enable Register”
0xF98	BISF_TEA_INT_SET	“Bridge ISF TEA Interrupt Set Register”
0xF9C	BISF_ECC_CE_INT_STAT	“Bridge ISF ECC CE Interrupt Status Register”
0xFA0	BISF_ECC_CE_INT_EN	“Bridge ISF ECC CE Interrupt Enable Register”
0xFA4	BISF_ECC_CE_INT_SET	“Bridge ISF ECC CE Interrupt Set Register”
0xFA8	BISF_ECC_UE_INT_STAT	“Bridge ISF ECC UE Interrupt Status Register”
0xFAC	BISF_ECC_UE_INT_EN	“Bridge ISF ECC UE Interrupt Enable Register”
0xFB0	BISF_ECC_UE_INT_SET	“Bridge ISF ECC UE Interrupt Set Register”
0xFB4–FFF	Reserved	

27.3 Register Descriptions

The following sections describe the Bridge ISF registers. These registers are reset by a chip reset.

27.3.1 Bridge ISF Port Capability Register

This register provides information about the Bridge ISF port capabilities.

Register name: BISF_P{0..1}_CAP **Reset value: 0x0000_0002**	**Register offset: 0x000, 0x080**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	Reserved							
23:16	Reserved							
15:08	Reserved							
07:00	CAP							

Bits	**Name**	**Description**	**Type**	**Reset value**
31:8	Reserved	Reserved	R	0
7:0	CAP	Port Capability 0x02 = Port supports ECC (both checking and generation) All other values are reserved.	R	0x02

27.3.2 Bridge ISF Port ECC Control Register

This register controls the ECC capabilities of the port.

Register name: BISF_P{0..1}_ECC_CTRL Reset value: 0x0000_0000	Register offset: 0x008, 0x088

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	Reserved							
07:00	Reserved						ECC_GEN_DIS	ECC_CHK_DIS

Bits	Name	Description	Type	Reset value
31:2	Reserved	Reserved	R	0
1	ECC_GEN_DIS	Disable ECC Generation in the ingress port. This is only valid when ECC_EN is set in the "Bridge ISF Control Register"; otherwise, this bit has no effect. When set, the ECC check words are expected to be provided along with the packet data so that it can be checked by the port's downstream logic. 0 = Enable ECC generation 1 = Disable ECC generation	R/W	0
0	ECC_CHK_DIS	Disable ECC Checking in the egress port. This is only valid when ECC_EN is set in the "Bridge ISF Control Register"; otherwise, this bit has no effect. When set, the ECC check is expected to occur in the port's downstream logic. 0 = Enable ECC checking 1 = Disable ECC checking	R/W	0

27.3.3 Bridge ISF Port TEA Timeout Register

This register controls the Transaction Error Acknowledge (TEA) timeout value of the port.

Register name: BISF_P{0..1}_TEA_TIMEOUT Reset value: 0x0001_0008	Register offset: 0x00C, 0x08C

Bits	7	6	5	4	3	2	1	0
31:24	TOUT[15:8]							
23:16	TOUT[7:0]							
15:08	Reserved							
07:00	Reserved					DB_LIM		

Bits	Name	Description	Type	Reset value
31:16	TOUT	This value is multiplied by 2^16 to determine the number of clock cycles a request waits for an acknowledge before a TEA occurs. A setting of 0x0000 disables the TEA timer. Note: This clock is run from the RapidIO reference clock, and so can be either 125 MHz or 156.25 MHz as a base frequency.	R/W	0x0001
15:4	Reserved	Reserved	R	0
3:0	DB_LIM	Reserved This field must not be changed from its reset value.	R/W	0x8

27.3.4 Bridge ISF Port Error Status Register

This register reports the status of the Bridge ISF error conditions and provides information and control over the error information. Bridge ISF error sources include correctable and uncorrectable ECC errors, as well as TEA errors. Information that is stored in this register includes the type of error and the syndrome from an ECC error.

The information in this register is updated only when the register are not in the locked state. When the device is reset, the register becomes locked. As a result, its contents are undefined on power-up. The register remains locked until the LOCKED bit is cleared. Once unlocked, the information fields are updated with current information. When the interrupt bit for one of the error sources is set in the interrupt status registers (see "Bridge ISF TEA Interrupt Status Register", "Bridge ISF ECC CE Interrupt Status Register" and "Bridge ISF ECC UE Interrupt Status Register"), the register re-enters the locked state. The register's contents are preserved (even during reset) until software unlocks the register by writing a 1 to clear the LOCKED bit.

Register name: BISF_P{0..1}_ERR_STAT **Reset value: Undefined**	**Register offset: 0x010, 0x090**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	ECC_SYND[15:8]							
23:16	ECC_SYND[7:0]							
15:08	Reserved					TEA	ECC_CE	ECC_UE
07:00	Reserved							LOCKED

Bits	**Name**	**Description**	**Type**	**Reset value**
31:16	ECC_SYND	ECC Syndrome This is the syndrome of the most recent ECC check. This field is updated on every clock cycle when in the unlocked state (see "Error Logging"). The syndrome can be decoded to determine which bit was in error.	R	Undefined
15:11	Reserved	Reserved	R	0
10	TEA	TEA Error This bit indicates that a TEA error was detected and caused the error information registers to enter the locked state. This bit is cleared when exiting the locked state. It is set when entering the locked state due to the TEA INT bit for the port being set in the "Bridge ISF TEA Interrupt Status Register".	R	Undefined

(Continued)

Bits	Name	Description	Type	Reset value
9	ECC_CE	ECC Correctable Error This bit indicates that a correctable ECC error was detected and caused the error information registers to enter the locked state. This bit is cleared when exiting the locked state. It is set when entering the locked state due to the ECC_CE INT bit for the port being set in the "Bridge ISF ECC CE Interrupt Status Register".	R	Undefined
8	ECC_UE	ECC Uncorrectable Error This bit indicates that an uncorrectable ECC error was detected and caused the error information registers to enter the locked state. This bit is cleared when exiting the locked state. It is set when entering the locked state due to the ECC_UE INT bit for the port being set in the "Bridge ISF ECC UE Interrupt Status Register".	R	Undefined
7:1	Reserved	Reserved	R	0
0	LOCKED	Error Status is Locked This bit indicates the locked status of the error information registers. 0 = Unlocked 1 = Locked This bit is set by default and must be cleared after a reset to enable the error information registers.	R/W1C	1

For test purposes, the SET bits can also set the interrupts and cause these registers to enter the locked state. For information on these bits, see the following registers:

- "Bridge ISF TEA Interrupt Set Register"
- "Bridge ISF ECC CE Interrupt Set Register"
- "Bridge ISF ECC UE Interrupt Set Register"

27.3.5 Bridge ISF Control Register

This register controls the general capabilities of the Bridge ISF.

Register name: BISF_CTRL Reset value: 0x0000_0000	Register offset: 0xF84

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	Reserved							
07:00	Reserved							ECC_EN

Bits	Name	Description	Type	Reset value
31:1	Reserved	Reserved	R	0
0	ECC_EN	SF ECC Enable 0 = Disable ECC generation and correction/detection 1 = Enable ECC generation and correction/detection When this bit is set, ECC is calculated at the ingress port of the Bridge ISF. The ECC is then checked at the egress port of the Bridge ISF. Single bit errors are corrected and reported in the “Bridge ISF ECC CE Interrupt Status Register”. Multi-bit errors are not correctable and are reported in the “Bridge ISF ECC UE Interrupt Status Register”.	R/W	0

27.3.6 Bridge ISF TEA Interrupt Status Register

This register monitors the illegal register bus access interrupt and TEA interrupt status of the Bridge ISF ports.

Register name: BISF_TEA_INT_STAT Reset value: 0x0000_0000	Register offset: 0xF90

Bits	7	6	5	4	3	2	1	0
31:24	RB_INT	Reserved						
23:16	Reserved							
15:08	Reserved							
07:00	Reserved						INT[1:0]	

Bits	Name	Description	Type	Reset value
31	RB_INT	Illegal Register Bus Access 0 = No illegal register bus access has occurred 1 = Illegal register bus access occurred. When software tries to access a register bus slave that does not exist, this bit is set. If this bit is set, and the interrupt is enabled in "Bridge ISF TEA Interrupt Status Register", the Bridge ISF interrupt signal to the Interrupt Controller is asserted. This interrupt can be forced to set in "Bridge ISF TEA Interrupt Set Register". Writing a 1 to the bit clears it and causes the Bridge ISF interrupt signal to the Interrupt Controller to be de-asserted.	R/W1C	0
30:2	Reserved	Reserved	R	0
1:0	INT	Port N TEA Interrupt There is 1 bit per port. When INT[N] is: 0 = No TEA occurred on port N 1 = TEA occurred on port N If any of these bits are set, and the interrupt is enabled in "Bridge ISF TEA Interrupt Enable Register", the Bridge ISF interrupt signal to the Interrupt Controller is asserted. These interrupts can be forced to set in "Bridge ISF TEA Interrupt Set Register". Writing a 1 to the bit clears it and causes the Bridge ISF interrupt signal to the Interrupt Controller to be de-asserted.	R/W1C	0

27.3.7 Bridge ISF TEA Interrupt Enable Register

This register enables the illegal register bus access and TEA interrupts in the "Bridge ISF TEA Interrupt Status Register".

Register name: BISF_TEA_INT_EN Reset value: 0x0000_0000	Register offset: 0xF94

Bits	7	6	5	4	3	2	1	0
31:24	RB_EN	Reserved						
23:16	Reserved							
15:08	Reserved							
07:00	Reserved						EN[1:0]	

Bits	Name	Description	Type	Reset value
31	RB_EN	Illegal register bus access Interrupt Enable This bit enables the associated status bit in the "Bridge ISF TEA Interrupt Status Register" to generate an interrupt. When RB_EN is: 0 = Disable RB_INT interrupt 1 = Enable RB_INT interrupt	R/W	0
30:2	Reserved	Reserved	R	0
1:0	EN	Port N TEA Interrupt Enable This bit enables the associated status bit in the "Bridge ISF TEA Interrupt Status Register" to generate an interrupt. There is 1 bit per port. When EN[N] is: 0 = Disable interrupt on port N 1 = Enable interrupt on port N	R/W	0

27.3.8 Bridge ISF TEA Interrupt Set Register

This register sets the illegal register bus access and TEA interrupts in the "Bridge ISF TEA Interrupt Status Register".

Register name: BISF_TEA_INT_SET Reset value: 0x0000_0000	Register offset: 0xF98

Bits	7	6	5	4	3	2	1	0
31:24	RB_SET	Reserved						
23:16	Reserved							
15:08	Reserved							
07:00	Reserved						SET[1:0]	

Bits	Name	Description	Type	Reset value
31	RB_SET	Illegal register bus access Interrupt Set Sets the associated status bit in "Bridge ISF TEA Interrupt Status Register" to generate an interrupt. When RB_SET is: 0 = Do not set RB_INT interrupt 1 = Set RB_INT interrupt	R/W1S	0
30:2	Reserved	Reserved	R	0
1:0	SET	Port N Transaction Error Acknowledge Interrupt Set Sets the associated status bit in "Bridge ISF TEA Interrupt Status Register" to generate an interrupt. There is 1 bit per port. When SET[N] is: 0 = Do not set interrupt on port N 1 = Set interrupt on port N	R/W1S	0

27.3.9 Bridge ISF ECC CE Interrupt Status Register

This register monitors the ECC Correctable Error (CE) interrupt status the Bridge ISF ports.

Register name: BISF_ECC_CE_INT_STAT Reset value: 0x0000_0000	Register offset: 0xF9C

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	Reserved							
07:00	Reserved						INT[1:0]	

Bits	Name	Description	Type	Reset value
31:2	Reserved	Reserved	R	0
1:0	INT	Port N ECC CE Interrupt There is 1 bit per port. When INT[N] is: 0 = No ECC CE occurred on port N 1 = ECC CE occurred on port N If any of these bits are set, and the interrupt is enabled in "Bridge ISF ECC CE Interrupt Enable Register", the Bridge ISF interrupt signal to the Interrupt Controller is asserted. These interrupts can be forced to set in the "Bridge ISF ECC CE Interrupt Set Register". Writing a 1 to the bit clears it and causes the Bridge ISF interrupt signal to the Interrupt Controller to be de-asserted.	R/W1C	0

27.3.10 Bridge ISF ECC CE Interrupt Enable Register

This register enables the ECC Correctable Error interrupts in the "Bridge ISF ECC CE Interrupt Status Register".

Register name: BISF_ECC_CE_INT_EN Reset value: 0x0000_0000	Register offset: 0xFA0

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	Reserved							
07:00	Reserved						EN[1:0]	

Bits	Name	Description	Type	Reset value
31:2	Reserved	Reserved	R	0
1:0	EN	Port N ECC CE Interrupt Enable Enables the associated status bit in the "Bridge ISF ECC CE Interrupt Status Register" to generate an interrupt. There is 1 bit per port. When EN[N] is: 0 = Disable interrupt on port N 1 = Enable interrupt on port N	R/W	0

27.3.11 Bridge ISF ECC CE Interrupt Set Register

This register sets the ECC Correctable Error interrupts in the "Bridge ISF ECC CE Interrupt Status Register".

Register name: BISF_ECC_CE_INT_SET Reset value: 0x0000_0000	Register offset: 0xFA4

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	Reserved							
07:00	Reserved						SET[1:0]	

Bits	Name	Description[a]	Type	Reset value
31:2	Reserved	Reserved	R	0
1:0	SET	Port N ECC Correctable Error Interrupt Set Sets the associated status bit in the "Bridge ISF ECC CE Interrupt Status Register" to generate an interrupt. There is 1 bit per port. When SET[N] is: 0 = Do not set interrupt on port N 1 = Set interrupt on port N	R/W1S	0

a. The bits in this register also assert the BISF_ERR global status reporting bit described in "Block Event Status Register". This provides a method to assert these status signals for software testing.

27.3.12 Bridge ISF ECC UE Interrupt Status Register

This register monitors the ECC Uncorrectable Error (UE) interrupt status of the Bridge ISF ports.

Register name: BISF_ECC_UE_INT_STAT Reset value: 0x0000_0000	Register offset: 0xFA8

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	Reserved							
07:00	Reserved						INT[1:0]	

Bits	Name	Description	Type	Reset value
31:2	Reserved	Reserved	R	0
1:0	INT	Port N ECC UE Interrupt There is 1 bit per port. When INT[N] is: 0 = No ECC UE occurred on port N 1 = ECC UE occurred on port N If any of these bits are set, and the interrupt is enabled in the "Bridge ISF ECC UE Interrupt Status Register", the Bridge ISF interrupt signal to the Interrupt Controller is asserted. These interrupts can be forced to set in the "Bridge ISF ECC UE Interrupt Set Register". Writing a 1 to the bit clears it and causes the Bridge ISF interrupt signal to the Interrupt Controller to be de-asserted.	R/W1C	0

27.3.13 Bridge ISF ECC UE Interrupt Enable Register

This register enables the ECC Uncorrectable Error interrupts in the "Bridge ISF ECC UE Interrupt Status Register".

Register name: BISF_ECC_UE_INT_EN Reset value: 0x0000_0000	Register offset: 0xFAC

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	Reserved							
07:00	Reserved						EN[1:0]	

Bits	Name	Description	Type	Reset value
31:2	Reserved	Reserved	R	0
1:0	EN	Port N ECC UE Interrupt Enable Enables the associated status bit in the "Bridge ISF ECC UE Interrupt Status Register" to generate an interrupt. There is 1 bit per port. When EN[N] is: 0 = Disable interrupt on port N 1 = Enable interrupt on port N	R/W	0

27.3.14 Bridge ISF ECC UE Interrupt Set Register

This register sets the ECC Uncorrectable Error interrupts in the "Bridge ISF ECC UE Interrupt Status Register".

Register name: BISF_ECC_UE_INT_SET Reset value: 0x0000_0000	Register offset: 0xFB0

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	Reserved							
07:00	Reserved						SET[1:0]	

Bits	Name	Description	Type	Reset value
31:2	Reserved	Reserved	R	0
1:0	SET	Port N ECC UE Interrupt Set Sets the associated status bit in the "Bridge ISF ECC UE Interrupt Status Register" to generate an interrupt. There is 1 bit per port. When SET[N] is: 0 = Do not set interrupt on port N 1 = Set interrupt on port N	R/W1S	0

28. PCI Registers

Topics discussed include the following:

28.1 Register Map

The following table summarizes the address offsets for Tsi620's PCI registers. For information about Tsi620's device register map, see the Register Map Overview section of this document.

Table 184: PCI Register Map Summary

Offset	Description	See
000-03C	PCI registers	"Configuration Register Map"
040-0C4	PCI Device-specific registers	"Device-specific Register Map"
0C8-0FC	PCI Capability list registers	"Miscellaneous Register Map"
100-17C	PCI Miscellaneous registers	"Miscellaneous Register Map"
200-24C	Bridge ISF registers	"Miscellaneous Register Map"
300-3F8	PCI FAB Lookup Table (32 entries x 64-bits per entry)	"Miscellaneous Register Map"
400-4FC	Reserved	
500-5F8	PCI BAR2 Lookup Table (32 entries x 64-bits per entry)	"Miscellaneous Register Map"
600-6F8	PCI BAR3 Lookup Table (32 entries x 64-bits per entry)	"Miscellaneous Register Map"

The following list describes some of the key characteristics of the PCI registers:

- They are available on the local register bus.
- Access is allowed only as a single 32-bit wide transaction per register.
- They are handled as 32 bits wide with byte enables used on writes, and all of the bytes returned on reads.
- They are reset by a chip reset.
- They are reset by a Bridge ISF reset if the Tsi620 is configured as a PCI host. The registers are unaffected by a Bridge ISF reset if the Tsi620 is configured as a PCI agent (see "Resets").
- Any offsets not specified are considered reserved.
- Reserved bits outside the PCI header space should be written 0, and should be ignored on reads.

28.2 Configuration Register Map

The following table describes Tsi620's PCI configuration register map.

Table 185: PCI Configuration Register Map

Offset	Register Name	See
0x000	PE_ID	"PCI ID Register"
0x004	PE_CSR	"PCI Control and Status Register"
0x008	PE_CLASS	"PCI Class Register"
0x00C	PE_MISC0	"PCI Miscellaneous 0 Register"
0x010	P2O_BAR0	"PCI Base Address Register 0"
0x014	P2O_BAR0_UPPER	"PCI Base Address Register 0 Upper"
0x018	P2O_BAR2	"PCI Base Address Register 2"
0x01C	P2O_BAR2_UPPER	"PCI Base Address Register 2 Upper"
0x020	P2O_BAR3	"PCI Base Address Register 3"
0x024	P2O_BAR3_UPPER	"PCI Base Address Register 3 Upper"
0x028	PCI Unimplemented	
0x02C	PE_SID	"PCI Subsystem ID Register"
0x030	PE_EROM	"PCI Expansion ROM Register"
0x034	PE_CAP	"PCI Capability Pointer Register"
0x038	PCI Unimplemented	
0x03C	PE_MISC2	"PCI Miscellaneous 2 Register"

28.3 Configuration Register Descriptions

The following section describes Tsi620‘s Configuration registers.

28.3.1 PCI ID Register

This register contains device and vendor identifiers. The read-only values in this register can be overridden by writing to the “PCI Vendor ID Override Register”.

Register name: PE_ID **Reset value: 0x0620_10E3**	**Register offset: 0x000**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	DID							
23:16	DID							
15:08	VID							
07:00	VID							

Bits	**Name**	**Description**	**Type**	**Reset value**
31:16	DID	Device ID IDT allocated Device Identifier	R	0x0620
15:0	VID	Vendor ID PCI SIG allocated Vendor Identifier	R	0x10E3

28.3.2 PCI Control and Status Register

This register defines configurable parameters for how devices interact with the PCI bus, and indicates status information for PCI bus events.

Register name: PE_CSR Reset value: 0x02B8_0000	Register offset: 0x004

Bits	7	6	5	4	3	2	1	0
31:24	D_PE	S_SERR	R_MA	R_TA	S_TA	DEVSEL		MDP_D
23:16	TFBBC	Reserved	DEV66	CAP_L	INT_STAT	Reserved		
15:08	Reserved					INT_DIS	MFBBC	SERR_EN
07:00	WAIT	PERESP	VGAPS	MWI_EN	SC	BM	MS	IOS

Bits	Name	Description	Type	Reset value
31	D_PE	Detected Parity Error 0 = No parity error 1 = Parity error This bit is set when the PCI Master Module detects a data parity error, or the PCI Target Module detects a data or address parity error.	R/W1C	0
30	S_SERR	Signaled PCI_SERRn 0 = PCI_SERRn not asserted 1 = PCI_SERRn asserted The PCI Target Module sets this bit when it asserts PCI_SERRn to signal an address parity error. Note: SERR_EN and PERESP must be set before PCI_SERRn can be asserted.	R/W1C	0
29	R_MA	Received Master Abort 0 = Master Abort is not detected 1 = Master Abort is detected This bit is set when a Tsi620-initiated transaction is terminated with a Master Abort.	R/W1C	0
28	R_TA	Received Target Abort 0 = Target Abort is not detected 1 = Target Abort is detected This bit is set when a Tsi620-initiated transaction is terminated with a Target Abort.	R/W1C	0

(Continued)

Bits	Name	Description	Type	Reset value
27	S_TA	Signaled Target Abort 0 = PCI Target Module did not terminate transaction with Target Abort 1 = PCI Target Module terminated transaction with Target Abort	R/W1C	0
26:25	DEVSEL	Device Select Timing 01 = Tsi620 is a medium-speed device.	R	01
24	MDP_D	Master Data Parity Detected 0 = PCI Master Module did not detect or generate a data parity error 1 = PCI Master Module detected or generated a data parity error This bit is set if the PERESP bit is set and one of the following occurs: • The PCI Master Module is the master of a transaction in which it asserts PCI_PERRn • The addressed target asserts PCI_PERRn	R/W1C	0
23	TFBBC	Target Fast Back-to-Back Capable 1 = Fast back-to-back target transactions are accepted	R	1
22	Reserved	Reserved	R	0
21	DEV66	66-MHz Capable Device 1 = 66-MHz PCI capable device	R	1
20	CAP_L	Capabilities List 1 = Capabilities list is supported	R	1
19	INT_STAT	Interrupt Status This read-only bit indicates the state of the interrupt in the Tsi620. Only when the interrupt disable bit in the command register is a 0 and this interrupt Status bit is a 1, is the Tsi620's PCI_INTn signal asserted. Setting the Interrupt Disable bit to a 1 has no effect on the state of this bit.	R	1
18:11	Reserved	Reserved	R	0
10	INT_DIS	Interrupt Disable 0 = Enable PCI_INTXn assertion 1 = Disable PCI_INTXn assertion This bit disables assertion of the PCI_INTXn signals.	R/W	0
09	MFBBC	Master Fast Back-to-Back Enable 0 = Fast back-to-back master transactions are not accepted	R	0

(Continued)

Bits	Name	Description	Type	Reset value
08	SERR_EN	PCI_SERRn Enable 0 = Disable PCI_SERRn driver 1 = Enable PCI_SERRn driver When SERR_EN and PERESP are set the Tsi620 can report address parity errors with PCI_SERRn as PCI target.	R/W	0
07	WAIT	Wait Cycle Control 0 = No address/data stepping	R	0
06	PERESP	Parity Error Response 0 = Disable 1 = Enable Controls the device's response to address and data parity errors. When enabled, PCI_PERRn is asserted and the MDP_D bit is set in response to data parity errors. When this bit and SERR_EN are set, the device reports address parity errors on PCI_SERRn. This bit does not affect the device's parity generation.	R/W	0
05	VGAPS	VGA Palette Snoop Always 0 = Disabled	R/W	0
04	MWI_EN	Memory Write and Invalidate Enable 0 = Disable. The Tsi620 does not generate Memory Write and Invalidate transactions.	R	0
03	SC	Special Cycles 0 = Tsi620 does not respond to Special cycles as a target	R	0
02	BM	Bus Master 0 = Disable 1 = Enable This bit enables the Tsi620 to generate transactions as a PCI master.	R/W	0
01	MS	Memory Space 0 = Disable 1 = Enable This bit enables the Tsi620 to accept Memory transactions as a PCI target.	R/W	0
00	IOS	I/O Space 0 = Disable 1 = Enable This bit enables the Tsi620 to accept I/O transactions as a PCI target	R/W	0

28.3.3 PCI Class Register

This register indicates the PCI classification of the Tsi620. The read-only values in this register can be overidden by writing to the "PCI Class Override Register".

Register name: PE_CLASS Reset value: 0x0680_0000	Register offset: 0x008

Bits	7	6	5	4	3	2	1	0
31:24	BASE							
23:16	SUB							
15:08	PROG							
07:00	RID							

Bits	Name	Description	Type	Reset value
31:24	BASE	Base Class Code Bridge device.	R	0x06
23:16	SUB	Sub Class Code Other bridge device.	R	0x80
15:08	PROG	Programming Interface	R	0x00
07:00	RID	Revision ID Note that this field should have the same value as the RIO Device Information CAR registers in the Switch and the SREP registers.	R	0x00

28.3.4 PCI Miscellaneous 0 Register

This register controls miscellaneous PCI functions, such as the latency timer value and cacheline size.

Register name: PE_MISC0 Reset value: Undefined	Register offset: 0x00C

Bits	7	6	5	4	3	2	1	0
31:24	BISTC	SBIST	Reserved		CCODE			
23:16	MFUNCT	LAYOUT						
15:08	LTIMER							
07:00	CLINE							

Bits	Name	Description	Type	Reset value
31	BISTC	BIST Capable 0 = Tsi620 is not BIST capable	R	0
30	SBIST	Start BIST 0 = Tsi620 is not BIST capable	R	0
29:28	Reserved	Reserved	R	0
27:24	CCODE	Completion Code 0 = Tsi620 is not BIST capable	R	0
23	MFUNCT	Multifunction Device 0 = Tsi620 is not a multifunction device	R	0
22:16	LAYOUT	Configuration Space Layout conforms to PCI device layout	R	0x00
15:08	LTIMER	Latency Timer This field defines the number of PCI bus clocks before the Tsi620 must initiate termination of a transaction as a master. It specifies in units of PCI bus clocks, the value of the latency timer for the Master. The lower 3 bits of the field are ignored and are set to 0 in the timer, giving the timer a resolution of 8 PCI clock cycles. When the Tsi620 is configured as a PCI device, the LTIMER reset value is 64 clocks.	R/W	Undefined

(Continued)

Bits	Name	Description	Type	Reset value
07:00	CLINE	Cacheline Size 04 = 4 x 32-bit word (16 bytes) 08 = 8 x 32-bit word (32 bytes) 10 = 16 x 32-bit word (64 bytes) 20 = 32 x 32-bit word (128 bytes) This field specifies the system cacheline size in units of 32-bit words. It is used by the PCI Master to determine the PCI read transaction (that is, memory read, memory read line, or memory read multiple) it should generate on the PCI bus. CLINE is also used by the PCI Target to decide how much data to read on the destination bus. Note: If CLINE is programmed to a value not specified, the behavior is undefined.	R/W	0

28.3.5 PCI Base Address Register 0

When enabled, this register maps Tsi620's internal registers to memory space so that external PCI devices can access Tsi620's registers through memory cycles. This register specifies the lower 32 bits of the 256-KB aligned Configuration address window. The lower portion of the register space is only 256 KB; therefore, PCI address lines [17:0] select the register.

This register must be configured before Tsi620's registers can be accessed through PCI memory transactions. Writes are enabled to this register only if BAR0_EN is set in the "PCI Miscellaneous Control and Status Register". If BAR0_EN is set, then this register has a reset value of 0x0000_0004. Reads from this register return all 0 unless BAR0_EN is set. Changes made to this register while the BAR is enabled are not cleared by disabling the BAR even though this register is read back as 0x0000_0000 upon disabling the BAR.

Register name: P2O_BAR0 **Reset value: 0x0000_0004**	**Register offset: 0x010**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	BA[31:24]							
23:16	BA[23:16]							
15:08	Reserved							
07:00	Reserved				PRFTCH	TYPE		IO_MODE

Bits	**Name**	**Description**	**Type**	**Reset value**
31:16	BA[31:16]	Base Address These bits are compared with PCI address bits A[31:18] to determine if the transaction falls within the 256-KB register BAR.	R/W	0
15:04	Reserved	Reserved	R	0
03	PRFTCH	Prefetchable 0 = Window is not prefetchable	R	0
02:01	TYPE	Address Type 10 = 64-bit BAR located anywhere in 64-bit address space. The reset value is 0 unless BAR0_EN is set in the "PCI Miscellaneous Control and Status Register".	R	2
00	IO_MODE	PCI Bus Address Space 0 = Memory 1 = I/O (not supported)	R	0

28.3.6 PCI Base Address Register 0 Upper

When enabled, this register maps Tsi620's internal registers to memory space so that external PCI devices can access Tsi620's registers through memory cycles. This register specifies the upper 32 bits of the 256-KB aligned Configuration address window.

This register must be configured before Tsi620's registers can be accessed through PCI memory transactions. Writes are enabled to this register only if BAR0_EN is set in the "PCI Miscellaneous Control and Status Register". Reads from this register return 0x0000_0000 unless BAR0_EN is set. Changes made to this register while the BAR is enabled are not cleared by disabling the BAR even though it reads back as 0x0000_0000 when disabled.

Register name: P2O_BAR0_UPPER **Reset value: 0x0000_0000**	**Register offset: 0x014**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	BA[63:56]							
23:16	BA[55:48]							
15:08	BA[47:40]							
07:00	BA[39:32]							

Bits	**Name**	**Description**	**Type**	**Reset value**
31:00	BA[63:32]	Base Address Upper Bits These bits are compared with PCI address bits A[63:32] to determine if the transaction falls within the 64-bit register BAR.	R/W	0

28.3.7 PCI Base Address Register 2

This register specifies a prefetchable memory address window for PCI to Bridge ISF transactions. The window size is a minimum of 32 KB and maximum of 1 GB. This window contains 32 index pages.

Writes are enabled to this register by setting the BAR2_EN bit in the "PCI Page Size Register". Reads of this register return 0x0000_0000 unless BAR2_EN is set. If BAR2_EN is set, the default value for this register is 0x0000_000C. Changes made to this register while the BAR is enabled are not cleared by disabling the BAR even though it reads back as 0x0000_0000 when disabled.

Register name: P2O_BAR2 **Reset value: 0x0000_000C**	**Register offset: 0x018**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	BA[31:24]							
23:16	BA[23:16]							
15:08	BA[15]	Reserved						
07:00	Reserved				PRFTCH	TYPE		IO_MODE

Bits	**Name**	**Description**	**Type**	**Reset value**
31:15	BA[31:15]	Base Address These bits are compared with PCI address bits A[31:15] to determine if the transaction falls within the P2O_BAR2 address window.	R/W	0
14:04	Reserved	Reserved	R	0
03	PRFTCH	Prefetchable 0 = Window is non-prefetchable 1 = Window is prefetchable Note: The reset value of this bit automatically changes from 0 to 1 when this BAR is enabled.	R	1
02:01	TYPE	Address Type 10 = 64-bit BAR located anywhere in 64-bit address space. All other values are reserved. Note: The reset value of this field automatically changes from 00 to 10 when BAR2_EN is set in the "PCI Page Size Register".	R	0b10
00	IO_MODE	PCI Bus Address Space 0 = Memory 1 = I/O (not supported)	R	0

28.3.8 PCI Base Address Register 2 Upper

This register specifies the upper bits of a prefetchable memory address window for PCI to Bridge ISF transactions.

Writes are enabled to this register only if BAR2_EN is set in the "PCI Page Size Register". Reads from this register return 0x0000_0000 unless P2O_PAGE_SIZES[BAR2_EN] is set. Changes made to this register while the BAR is enabled are not cleared by disabling the BAR even though it reads back as 0x0000_0000 when disabled.

Register name: P2O_BAR2_UPPER **Reset value: 0x0000_0000**	**Register offset: 0x01C**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	BA[63:56]							
23:16	BA[55:48]							
15:08	BA[47:40]							
07:00	BA[39:32]							

Bits	**Name**	**Description**	**Type**	**Reset value**
31:00	BA[63:32]	Base Address These bits define the upper address portion of the 64-bit P2O_BAR2. These bits are compared with PCI address bits A[63:32] to determine if the transaction falls within the P2O_BAR2 address window. If the P2O_BAR2 window is based on a 32-bit address bus these bits remain zero.	R/W	0

28.3.9 PCI Base Address Register 3

This register specifies a prefetchable memory address window for PCI to Bridge ISF transactions. The window size is a minimum of 32 KB and maximum of 1 GB. This window contains 32 index pages.

Writes are enabled to this register by setting BAR3_EN in the "PCI Page Size Register". Reads of this register return 0x0000_0000 unless BAR3_EN is set. If BAR3_EN is set, the default value for this register is 0x0000_000C. Changes made to this register while the BAR is enabled are not cleared by disabling the BAR even though it reads back as 0x0000_0000 when disabled.

Register name: P2O_BAR3 **Reset value: 0x0000_0000**	**Register offset: 0x020**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	BA[31:24]							
23:16	BA[23:16]							
15:08	BA[15]	Reserved						
07:00	Reserved				PRFTCH	TYPE		IO_MODE

Bits	**Name**	**Description**	**Type**	**Reset value**
31:15	BA[31:15]	Base Address These bits are compared with PCI address bits A[31:15] to determine if the transaction falls within the P2O_BAR3 address window.	R/W	0
14:04	Reserved	Reserved	R	0
03	PRFTCH	Prefetchable 0 = Window is non-prefetchable 1 = Window is prefetchable Note: The reset value of this bit automatically changes from 0 to 1 when this BAR is enabled.	R	0
02:01	TYPE	Address Type 10 = 64-bit BAR located anywhere in 64-bit address space. All other values are reserved Note: The reset value of this field automatically changes from 00 to 10 when BAR3_EN is set in the "PCI Page Size Register".	R	0b00
00	IO_MODE	PCI Bus Address Space 0 = Memory 1 = I/O (not supported)	R	0

28.3.10 PCI Base Address Register 3 Upper

This register specifies the upper bits of a prefetchable memory address window for PCI to Bridge ISF transactions.

Writes are enabled to this register only if BAR3_EN is set in the "PCI Page Size Register". Reads from this register return 0x0000_0000 unless BAR3_EN is set. Changes made to this register while the BAR is enabled are not cleared by disabling the BAR even though it reads back as 0x0000_0000 when disabled.

Register name: P2O_BAR3_UPPER **Reset value: 0x0000_0000**	**Register offset: 0x024**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	BA[63:56]							
23:16	BA[55:48]							
15:08	BA[47:40]							
07:00	BA[39:32]							

Bits	**Name**	**Description**	**Type**	**Reset value**
31:00	BA[63:32]	Base Address These bits define the upper address portion of the 64-bit P2O_BAR3. These bits are compared with PCI address bits A[63:32] to determine if the transaction falls within the P2O_BAR3 address window. If the P2O_BAR3 window is based on a 32-bit address bus these bits remain zero.	R/W	0

28.3.11 PCI Subsystem ID Register

This register provides the subsystem vendor and device IDs. The read-only values in this register can be overridden by writing to the "PCI SID Override Register".

Register name: PE_SID Reset value: 0x0000_0000	Register offset: 0x02C

Bits	7	6	5	4	3	2	1	0
31:24	SID							
23:16	SID							
15:08	SVID							
07:00	SVID							

Bits	Name	Description	Type	Reset value
31:16	SID	Subsystem ID Values for subsystem ID are vendor specific	R	0
15:00	SVID	Subsystem Vendor ID Subsystem Vendor IDs are obtained from the PCI SIG and identify the vendor of the add-in board or subsystem.	R	0

28.3.12 PCI Expansion ROM Register

This register defines the base address of an expansion ROM device. When using this register, make sure that the "Expansion ROM Map Register" is programmed correctly. The EROM_MAP register defines the destination port and address offset for transactions that are claimed by the expansion ROM BAR.

Register name: PE_ROM **Reset value: 0x0000_0000**	**Register offset: 0x030**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	BA[6:0]							Reserved
23:16	Reserved							
15:08	Reserved							
07:00	Reserved							EN

Bits	**Name**	**Description**	**Type**	**Reset value**
31:25	BA[6:0]	Expansion ROM Base Address	R/W	0x0
24:01	Reserved	Reserved	R	0
00	EN	Expansion ROM Decode Enable 0 = Disabled 1 = Enabled	R/W	0x0

28.3.13 PCI Capability Pointer Register

This register indicates the offset in PCI configuration space of the first capabilities pointer in the PCI capabilities linked-list.

Register name: PE_CAP Reset value: 0x0000_00C8	Register offset: 0x034

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	Reserved							
07:00	CAP_PTR							

Bits	Name	Description	Type	Reset value
31:08	Reserved	Reserved	R	0
07:00	CAP_PTR	Capabilities Pointer This field indicates the offset in PCI configuration space of the first capabilities pointer in the capabilities linked-list (see "PCI Slot Identification Capabilities Register").	R	0xC8

28.3.14 PCI Miscellaneous 2 Register

This register controls the miscellaneous PCI functions, such as interrupt pin and line information.

Register name: PE_MISC2 Reset value: 0x0000_0000	Register offset: 0x03C

Bits	7	6	5	4	3	2	1	0
31:24	MAX_LAT							
23:16	MIN_GNT							
15:08	INT_PIN							
07:00	INT_LINE							

Bits	Name	Description	Type	Reset value
31:24	MAX_LAT	Maximum Latency No special latency requirements.	R	0
23:16	MIN_GNT	Minimum Grant No special requirements.	R	0
15:08	INT_PIN	Interrupt Pin	R	0x00
07:00	INT_LINE	Interrupt Line The Tsi620 uses this field to define interrupt routing information between initialization code and the device driver.	R/W	0

28.4 Device-specific Register Map

The following table describes the PCI device-specific register map.

Table 186: Device-specific Register Map

Offset	Register Name	See
040	MISC_CSR	“PCI Miscellaneous Control and Status Register”
044	SERR_DIS	“PCI SERRn Disable Register”
048	SERR_STAT	“PCI SERRn Status Register”
04C	P2O_PAGE_SIZES	“PCI Page Size Register”
050	P_VPD_CSR	“PCI VPD Control and Status Register”
054–0C4	Reserved	
0C8	P_SLOT_ID	“PCI Slot Identification Capabilities Register”
0CC	P_PMC	“PCI Power Management Capabilities Register”
0D0	P_PMCS	“PCI Power Management Control and Status Register”
0D4	P_HS_CSR	“PCI Compact PCI Hot Swap Control and Status Register”
0D8	P_VPDC	“PCI Vital Product Data Capability Register”
0DC	P_VPDD	“PCI Vital Product Data Register”
0E0	P_MSIC	“PCI Message Signaled Interrupt Control Capability Register”
0E4	P_MSIA	“PCI Message Signaled Interrupt Address Register”
0E8	P_MSIA_UPPER	“PCI Message Signaled Interrupt Address Upper Register”
0EC	P_MSID	“PCI Message Signaled Interrupt Data Register”
0F4	P_PCI_S	“PCI Bus Number Register”
0F8–FFC	Reserved	

28.5 Device-specific Register Descriptions

This section describes the PCI device-specific registers.

28.5.1 PCI Miscellaneous Control and Status Register

This register controls miscellaneous functions, such as the P2O_BAR0 enable setting as well as status information concerning PCI power-up.

Register name: MISC_CSR **Reset value: Undefined**	**Register offset: 0x040**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	IDT_DEV_ID							
23:16	IDT_VER_ID							
15:08	VPD_EN	PRE_DRIVE	SYS32	Reserved		M66EN	CTL_RSC	D64
07:00	LOCKOUT	SOFT_ RESET	PRIMARY	Reserved		EN_1K	BAR0_EN	Reserved

Bits	**Name**	**Description**	**Type**	**Reset value**
31:24	IDT_DEV_ID	IDT Internal Device ID	R	0x00
23:16	IDT_VER_ID	IDT Internal Version ID	R	0x01
15	VPD_EN	PCI Vital Product Data (VPD) 0 = Disable VPD 1 = Enable VPD This bit enables PCI VPD (see "Vital Product Data"). When enabled, the VPD registers in the PCI configuration space can access PCI Vital Product Data.	R/W	0
14	PRE_DRIVE	PCI (2.3) address pre-drive enable for configuration cycles. 0 = PCI Configuration address driven simultaneously with the assertion of PCI_FRAMEn 1 = PCI Configuration address driven four PCI_CLK clock cycles before the assertion of PCI_FRAMEn	R/W	0

(Continued)

Bits	Name	Description	Type	Reset value
13	SYS32	Selects 32-bit PCI system operation when the Tsi620 is the PCI central resource. 0 = 64-bit PCI operation. Selection between 32-bit and 64-bit operations is determined by the PCI_ACK64. 1 = 32-bit PCI operation. All operations are 32-bit mode. It is a programming error to change this bit to 1 in the Tsi620. Note: The state of this bit must not be changed while the BM bit is set in the "PCI Control and Status Register". Note: This feature is provided as a convenience for 32-bit (only) PCI systems to remove the need for pull-up resistors on these 37 signals. This bit must not be enabled in systems that support 64-bit operation. Note: See also the D64 field of this register. Note: See *PCI Local Bus Specification (Revision 2.3)* and the section entitled "64-bit Bus Extension."	R/W	0
12:11	Reserved	Reserved	R	0
10	M66EN	Latched value of the PCI_M66EN pin.	R	Undefined
09	CTL_RSC	Controlling Resource at Power-up The reset state of this bit is set by the state of PCI_RSTDIR.	R	Undefined
08	D64	PCI Data Bus Width at Power-up 0 = 32-bit data bus	R	0
07	LOCKOUT	PCI Lockout 0 = Not set 1 = Set When set, all Configuration (excluding Type 0 Configuration accesses) and Memory accesses from the PCI Interface are retried. This bit must be cleared before the PCI Interface can claim transactions. This bit can be cleared by an agent on a non-locked bus or by internal register bus.	R/W	0
06	SOFT_RESET	Hold in Reset for Software Initialization 0 = Not reset (normal operation) 1 = Reset If set, the PCI Interface (except its control registers) is placed into a reset state. The following actions are taken: PCI_PME, PCI_INT[A:D]n are tristated; memory and I/O spaces are disabled; the PCI Interface and the logic between the Interface and the Bridge ISF are reset. Note: The PCI_HOLD_BOOT pin determines the reset state of this bit.	R/W	Undefined
05	PRIMARY	Power-up as Primary Interface 0 = Secondary (not supported) 1 = Primary	R	1

(Continued)

Bits	Name	Description	Type	Reset value
04:03	Reserved	Reserved	R	0
02	EN_1K	Enable 1-KB Transfers 0 = 256-byte transfers 1 = 1-KB transfers This bit enables 1-KB transfers from the PCI Interface to the Bridge ISF.	R/W	1
01	BAR0_EN	PCI Registers Base Address Register 0 Enable 0 = Disable 1 = Enable When this bit is cleared, the P2O_BAR0 register is not visible in PCI configuration space and is read zero only. Writes to BAR0 have no effect when this bit is cleared; therefore, the PCI Interface does not request PCI memory space for access to its internal registers.	R/W	1
00	Reserved	Reserved	R	0

Improper configuration of this register (for example, LOCKOUT and SOFT_RESET) can lock or reset the PCI Interface.

28.5.2 PCI SERRn Disable Register

This register controls the assertion of the PCI_SERRn signal due to certain error conditions, when the Tsi620 is PCI bus master. To assert PCI_SERRn in response to an error condition, SERR_EN must also be set in the “PCI Control and Status Register”.

Register name: SERR_DIS **Reset value: 0x0000_0000**	**Register offset: 0x044**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	Reserved							
23:16	Reserved							
15:08	Reserved							
07:00	Reserved	DR_ND	DW_ND	PW_MA	PW_TA	PW_RETRY	Reserved	

Bits	**Name**	**Description**	**Type**	**Reset value**
31:07	Reserved	Reserved	R	0
06	DR_ND	Delayed Read No Data 0 = Assert PCI_SERRn when discard timer expires 1 = Do not assert PCI_SERRn when discard timer expires	R/W	0
05	DW_ND	Delayed Write Non Delivery 0 = Assert PCI_SERRn when discard timer expires 1 = Do not assert PCI_SERRn when discard timer expires	R/W	0
04	PW_MA	Posted Write Master Abort 0 = Assert PCI_SERRn when master abort occurs 1 = Do not assert PCI_SERRn when master abort occurs	R/W	0
03	PW_TA	Posted Write Target Abort 0 = Assert PCI_SERRn when target abort occurs 1 = Do not assert PCI_SERRn when target abort occurs	R/W	0
02	PW_RETRY	Posted Write Maximum Retry 0 = Assert PCI_SERRn when 2^{24} retries occur 1 = Do not assert PCI_SERRn when 2^{24} retries occur	R/W	0
01:00	Reserved	Reserved	R	0

28.5.3 PCI SERRn Status Register

This register provides error status information because of a condition that caused the assertion of PCI_SERRn. Note that the relevant error condition must be enabled (set to 0) in the "PCI SERRn Disable Register". In addition, the conditions that allow PCI_SERRn to be asserted must be enabled in the "PCI Control and Status Register".

Register name: SERR_STAT **Reset value: 0x0000_0000**	**Register offset: 0x048**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	Reserved							
23:16	Reserved							
15:08	Reserved							
07:00	D_TOUT	DR_ND	DW_ND	PW_MA	PW_TA	PW_RETRY	PW_DPE	APE

Bits	**Name**	**Description**[a]	**Type**	**Reset value**
31:08	Reserved	Reserved	R	0
07	D_TOUT	Delayed Cycle Timeout PCI_SERRn was asserted, if enabled, due to a master timeout.	R/W1C	0
06	DR_ND	Delayed Read No Data from Target PCI_SERRn was asserted, if enabled, due to a discard timer.	R/W1C	0
05	DW_ND	Delayed Write Non Delivery PCI_SERRn was asserted, if enabled, due to a discard timer.	R/W1C	0
04	PW_MA	Posted Write Master Abort PCI_SERRn was asserted, if enabled, due to a master abort.	R/W1C	0
03	PW_TA	Posted Write Target Abort PCI_SERRn was asserted, if enabled, due to a target abort.	R/W1C	0
02	PW_RETRY	Posted Write Maximum Retry PCI_SERRn was asserted, if enabled, after 2^{24} retries.	R/W1C	0
01	PW_DPE	Posted Write Data Parity Error PCI_SERRn was asserted, if enabled, due to a data parity error.	R/W1C	0
00	APE	Address Parity Error PCI_SERRn was asserted, if enabled, due to an address parity error.	R/W1C	0

a. The interrupt status bits in this register can be set even if SERR_EN is not enabled or asserted.

28.5.4 PCI Page Size Register

This register sets the base address window sizes and enables their operations.

Register name: P2O_PAGE_SIZES Reset value: 0x0000_5900	Register offset: 0x04C

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	BAR2_SIZE					BAR2_NOTRAN	Reserved	BAR2_EN
07:00	BAR3_SIZE					BAR3_NOTRAN	Reserved	BAR3_EN

Bits	Name	Description	Type	Reset value
31:16	Reserved	Reserved	R	0
15:11	BAR2_SIZE[4:0]	BAR2 Page Size When the P2O_BAR2 address window is enabled these bits define the size of the window. The address window consists of 32 memory pages (see Table 187).	R/W	0x0B
10	BAR2_NOTRAN	BAR2 Address Translation Control 0 = Address translation enabled 1 = Address translation disabled This bit controls whether address translation is enabled or disabled. Unlike other BARs, the PCI BAR can either use the translation address placed in the appropriate lookup tables (P2O_BAR2_LUT[0:31] and "P2O_BAR2_LUT{0..31} Register" and "P2O_BAR2_LUT_UPPER{0..31} Register"), or the PCI bus address can be passed directly to the Bridge ISF. When address translation is disabled, the lookup tables are used only to determine the destination port. The reset value of this register enables address translation using the lookup tables.	R/W	0
09	Reserved	Reserved	R	0
08	BAR2_EN	Fabric BAR Enable 0 = Disable 1 = Enable Note: This register can only be written to when the BAR2_EN bit is set to 1. Writes to this register are disabled when the BAR2_EN bit is set to 0.	R/W	1

(Continued)

Bits	Name	Description	Type	Reset value
07:03	BAR3_SIZE[4:0]	BAR3 Page Size When the P2O_BAR3 address window is enabled these bits define the size of the window. The address window consists of 32 memory pages (see Table 187).	R/W	0
02	BAR3_NOTRAN	BAR3 Address Translation Control 0 = Address translation enabled 1 = Address translation disabled This bit controls whether address translation is enabled or disabled. Unlike other BARs, the PCI BAR can either use the translation address placed in the appropriate lookup tables (P2O_BAR3_LUT[0:31] and P2O_BAR3_LUT_UPPER[0:31], see "P2O_BAR3_LUT{0..31} Register" and "P2O_BAR3_LUT_UPPER{0..31} Register"), or the PCI bus address can be passed directly to the Bridge ISF. When address translation is disabled, the lookup tables are used only to determine the destination port. The reset value of this register enables address translation using the lookup tables.	R/W	0
01	Reserved	Reserved	R	0
00	BAR3_EN	Fabric BAR Enable 0 = Disable 1 = Enable Note: This register can only be written to when the BAR3_EN bit is set to 1. Writes to this register are disabled when the BAR3_EN bit is set to 0.	R/W	0

Table 187: P2O_BAR2_SIZE, P2O_BAR3_SIZE

BAR_SIZE (Bytes)	Window Size (Bytes)	Page Size (Bytes)	Active BAR (Bits)	Lookup Table Index (Bits)	Offset (Bits)
00	32K	1K	AD[63:15]	AD[14:10]	AD[9:0]
01	64K	2K	AD[63:16]	AD[15:11]	AD[10:0]
02	128K	4K	AD[63:17]	AD[16:12]	AD[11:0]
03	256K	8K	AD[63:18]	AD[17:13]	AD[12:0]
04	512K	16K	AD[63:19]	AD[18:14]	AD[13:0]
05	1M	32K	AD[63:20]	AD[19:15]	AD[14:0]
06	2M	64K	AD[63:21]	AD[20:16]	AD[15:0]
07	4M	128K	AD[63:22]	AD[21:17]	AD[16:0]
08	8M	256K	AD[63:23]	AD[22:18]	AD[17:0]
09	16M	512K	AD[63:24]	AD[23:19]	AD[18:0]
0A	32M	1M	AD[63:25]	AD[24:20]	AD[19:0]
0B	64M	2M	AD[63:26]	AD[25:21]	AD[20:0]
0C	128M	4M	AD[63:27]	AD[26:22]	AD[21:0]
0D	256M	8M	AD[63:28]	AD[27:23]	AD[22:0]
0E	512M	16M	AD[63:29]	AD[28:24]	AD[23:0]
0F	1G	32M	AD[63:30]	AD[29:25]	AD[24:0]
10	2G	64M	AD[63:31]	AD[30:26]	AD[25:0]
11	4G	128M	AD[63:32]	AD[31:27]	AD[26:0]
12	8G	256M	AD[63:33]	AD[32:28]	AD[27:0]
13	16G	512M	AD[63:34]	AD[33:29]	AD[28:0]
14	32G	1G	AD[63:35]	AD[34:30]	AD[29:0]
15	64G	2G	AD[63:36]	AD[35:31]	AD[30:0]
16	128G	4G	AD[63:37]	AD[36:32]	AD[31:0]
17	256G	8G	AD[63:38]	AD[37:33]	AD[32:0]
18	512G	16G	AD[63:39]	AD[38:34]	AD[33:0]
19	1T	32G	AD[63:40]	AD[39:35]	AD[34:0]
1A	2T	64G	AD[63:41]	AD[40:36]	AD[35:0]

Table 187: P2O_BAR2_SIZE, P2O_BAR3_SIZE (Continued)

BAR_SIZE (Bytes)	Window Size (Bytes)	Page Size (Bytes)	Active BAR (Bits)	Lookup Table Index (Bits)	Offset (Bits)
1B	4T	128G	AD[63:42]	AD[41:37]	AD[36:0]
1C	8T	256G	AD[63:43]	AD[42:38]	AD[37:0]
1D	16T	512G	AD[63:44]	AD[43:39]	AD[38:0]
1E	32T	1T	AD[63:45]	AD[44:40]	AD[39:0]
1F	64T	2T	AD[63:46]	AD[45:41]	AD[40:0]

28.5.5 PCI VPD Control and Status Register

This register controls how vital product data transactions are handled.

Register name: P_VPD_CSR Reset value: 0x0000_0000	Register offset: 0x050

Bits	7	6	5	4	3	2	1	0
31:24	VPD_PORT_NUM				Reserved			
23:16	VPD_OFFSET							
15:08	VPD_OFFSET							
07:00	VPD_OFFSET				Reserved			

Bits	Name	Description	Type	Reset value
31:28	VPD_PORT_NUM	VPD Port Number This field specifies the Bridge ISF port to which VPD reads and writes are directed (see "PCI Vital Product Data Capability Register"). The only valid value for this field is 0. Note: Care must be taken when setting the VPD_PORT_NUM so that writes performed by the PCI Interface target interfaces that can store the data. This requires the RapidIO Interface to be initialized to allow access to a RapidIO memory-mapped location.	R/W	0
27:24	Reserved	Reserved	R	0
23:04	VPD_OFFSET	VPD Offset This field, when concatenated with P_VPDC[VPDA], provides bits 27 to 8 of the 64-bit Bridge ISF address to which VPD reads and writes are directed (see "PCI Vital Product Data Capability Register").	R/W	0
03:00	Reserved	Reserved	R	0

28.5.6 PCI Slot Identification Capabilities Register

This register is not supported by the Tsi620.

The read-only values in this register can be overridden by:

- Using I2C boot (see "I2C Boot")
- Software writing to the "PCI Slot ID Override Register".

Register name: P_SLOT_ID Reset value: 0x0000_CC04	Register offset: 0x0C8

Bits	7	6	5	4	3	2	1	0
31:24	CHASSIS							
23:16	Reserved		FIC	EXP_SLOT				
15:08	NXT_PTR							
07:00	CAP_ID							

Bits	Name	Description	Type	Reset value
31:24	CHASSIS	Chassis Number This field indicates the physical chassis number for the slots on the PCI Interface.	R/W	0
23:22	Reserved	Reserved	R	0
21	FIC	First In Chassis 0 = Tsi620 is not a parent bridge 1 = Tsi620 is a parent bridge	R	0
20:16	EXP_SLOT	Expansion Slot The number of PCI expansion slots on the PCI Interface.	R	0
15:08	NXT_PTR	Next Pointer to PMC block	R	0xCC
07:00	CAP_ID	Capability ID	R	0x04

28.5.7 PCI Power Management Capabilities Register

This register specifies Tsi620's PCI power management capabilities.

Register name: P_PMC Reset value: 0x0002_D401	Register offset: 0x0CC

Bits	7	6	5	4	3	2	1	0
31:24	PME_SUP					D2_SP	D1_SP	Reserved
23:16	Reserved		DSI	Reserved	PME_CK	PM_VER		
15:08	NXT_PTR							
07:00	CAP_ID							

Bits	Name	Description	Type	Reset value
31:27	PME_SUP	PME Support 0X001b = PCI_PMEn can be asserted from D0 0100Xb = PCI_PMEn can be asserted from $D3_{hot}$ This field indicates the power states in which PCI_PMEn is asserted. Bits 4, 2 and 1 are set to 0 since PCI_PMEn is never asserted when in $D3_{cold}$ D2 or D1. Note: This field has no effect on non-PCI Tsi620 interfaces	R	0
26	D2_SP	D2 Support 0 = Tsi620 does not support D2 power state	R	0
25	D1_SP	D1 Support 0 = Tsi620 does not support D1 power state	R	0
24:22	Reserved	Reserved	R	0
21	DSI	Device-Specific Initialization	R	0
20	Reserved	Reserved	R	0
19	PME_CK	PME Clock 0 = a PCI clock is not required for PCI_PMEn operation 1 = a PCI clock is required for PCI_PMEn operation	R	0
18:16	PM_VER	Power Management Version This field indicates that the Tsi620 complies with the *PCI Bus Power Management Interface Specification (Revision 1.1).*	R	010
15:08	NXT_PTR	Next Pointer to Hot Swap Block	R	0xD4
07:00	CAP_ID	Capability ID	R	0x01

28.5.8 PCI Power Management Control and Status Register

This register specifies Tsi620's power management settings.

Register name: P_PMCS Reset value: 0x0000_0000	Register offset: 0x0D0

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	BPCC_EN	B2_B3	Reserved					
15:08	PME_ST	Reserved						PME_EN
07:00	Reserved						PWR_ST	

Bits	Name	Description	Type	Reset value
31:24	Reserved	Reserved	R	0
23	BPCC_EN	Bus Power/Clock Control Enable 0 = Disabled. The clock cannot be controlled by the PCI Interface. In addition, the power/clock control capabilities are disabled.	R	0
22	B2_B3	B2_B3 Support for D3hot This bit is forced to 0 and has no meaning because the Bus Power/Clock Control capabilities are disabled.	R	0
21:16	Reserved	Reserved	R	0
15	PME_ST	PME Status This bit is set when PCI_PMEn would normally be asserted, regardless of the PME_EN bit status.	R	0
14:09	Reserved	Reserved	R	0
08	PME_EN	PME Enable This bit controls the assertion of PCI_PMEn. 0 = Disabled -- PCI_PMEn not asserted 1 = Enabled -- PCI_PMEn can be asserted Note: When enabled, and the interrupt output for PCI INTA from the Tsi620 Interrupt Controller is asserted, PCI_PMEn is driven low; otherwise, PCI_PMEn is tristated.	R/W	0
07:02	Reserved	Reserved	R	0

(Continued)

Bits	Name	Description	Type	Reset value
01:00	PWR_ST	Power State This field determines the current power state and sets the PCI Interface into a new power state. If a non-implemented power state is written to this register, the PCI Interface completes the write, but ignores the data. D0 and $D3_{hot}$ are the only states implemented. 00: D0 11: $D3_{hot}$	R/W	0

28.5.9 PCI Compact PCI Hot Swap Control and Status Register

This register controls the hot swap settings of the PCI Interface. Hot swap is not supported by the Tsi620 PCI interface.

Register name: P_HS_CSR Reset value: 0x0000_E006	Register offset: 0x0D4

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	INS	EXT	Reserved		LOO	Reserved	EIM	Reserved
15:08	NXT_PTR							
07:00	CAP_ID							

Bits	Name	Description	Type	Reset value
31:24	Reserved	Reserved	R	0
23	INS	PCI_ENUMn Status — Insertion 0 = PCI_ENUMn negated 1 = PCI_ENUMn asserted	R/W1C	0
22	EXT	PCI_ENUMn Status — Extraction 0 = PCI_ENUMn negated 1 = PCI_ENUMn asserted	R/W1C	0
21:20	Reserved	Reserved	R	0
19	LOO	LED ON/OFF 0 = LED off 1 = LED on	R/W	0
18	Reserved	Reserved	R	0
17	EIM	PCI_ENUMn Signal Mask 0 = Enable signal 1 = Mask signal	R/W	0
16	Reserved	Reserved	R	0
15:08	NXT_PTR	Next Pointer to Vital Product Data If MISC_CSR[VPD_EN] is set, then this field reads back 0xD8 (see "PCI Miscellaneous Control and Status Register"). If MISC_CSR[VPD_EN] is cleared, then this field reads back 0xE0.	R	0xE0
07:00	CAP_ID	Capability ID	R	0x06

28.5.10 PCI Vital Product Data Capability Register

This PCI register specifies Tsi620's Vital Product Data capabilities. Writes to this register are only enabled if VPD_EN is set in the "PCI Miscellaneous Control and Status Register".

This register can only be written to from the PCI bus. For more information on use of this register, see "Vital Product Data".

Register name: P_VPDC **Reset value: 0x0000_E003**	**Register offset: 0x0D8**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	F	Reserved						
23:16	VPDA							
15:08	NXT_PTR							
07:00	CAP_ID							

Bits	**Name**	**Description**	**Type**	**Reset value**
31	F	Data Transfer Complete Flag This bit indicates when the transfer between the VPD Data register and the memory device is complete. Software clears this bit to initiate a read; the Tsi620 sets the bit when read data is available in the VPD Data register. Software sets the bit to initiate a write; the Tsi620 clears this bit to indicate when the data is transferred.	R/W	0
30:24	Reserved	Reserved	R	0
23:16	VPDA	Vital Product Data Address The 8-bit address specifies the VPD address offset for the VPD Read or VPD Write to the EEPROM. Note that the lower two bits of this field are ignored - a 4-byte aligned address is always generated. Note: The Tsi620 does not provide any write protection of VPD fields.	R/W	0x00
15:08	NXT_PTR	Next Pointer to Message Signaled Interrupt	R	0xE0
07:00	CAP_ID	Capability ID	R	0x03

28.5.11 PCI Vital Product Data Register

This register contains VPD read and write data. It is enabled if MISC_CSR[VPD_EN] is set to 1; otherwise, this register reads all zeros (see "PCI Miscellaneous Control and Status Register"). This register can only be written to from the PCI bus.

Register name: P_VPDD **Reset value: 0x0000_0000**	**Register offset: 0x0DC**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	VPD_DATA							
23:16	VPD_DATA							
15:08	VPD_DATA							
07:00	VPD_DATA							

Bits	**Name**	**Description**	**Type**	**Reset value**
31:00	VPD_DATA	VPD Data	R/W	0

28.5.12 PCI Message Signaled Interrupt Control Capability Register

This register specifies Tsi620's Message Signaled Interrupt (MSI) capabilities.

Register name: P_MSIC Reset value: 0x0084_0005	Register offset: 0x0E0

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	CAP64	MM_EN			MM_CAP			MSI_EN
15:08	NXT_PTR							
07:00	CAP_ID							

Bits	Name	Description	Type	Reset value
31:24	Reserved	Reserved	R	0
23	CAP64	64-bit Address Capable The Tsi620 supports 64-bit message addresses.	R	1
22:20	MM_EN	Multiple Message Enable 000 = 1 message 001 = 2 010 = 4 011 = 8 100 = 16 101 = 32 11x = Reserved This field defines the number of messages that can be generated.	R/W	000
19:17	MM_CAP	Multiple Message Capable The Tsi620 can generate up to four messages at the same time.	R	010
16	MSI_EN	Message Signaled Interrupt Enable 0 = Disable MSI 1 = Enable MSI	R/W	0
15:08	NXT_PTR	Next Pointer The PCI MSI capabilities registers are the last registers.	R	0x00
07:00	CAP_ID	Capability ID	R	0x05

28.5.13 PCI Message Signaled Interrupt Address Register

This register specifies the address for MSI memory write transactions.

Register name: P_MSIA Reset value: 0x0000_0000	Register offset: 0x0E4

Bits	7	6	5	4	3	2	1	0
31:24	MSI_ADDR							
23:16	MSI_ADDR							
15:08	MSI_ADDR							
07:00	MSI_ADDR						Reserved	

Bits	Name	Description	Type	Reset value
31:02	MSI_ADDR[31:2]	Message Signaled Interrupt Address This field specifies the DWORD aligned address for MSI memory write transactions.	R/W	0
01:00	Reserved	Reserved	R	0

28.5.14 PCI Message Signaled Interrupt Address Upper Register

This register specifies the upper address for MSI memory write transactions.

Register name: P_MSIA_UPPER Reset value: 0x0000_0000	Register offset: 0x0E8

Bits	7	6	5	4	3	2	1	0
31:24	MSI_ADDR							
23:16	MSI_ADDR							
15:08	MSI_ADDR							
07:00	MSI_ADDR							

Bits	Name	Description	Type	Reset value
31:00	MSI_ADDR[31:0]	Message Signaled Interrupt Address Upper This field specifies the upper address for MSI memory write transactions. If the value is zero, a single address cycle is generated.	R/W	0

28.5.15 PCI Message Signaled Interrupt Data Register

This register specifies the data for MSI write transactions.

Register name: P_MSID Reset value: 0x0000_0000	Register offset: 0x0EC

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	MSI_DATA							
07:00	MSI_DATA							

Bits	Name	Description	Type	Reset value
31:16	Reserved	Reserved	R	0
15:00	MSI_DATA	Message Signaled Interrupt Data This field specifies the data for MSI write transactions. Note: When two messages are enabled through the P_MSIC[MM_EN] bits, the message number is encoded in bit 0 of the MSI data word and the value in MSI_DATA[0] is ignored (see "PCI Message Signaled Interrupt Control Capability Register"). When three or four messages are enabled the message number is encoded in bits 0 and 1 of the data word and the value in MSI_DATA[1:0] is ignored.	R/W	0

28.5.16 PCI Bus Number Register

This register specifies the PCI bus number information of the Tsi620. It cannot be directly written with configuration writes from the PCI bus; that is, configuration writes to this offset do not modify the contents of the register. However, BUS_NUM [7:0] and DEV_NUM [5:0] can be written with PCI BAR0 accesses or register bus accesses to the register.

Register name: PE_PCI_S **Reset value: 0x0193_FFF8**	**Register offset: 0x0F4**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	Reserved							
23:16	Reserved							
15:08	BUS_NUM							
07:00	DEV_NUM					Reserved		

Bits	**Name**	**Description**	**Type**	**Reset value**
31:16	Reserved	Reserved	R	0x0193
15:08	BUS_NUM	Bus Number This field is updated when a configuration type 0 access is received. The master uses the bus number in the attribute phase of a PCI transaction. Note: This field is R/W from other Tsi620 interfaces.	R	0xFF
07:03	DEV_NUM	Device Number This field is updated when a configuration type 0 access is received. The master uses the device number in the attribute phase of a PCI transaction. Note: This field is R/W from other Tsi620 interfaces.	R	0x1F
02:00	Reserved	Reserved	R	0

28.6 Miscellaneous Register Map

The following table describes Tsi620's PCI Miscellaneous register map.

Table 188: Miscellaneous Register Map

Offset	Register Name	See
Miscellaneous Registers		
0x100–108	Reserved	
0x10C	ARB_CTRL	"PCI Arbiter Control and Status Register"
0x110	P_CSR_SHADOW	"PCI Control and Status Shadow Register"
0x11C	SERR_STAT_SHADOW	"PCI SERRn Status Shadow Register"
0x128	P_LAST_OP_SHADOW	"PCI Last Transaction Shadow Register"
0x12C	P_LAST_ADDR_UPPER_SHADOW	"PCI Last Upper Address Shadow Register"
0x130	P_LAST_ADDR_LOWER_SHADOW	"PCI Last Lower Address Shadow Register"
0x134	P_LAST_ATTR_SHADOW	"PCI Last Attribute Shadow Register"
0x138–17C	Reserved	
0x180	IRP_CFG_CTL	"PCI Interrupt Control Register"
0x184	IRP_STAT	"PCI Interrupt Status Register"
0x188	IRP_ENABLE	"PCI Interrupt Enable Register"
0x18C	IRP_INTAD	"PCI Interrupt CSR"
0x190–198	Reserved	
0x19C	P_VID_OVERRIDE	"PCI Vendor ID Override Register"
0x1A0	P_CLASS_OVERRIDE	"PCI Class Override Register"
0x1A4	PE_SID_OVERRIDE	"PCI SID Override Register"
0x1A8	P_SLOT_ID_OVERRIDE	"PCI Slot ID Override Register"
0x1AC–1FC	Reserved	
Bridge ISF Registers		
0x200	PFAB_CSR	"PFAB Control and Status Register"
0x204	PFAB_BAR0	"PFAB_BAR0 Register"
0x208	PFAB_BAR0_UPPER	"PFAB_BAR0_UPPER Register"
0x20C	PFAB_IO	"PFAB_IO Base Address Register"

Table 188: Miscellaneous Register Map (Continued)

Offset	Register Name	See
0x210	PFAB_IO_UPPER	“PFAB_IO_UPPER Register”
0x214	PFAB_MEM32	“PFAB_MEM32 Base Address Register”
0x218	PFAB_MEM32_REMAP	“PFAB_MEM32_REMAP Register”
0x21C	PFAB_MEM32_MASK	“PFAB_MEM32_MASK Register”
0x220	PFAB_PFM3	“PFAB_PFM3 Base Address Register”
0x224	PFAB_PFM3_REMAP_UPPER	“PFAB_PFM3_REMAP_UPPER Register”
0x228	PFAB_PFM3_REMAP_LOWER	“PFAB_PFM3_REMAP_LOWER Register”
0x22C	PFAB_PFM3_MASK	“PFAB_PFM3_MASK Register”
0x230	PFAB_PFM4	“PFAB_PFM4 Base Address Register”
0x234	PFAB_PFM4_REMAP_UPPER	“PFAB_PFM4_REMAP_UPPER Register”
0x238	PFAB_PFM4_REMAP_LOWER	“PFAB_PFM4_REMAP_LOWER Register”
0x23C	PFAB_PFM4_MASK	“PFAB_PFM4_MASK Register”
0x240–2F8	Reserved	
Expansion ROM		
0x2FC	EROM MAP	“Expansion ROM Map Register”
Lookup Tables		
0x300–4FC	Reserved	
0x500–5F8	P2O_BAR2_LUT	“P2O_BAR2_LUT{0..31} Register”
0x5FC	Reserved	
0x600–6F8	P2O_BAR3_LUT	“P2O_BAR3_LUT{0..31} Register”
0x6FC-FF8	Reserved	
0x1000-0xFFFC	Reserved, used by other blocks in Tsi620	

28.7 Miscellaneous Register Descriptions

The following section describes the PCI Miscellaneous registers.

28.7.1 PCI Arbiter Control and Status Register

This register controls the PCI arbiter settings of the PCI Interface.

Register name: ARB_CTRL **Reset value: 0x0000_00C0**	**Register offset: 0x10C**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	Reserved							
23:16	Reserved							
15:08	M7_PRI	M6_PRI	M5_PRI	M4_PRI	M3_PRI	M2_PRI	M1_PRI	MY_PRI
07:00	TO_CNT		Reserved		PARK	BM_PARK		

Bits	**Name**	**Description**	**Type**	**Reset value**
31:16	Reserved	Reserved	R	0
15	M7_PRI	Reserved	R	0
14	M6_PRI	Reserved	R	0
13	M5_PRI	Reserved	R	0
12	M4_PRI	Arbitration Priority for Bus Master M4. 0 = Low priority 1 = High priority This bit defines the arbitration priority for external bus master, M4, on the PCI bus.	R/W	0
11	M3_PRI	Arbitration Priority for Bus Master M3 (see M4_PRI).	R/W	0
10	M2_PRI	Arbitration Priority for Bus Master M2 (see M4_PRI).	R/W	0
09	M1_PRI	Arbitration Priority for Bus Master M1 (see M4_PRI).	R/W	0
08	MY_PRI	Arbitration Priority for the PCI Interface on the PCI Bus 0 = Low priority 1 = High priority	R/W	0

(Continued)

Bits	Name	Description	Type	Reset value
07:06	TO_CNT	Timeout Counter The arbiter has an idle clock counter that starts counting after a PCI_GNTn is issued, and resets when a PCI_FRAMEn is asserted. This counter checks whether or not a Master starts a transaction *x* number of clocks after receiving a PCI_GNTn. The *PCI Local Bus Specification (Revision 2.3)* says that an arbiter can handle a master that does not start a transaction in 16 clocks as broken. A broken master's PCI_REQn is then ignored. This field enables the user to program the number of clocks the timeout event is triggered and when a master is labeled as broken. **TO_CNT** **PCI Clocks** 00 16 01 32 10 64 11 OFF	R/W	11
05:04	Reserved	Reserved	R	0
3	PARK	Arbiter Park Type 0 = Last master 1 = Master specified in BM_PARK When this bit is set, the arbiter parks the bus on the master defined in the BM_PARK field. When this bit is cleared, the arbiter parks the bus on the last master to be granted the bus.	R/W	0
2:0	BM_PARK	Parked Bus Master This field identifies the bus master to be parked. The PCI bus is parked at this master when PARK = 1. It is a programming error to set BM_PARK to 0b101, 0b110, or 0b111. The operation of the PCI Interface is undefined. **BM_PARK [2:0]** **Parked PCI Master** **External Pins** 000 PCI Inter. None 001 M1 P1_REQ_[1]/P1_GNT_[1] 010 M2 P1_REQ_[2]/P1_GNT_[2] 011 M3 P1_REQ_[3]/P1_GNT_[3] 100 M4 P1_REQ_[4]/P1_GNT_[4]	R/W	000

28.7.2 PCI Control and Status Shadow Register

This register is a shadow of "PCI Control and Status Register". The contents of this register are not reset by a chip reset, and are undefined after a power-up event. This register maintains its last state across a chip reset provided power is not lost. This state is maintained until a write of any value is performed to this register. Writing to this register enables all the shadow registers for update. Once enabled, the shadow registers continually update to mirror the contents in the shadowed registers.

Register name: P_CSR_SHADOW **Reset value: Undefined**	**Register offset: 0x110**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	D_PE	S_SERR	R_MA	R_TA	S_TA	Reserved		MDP_D
23:16	Reserved							
15:08	Reserved							SERR_EN
07:00	Reserved	PERESP	Reserved			BM	MS	IOS

Bits	**Name**	**Description**	**Type**	**Reset value**
31	D_PE	Detected Parity Error 0 = No parity error 1 = Parity error This bit is set when the PCI Master Module detects a data parity error, or the PCI Target Module detects a data or address parity error.	R	Undefined
30	S_SERR	Signaled PCI_SERRn 0 = PCI_SERRn not asserted 1 = PCI_SERRn asserted This bit is set by the PCI Target Module when it asserts PCI_SERRn to signal an address parity error. Note: SERR_EN and PERESP must be set before PCI_SERRn can be asserted.	R	Undefined
29	R_MA	Received Master Abort 0 = Master Abort is not detected 1 = Master Abort is detected This bit is set when a Tsi620-initiated transaction is terminated with a Master Abort.	R	Undefined
28	R_TA	Received Target Abort 0 = Target Abort is not detected 1 = Target Abort is detected This bit is set when a Tsi620-initiated transaction is terminated with a Target Abort.	R	Undefined

(Continued)

Bits	Name	Description	Type	Reset value
27	S_TA	Signaled Target Abort 0 = PCI Target Module did not terminate transaction with Target Abort 1 = PCI Target Module terminated transaction with Target Abort	R	Undefined
26:25	Reserved	Reserved	R	0
24	MDP_D	Master Data Parity Detected 0 = PCI Master Module did not detect or generate a data parity error 1 = PCI Master Module detected or generated a data parity error This bit is set if the PERESP bit is set and one of the following occurs: • The PCI Master Module is the master of a transaction in which it asserts PCI_PERRn • The addressed target asserts PCI_PERRn	R	Undefined
23:09	Reserved	Reserved	R	0
08	SERR_EN	PCI_SERRn Enable 0 = Disable PCI_SERRn driver 1 = Enable PCI_SERRn driver When SERR_EN and PERESP are set the Tsi620 can report address parity errors with PCI_SERRn as PCI target.	R	Undefined
07	Reserved	Reserved	R	0
06	PERESP	Parity Error Response 0 = Disable 1 = Enable Controls the device's response to address and data parity errors. When enabled, PCI_PERRn is asserted and the MDP_D bit is set in response to data parity errors. When this bit and SERR_EN are set, the device reports address parity errors on PCI_SERRn. This bit does not affect the device's parity generation.	R	Undefined
05:03	Reserved	Reserved	R	0
02	BM	Bus Master 0 = Disable 1 = Enable This bit enables the Tsi620 to generate transactions as a PCI master.	R	Undefined

(Continued)

Bits	Name	Description	Type	Reset value
01	MS	Memory Space 0 = Disable 1 = Enable This bit enables the Tsi620 to accept Memory transactions as a PCI target.	R	Undefined
00	IOS	I/O Space 0 = Disable 1 = Enable This bit enables the Tsi620 to accept I/O transactions as a PCI target.	R	Undefined

28.7.3 PCI SERRn Status Shadow Register

This register is a shadow of "PCI SERRn Status Register". The contents of this register are not reset by a chip reset, and are undefined after a power-up event. This register maintains its last state across a chip reset provided power is not lost. This state is maintained until a write of any value is performed to "PCI Control and Status Shadow Register". Writing to "PCI Control and Status Shadow Register" enables all the shadow registers for update. Once enabled, the shadow registers continually update to mirror the contents in the shadowed registers.

Register name: SERR_STAT_SHADOW **Reset value: Undefined**	**Register offset: 0x11C**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	Reserved							
23:16	Reserved	DR_ND	DW_ND	PW_MA	PW_TA	PW_ RETRY	PW_DPE	APE
15:08	Reserved							
07:00	Reserved							

Bits	**Name**	**Description**	**Type**	**Reset value**
31:23	Reserved	Reserved	R	0
22	DR_ND	Delayed Read No Data from Target PCI_SERRn was asserted due to a discard timer.	R	Undefined
21	DW_ND	Delayed Write Non Delivery PCI_SERRn was asserted due to a discard timer.	R	Undefined
20	PW_MA	Posted Write Master Abort PCI_SERRn was asserted due to a master abort.	R	Undefined
19	PW_TA	Posted Write Target Abort PCI_SERRn was asserted due to a target abort.	R	Undefined
18	PW_RETRY	Posted Write Maximum Retry PCI_SERRn was asserted after 2^{24} retries.	R	Undefined
17	PW_DPE	Posted Write Data Parity Error PCI_SERRn was asserted due to a data parity error.	R	Undefined
16	APE	Address Parity Error PCI_SERRn was asserted due to an address parity error.	R	Undefined
15:00	Reserved	Reserved	R	0

28.7.4 PCI Last Transaction Shadow Register

This register contains a copy of the last PCI transaction's internal queue header. The contents of this register are not reset by a chip reset, and are undefined after a power-up event. This register maintains its last state across a chip reset provided power is not lost. This state is maintained until a write of any value is performed to "PCI Control and Status Shadow Register". Writing to "PCI Control and Status Shadow Register" enables all the shadow registers for update. Once enabled, the shadow registers continually update to mirror the contents in the shadowed registers.

Register name: P_LAST_OP_SHADOW **Reset value: Undefined**	**Register offset: 0x128**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	Reserved							
23:16	Reserved							
15:08	Reserved				L	CMD		
07:00	BE_L					BE_R		

Bits	**Name**	**Description**	**Type**	**Reset value**
31:13	Reserved	Reserved	R	0
12	L	Last PCI transaction address size 0 = 32-bit address 1 = 64-bit address	R	Undefined
11:8	CMD	Last PCI command	R	Undefined
07:04	BE_L	Left-most byte enables with 64 bit PCI bus	R	Undefined
03:00	BE_R	Right-most byte enables with 64 or 32 bit PCI bus	R	Undefined

28.7.5 PCI Last Upper Address Shadow Register

This register contains a copy of the last PCI transaction's internal queue header. The contents of this register are not reset by a chip reset, and are undefined after a power-up event. This register maintains its last state across a chip reset provided power is not lost. This state is maintained until a write of any value is performed to "PCI Control and Status Shadow Register". Writing to "PCI Control and Status Shadow Register" enables all the shadow registers for update. Once enabled, the shadow registers continually update to mirror the contents in the shadowed registers.

Register name: P_LAST_ADDR_UPPER_SHADOW **Reset value: Undefined**	**Register offset: 0x12C**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	ADDR							
23:16	ADDR							
15:08	ADDR							
07:00	ADDR							

Bits	**Name**	**Description**	**Type**	**Reset value**
31:00	ADDR[63:32]	Upper part of last PCI transaction address	R	Undefined

28.7.6 PCI Last Lower Address Shadow Register

This register contains a copy of the last PCI transaction's internal queue header. The contents of this register are not reset by a chip reset, and are undefined after a power-up event. This register maintains its last state across a chip reset provided power is not lost. This state is maintained until a write of any value is performed to "PCI Control and Status Shadow Register". Writing to "PCI Control and Status Shadow Register" enables all the shadow registers for update. Once enabled, the shadow registers continually update to mirror the contents in the shadowed registers.

Register name: P_LAST_ADDR_LOWER_SHADOW **Reset value: Undefined**	**Register offset: 0x130**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	ADDR							
23:16	ADDR							
15:08	ADDR							
07:00	ADDR							

Bits	**Name**	**Description**	**Type**	**Reset value**
31:00	ADDR[31:0]	Lower part of last PCI transaction address	R	Undefined

28.7.7 PCI Last Attribute Shadow Register

The contents of this register are not reset by a chip reset, and are undefined after a power-up event. This register maintains its last state across a chip reset provided power is not lost. This state is maintained until a write of any value is performed to "PCI Control and Status Shadow Register". Writing to "PCI Control and Status Shadow Register" enables all the shadow registers for update. Once enabled, the shadow registers continually update to mirror the contents in the shadowed registers.

Register name: P_LAST_ATTR_SHADOW **Reset value: Undefined**	**Register offset: 0x134**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	Reserved	NS	RO	TAG				
23:16	BUS							
15:08	DEVICE					FUNCTION		
07:00	BYT_CNT							

Bits	**Name**	**Description**	**Type**	**Reset value**
31	Reserved	Reserved	R	0
30	NS	No Snoop	R	Undefined
29	RO	Relaxed ordering	R	Undefined
28:24	TAG	Transaction Tag	R	Undefined
23:16	BUS	Requester bus number	R	Undefined
15:11	DEVICE	Requester device number	R	Undefined
10:08	FUNCTION	Requester function number	R	Undefined
07:00	BYT_CNT	Transaction size	R	Undefined

28.7.8 PCI Interrupt Control Register

This register specifies the PCI interrupt capabilities.

Register name: IRP_CFG_CTL Reset value: 0x0000_0000	Register offset: 0x180

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	INTD_TYPE		INTC_TYPE		INTB_TYPE		INTA_TYPE	
07:00	Reserved		LOC_INT_DEST		INTD_DIR	INTC_DIR	INTB_DIR	INTA_DIR

Bits	Name	Description	Type	Reset value
31:16	Reserved	Reserved	R	0
15:14	INTD_TYPE	Destination of Interrupt 00 = Unused 01 = Reserved 10 = Reserved 11 = To Interrupt Controller	R/W	0x00
13:12	INTC_TYPE	See INTD_TYPE.	R/W	0x00
11:10	INTB_TYPE	See INTD_TYPE.	R/W	0x00
09:08	INTA_TYPE	Source or Destination of Interrupt 00 = Unused 01 = Reserved 10 = Reserved 11 = To/From Interrupt Controller	R/W	0x00
07:06	Reserved	Reserved	R	0
05:04	LOC_INT_DEST	Destination of PCI Interface Internal Interrupt 00 = Routes local PCI interrupt sources to the Interrupt Controller 01 = Routes local PCI interrupt sources to PCI_INTAn pin if PCI_INTAn is configured as an output (see Bit 00 in this register) 10 = Routes local PCI interrupt sources to an MSI event 11 = Reserved	R/W	0x00
03	INTD_DIR	PCI_INTDn Interrupt Direction 0 = Input	R/W	0

(Continued)

Bits	Name	Description	Type	Reset value
02	INTC_DIR	PCI_INTCn Interrupt Direction 0 = Input	R/W	0
01	INTB_DIR	PCI_INTBn Interrupt Direction 0 = Input	R/W	0
00	INTA_DIR	PCI_INTAn Interrupt Direction 0 = Input 1 = Output	R/W	0

28.7.9 PCI Interrupt Status Register

This register indicates the source of PCI interrupts. When set, each bit in this register indicates the corresponding interrupt source is active.

Register name: IRP_STAT Reset value: 0x0000_0000	Register offset: 0x184

Bits	7	6	5	4	3	2	1	0
31:24	Reserved	DR_ND	DW_ND	PW_MA	PW_TA	PW_ RETRY	PW_DPE	APE
23:16	P_CSR	P_INT	Reserved	HS_CSR	Reserved		FAB	Reserved
15:08	Reserved							
07:00	Reserved							

Bits	Name	Description	Type	Reset value
31	Reserved	Reserved	R	0
30	DR_ND	Copy of Delayed Read No Data from Target [a]	R	0
29	DW_ND	Copy of Delayed Write Non Delivery [a]	R	0
28	PW_MA	Copy of Posted Write Master Abort [a]	R	0
27	PW_TA	Copy of Posted Write Target Abort [a]	R	0
26	PW_RETRY	Copy of Posted Write Maximum Retry [a]	R	0
25	PW_DPE	Copy of Posted Write Data Parity Error [a]	R	0
24	APE	Copy of Address Parity Error [a]	R	0
23	P_CSR	There is an interrupt in the PE_CSR Register (see “PCI Control and Status Register”)	R	0
22	P_INT	There is an interrupt in the IRP_INTAD Register (see “PCI Interrupt CSR”)	R	0
21	Reserved	Reserved	R	0
20	HS_CSR	There is an interrupt in the P_HS_CSR Register.	R	0
19:18	Reserved	Reserved	R	00
17	FAB	Bridge ISF Interrupt. There is an interrupt in the PFAB_CSR Register.	R	0
16:00	Reserved	Reserved	R	0

a. See “PCI SERRn Status Register”.

28.7.10 PCI Interrupt Enable Register

This register enables the PCI interrupt functions.

Register name: IRP_ENABLE Reset value: 0x0000_0000	Register offset: 0x188

Bits	7	6	5	4	3	2	1	0
31:24	Reserved	DR_ND	DW_ND	PW_MA	PW_TA	PW_ RETRY	PW_DPE	APE
23:16	P_CSR	P_INT	Reserved	HS_CSR	Reserved		FAB	Reserved
15:08	Reserved							
07:00	Reserved							

Bits	Name	Description	Type	Reset value
31	Reserved	Reserved	R	0
30	DR_ND	Enable Delayed Read No Data from Target interrupt	R/W	0
29	DW_ND	Enable Delayed Write Non Delivery interrupt	R/W	0
28	PW_MA	Enable Posted Write Master Abort interrupt	R/W	0
27	PW_TA	Enable Posted Write Target Abort interrupt	R/W	0
26	PW_RETRY	Enable Maximum Retry interrupt	R/W	0
25	PW_DPE	Enable Posted Write Data Parity Error interrupt	R/W	0
24	APE	Enable Address Parity Error interrupt	R/W	0
23	P_CSR	Enable P_CSR Register interrupt	R/W	0
22	P_INT	Enable IRP_INTAD Register interrupt	R/W	0
21	Reserved	Reserved	R	0
20	HS_CSR	Enable P_HS_CSR Register interrupt	R/W	0
19:18	Reserved	Reserved	R	00
17	FAB	Enable PFAB_CSR Register interrupt	R/W	0
16:00	Reserved	Reserved	R	0

28.7.11 PCI Interrupt CSR

Register name: IRP_INTAD Reset value: 0x0000_0000	Register offset: 0x18C

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved				INTD_EN	INTC_EN	INTB_EN	INTA_EN
15:08	Reserved							
07:00	Reserved			PME	INTD	INTC	INTB	INTA

Bits	Name	Description	Type	Reset value
31:20	Reserved	Reserved	R	0
19	INTD_EN	Enable PCI_INTDn interrupt-to-interrupt controller. This bit controls no functionality in the Tsi620.	R/W	0
18	INTC_EN	Enable PCI_INTCn interrupt-to-interrupt controller. This bit controls no functionality in the Tsi620.	R/W	0
17	INTB_EN	Enable PCI_INTBn interrupt-to-interrupt controller. This bit controls no functionality in the Tsi620.	R/W	0
16	INTA_EN	Enable PCI_INTAn interrupt-to-interrupt controller. This bit controls no functionality in the Tsi620.	R/W	0
15:05	Reserved	Reserved	R	0
04	PME	Power transitioning interrupt. This bit is an inverted copy of PCI_PMEn. When set, this bit contributes to IRP_STAT[P_INT] which may be enabled to generate an interrupt (see "PCI Interrupt Status Register" and "PCI Interrupt Enable Register"). Note: Unlike the INT{A..D}n inputs, this bit indicates the state of PCI_PMEn if it is driven by the Tsi620 or an external agent.	R	0

(Continued)

Bits	Name	Description	Type	Reset value
03	INTD	INTD pending interrupt on PCI_INTDn. When set, this bit contributes to IRP_STAT[P_INT] which may be enabled to generate an interrupt (see "PCI Interrupt Status Register" and "PCI Interrupt Enable Register"). INTD is only set when PCI_INTDn is configured as an input (see "PCI Interrupt Control Register").	R	0
02	INTC	See INTD.	R	0
01	INTB	See INTD.	R	0
00	INTA	See INTD.	R	0

28.7.12 PCI Vendor ID Override Register

This register is written to update and overwrite the current settings for the "PCI ID Register".

Register name: P_VID_OVERRIDE **Reset value: 0x0000_0000**	**Register offset: 0x19C**

Bits	7	6	5	4	3	2	1	0
31:24	DID							
23:16	DID							
15:08	VID							
07:00	VID							

Bits	Name	Description	Type	Reset value
31:16	DID	Device ID A write to this field updates the read-only value at offset 0x00.	R/W1S	0
15:00	VID	Vendor ID A write to this field updates the read-only value at offset 0x00.	R/W1S	0

28.7.13 PCI Class Override Register

This register is written to update and overwrite the current settings for the "PCI Class Register".

Register name: P_CLASS_OVERRIDE Reset value: 0x0000_0000	Register offset: 0x1A0

Bits	7	6	5	4	3	2	1	0
31:24	BASE							
23:16	SUB							
15:08	PROG							
07:00	RID							

Bits	Name	Description	Type	Reset value
31:24	BASE	Base Class Code A write to this field updates the read-only value at register offset 0x008.	R/W1S	0
23:16	SUB	Sub Class Code A write to this field updates the read-only value at register offset 0x008.	R/W1S	0
15:08	PROG	Programming Interface A write to this field updates the read-only value at register offset 0x008.	R/W1S	0
07:00	RID	Revision ID A write to this field updates the read-only value at register offset 0x008.	R/W1S	0

28.7.14 PCI SID Override Register

This register is written to update and overwrite the current settings for the "PCI Subsystem ID Register".

Register name: PE_SID_OVERRIDE Reset value: 0x0000_0000	Register offset: 0x1A4

Bits	7	6	5	4	3	2	1	0
31:24	SID							
23:16	SID							
15:08	SVID							
07:00	SVID							

Bits	Name	Description	Type	Reset value
31:16	SID	Subsystem ID A write to this field updates the read-only value at register offset 0x02C.	R/W1S	0
15:00	SVID	Subsystem Vendor ID A write to this field updates the read-only value at register offset 0x02C.	R/W1S	0

28.7.15 PCI Slot ID Override Register

This register is written to update and overwrite the current settings for the "PCI Slot Identification Capabilities Register".

Register name: P_SLOT_ID_OVERRIDE Reset value: 0x0000_0000	Register offset: 0x1A8

Bits	7	6	5	4	3	2	1	0
31:24	CHASSIS							
23:16	Reserved		FIC	EXP_SLOT				
15:08	Reserved							
07:00	Reserved							

Bits	Name	Description	Type	Reset value
31:24	CHASSIS	Chassis Number A write to this field updates the read-only value at register offset 0x0C8.	R/W1S	0
23:22	Reserved	Reserved	R	0
21	FIC	First In Chassis A write to this field updates the read-only value at register offset 0x0C8.	R/W1S	0
20:16	EXP_SLOT	Expansion Slot A write to this field updates the read-only value at register offset 0x0C8.	R/W1S	0
15:00	Reserved	Reserved	R	0

28.7.16 PFAB Control and Status Register

Register name: PFAB_CSR Reset value: 0x0000_0000	Register offset: 0x200

Bits	7	6	5	4	3	2	1	0
31:24	O2P_BAR_RESP	SYNC_OVRN	BAD_P2O_PKT	TEA	Reserved	RESP_TIMOUT	Reserved	INVLD_RESP
23:16	INT_ENABLE_MASK							
15:08	BAR3_31	BAR3_30	BAR3_SIZE	BAR3_EN	TIME_EN1	OCN_RLX_ORDER	COMP_BAR_MODE	SW_RST
07:00	BAR4_31	BAR4_30	BAR4_SIZE	BAR4_EN	TIME_EN0	CFG_BSWAP	WSWAP	BSWAP

Bits	Name	Description	Type	Reset value
31	O2P_BAR_RESP	Bridge ISF to PCI BAR Response This bit is set when a response for an I/O write is discarded because it was generated by a Memory write that hits the PFAB_IO. This bit can indicate when the generation of the I/O write is finished.	R/W1C	0
30	SYNC_OVRN	Sync Packet Overrun This bit is reserved.	R/W1C	0
29	BAD_P2O_PKT	Bad PCI to Bridge ISF Packet This bit is set when a P2O packet is killed because the design entered an error state.	R/W1C	0
28	TEA	Timeout Error Acknowledgement This bit indicates a timeout error occurred on the Bridge ISF. Transaction was discarded.	R/W1C	0
27	Reserved	Reserved	R	0
26	RESP_TIMOUT	Request Timeout This bit is set when a request to the Bridge ISF timed out (see the TIME_EN0 and TIME_EN1 fields in this register).	R/W1C	0
25	Reserved	Reserved	R	0
24	INVLD_RESP	Invalid Response This bit is set when the PCI Interface received an unexpected response from the Bridge ISF.	R/W1C	0

(Continued)

Bits	Name	Description	Type	Reset value
23:16	INT_ENABLE_ MASK	Interrupt Enable Mask When one of these bits is set, that condition generates an internal hardware interrupt to the Interrupt Controller. For example, if INT_ENABLE_MASK[23] is set and the O2P_BAR_RESP status bit within this same register gets set due to an I/O write finishing, an internal hardware interrupt to the Interrupt Controller is generated. The purpose of the INT_ENABLE_MASK field is to allow interrupts defined within this register to be masked.	R/W	0
15	BAR3_31	Bit 31 of PFAB_PFM3 BAR when in non-compatible mode.	R/W	0
14	BAR3_30	Bit 30 of PFAB_PFM3 BAR when in non-compatible mode.	R/W	0
13	BAR3_SIZE	PFAB_PFM3 BAR Size when in non-compatible mode.	R/W	0
12	BAR3_EN	PFAB_PFM3 BAR Enable when in non-compatible mode.	R/W	0
11	TIME_EN1	Request Timer Value Bit 1 This bit, in conjunction with TIME_EN0, configures the PCI Interface's completion timers. When a request goes out to the Bridge ISF from the PCI Interface, a corresponding 16-bit timer discards the request if a response does not return from the Bridge ISF before the timer expires. This function helps prevent system deadlock. An error bit is also set which could generate an interrupt. Request timeout values: 00 = 2^18 clock timeout 01 = 2^17 clock timeout 10 = 2^16 clock timeout 11 = No timeout	R/W	0
10	OCN_RLX_ORDER	Bridge ISF Relaxed Order This bit controls no functionality in the Tsi620.	R/W	0
09	COMP_BAR_ MODE	Non-Compatible BAR Mode When this bit is set, the PFAB_PFM3 and PFAB_PFM4 BARs become [63:32] and the lower two bits are in this register, as well as BAR EN and BAR SIZE.	R/W	0
08	SW_RST	Reset Control of PCI block 0 = Normal operation 1 = Flushes Bridge ISF queues and resets their state machines	R/W	0
07	BAR4_31	Bit 31 of PFAB_PFM4 when in non-compatible mode.	R/W	0
06	BAR4_30	Bit 30 of PFAB_PFM4 when in non-compatible mode.	R/W	0
05	BAR4_SIZE	PFAB_PFM4 Size when in non-compatible mode.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset value
04	BAR4_EN	PFAB_PFM4 Enable when in non-compatible mode.	R/W	0
03	TIME_EN0	Request Timer Value Bit 0 See TIME_EN1.	R/W	0
02	CFG_BSWAP	Byte Swap PCI Register Accesses 0 = Do not swap registers 1 = Swap registers This bit enables byte swapping on Type 0 Configuration accesses. If this bit and the BSWAP bit are set to 1, Type 0 Configuration accesses are byte swapped.	R/W	0
01	WSWAP	32-bit within 64-bit Fabric Word Swap Enable 0 = Do not word swap 1 = Word swap When this bit is set to 1, the Tsi620 swaps the two 32-bit words of a 64-bit word. For example, a data pattern of 0x07060504_03020100 would be changed to 0x03020100_07060504. This bit has no effect on Type 0 Configuration accesses. Note: WSWAP should not be set in the Tsi620 because it can lead to illegal byte-enable combinations on the PCI bus or corrupted data.	R/W	0
00	BSWAP	Byte within 64-bit Fabric Word Swap Enable 0 = Do not byte swap 1 = Byte swap When this bit is set to 1, the Tsi620 swaps all bytes in a 64-bit word. For example, a data pattern of 0x07060504_03020100 would be changed to 0x00010203_04050607. This applies to all non-Type 0 Configuration transactions. For Type 0 Configuration transactions, the swapping is completed on a 32-bit dword. For example, a data pattern of 0x03020100 would be changed to 0x00010203. Note: BSWAP should not be set in the Tsi620 because it can lead to illegal byte-enable combinations on the PCI bus or corrupted data.	R/W	0

28.7.17 PFAB_BAR0 Register

This register specifies the Configuration address window for Bridge ISF to PCI transactions.

Register name: PFAB_BAR0 Reset value: 0x0000_0000	Register offset: 0x204

Bits	7	6	5	4	3	2	1	0
31:24	PFAB_BAR0							
23:16	Reserved							
15:08	Reserved							
07:00	Reserved							BAR0_EN

Bits	Name	Description	Type	Reset value
31:24	PFAB_BAR0[31:24]	PCI Configuration Generation Base Address These bits are compared with Bridge ISF address bits [31:24].	R/W	0
23:01	Reserved	Reserved	R	0
00	BAR0_EN	PFAB BAR0 Window Enable 0 = Disable 1 = Enable Note: Writing a 1 to this location enables the configuration address window. When enabled all Bridge ISF transactions with addresses that fall within the window get initiated as configuration cycles on the PCI bus.	R/W	0

28.7.18 PFAB_BAR0_UPPER Register

This register contains the upper address bits of the Configuration address window used for Bridge ISF to PCI transactions.

Register name: PFAB_BAR0_UPPER Reset value: 0x0000_0000	Register offset: 0x208

Bits	7	6	5	4	3	2	1	0
31:24	PFAB_BAR0							
23:16	PFAB_BAR0							
15:08	PFAB_BAR0							
07:00	PFAB_BAR0							

Bits	Name	Description	Type	Reset value
31:00	PFAB_BAR0[63:32]	PCI Configuration Generation Upper Base Address These bits are compared with Bridge ISF address bits [63:32].	R/W	0

28.7.19 PFAB_IO Base Address Register

This register specifies the I/O address window used for Bridge ISF to PCI transactions.

Register name: PFAB_IO Reset value: 0x0000_0000	Register offset: 0x20C

Bits	7	6	5	4	3	2	1	0
31:24	BAR							
23:16	BAR							
15:08	Reserved							
07:00	Reserved							EN

Bits	Name	Description	Type	Reset value
31:16	BAR[31:16]	PCI I/O Generation Base Address These bits are compared with the Bridge ISF address bits [31:16].	R/W	0
15:01	Reserved	Reserved	R	0
00	EN	I/O Window Enable 0 = Disable 1 = Enable Note: Writing a 1 to this location enables the I/O address window. When enabled all Bridge ISF transactions with addresses that fall within the window get initiated as I/O cycles on the PCI bus.	R/W	0

28.7.20 PFAB_IO_UPPER Register

This register contains the upper address bits of the I/O address window used for Bridge ISF to PCI transactions.

Register name: PFAB_IO_UPPER Reset value: 0x0000_0000	Register offset: 0x210

Bits	7	6	5	4	3	2	1	0
31:24	BAR							
23:16	BAR							
15:08	BAR							
07:00	BAR							

Bits	Name	Description	Type	Reset value
31:00	BAR[63:32]	PCI I/O Generation Base Address These bits are compared with the Bridge ISF address bits [63:32].	R/W	0

28.7.21 PFAB_MEM32 Base Address Register

This register specifies the Memory address window used for Bridge ISF to PCI transactions.

Register name: PFAB_MEM32 Reset value: 0x0000_0000	Register offset: 0x214

Bits	7	6	5	4	3	2	1	0
31:24	BA			Reserved				
23:16	Reserved						SIZE	EN
15:08	Reserved							
07:00	Reserved							

Bits	Name	Description	Type	Reset value
31:29	BA[31:29]	32-bit Memory (Mem32) Base Address These bits are compared with Bridge ISF address bits [31:29].	R/W	0
28:18	Reserved	Reserved	R	0
17	SIZE	Mem32 Window Size 0 = 512 MB 1 = 1 GB	R/W	0
16	EN	Mem32 Window Enable 0 = Disable 1 = Enable Note: Writing a 1 to this location enables the Mem32 address window. When enabled all Bridge ISF transactions with addresses that fall within the window get translated as defined by the appropriate remap and mask registers defined in “PFAB_MEM32_REMAP Register” and “PFAB_MEM32_MASK Register” onto the PCI bus.	R/W	0
15:00	Reserved	Reserved	R	0

28.7.22 PFAB_MEM32_REMAP Register

This register contains the Memory remap value that translates Bridge ISF addresses to PCI addresses.

Register name: PFAB_MEM32_REMAP Reset value: 0x0000_0000	Register offset: 0x218

Bits	7	6	5	4	3	2	1	0
31:24	Remap							
23:16	Remap							
15:08	Remap				Reserved			
07:00	Reserved							

Bits	Name	Description	Type	Reset value
31:12	Remap[31:12]	Mem32 Remap Value These bits are substituted with the Bridge ISF address bits [31:12] provided the corresponding Mask value is enabled.	R/W	0
11:00	Reserved	Reserved	R	0

28.7.23 PFAB_MEM32_MASK Register

This register contains the Memory mask value that specifies the window size for translating Bridge ISF addresses to PCI addresses.

Register name: PFAB_MEM32_MASK Reset value: 0x0000_0000	Register offset: 0x21C

Bits	7	6	5	4	3	2	1	0
31:24	Mask							
23:16	Mask							
15:08	Mask					Reserved		
07:00	Reserved							

Bits	Name	Description	Type	Reset value
31:12	Mask	Mem32 Mask Value These bits when enabled, enable the substitution of the Bridge ISF address bits [31:12] with the Mem32 Remap Value Remap[31:12]. Note: Setting Mask[28:12] results in aliasing within the BAR as should be avoided (see "PFAB_MEM32_REMAP and PFAB_MEM32_MASK Register Operation").	R/W	0
11:00	Reserved	Reserved	R	0

28.7.24 PFAB_PFM3 Base Address Register

This register specifies the Prefetchable Memory 3 (PFM3) address window for Bridge ISF to PCI transactions.

Register name: PFAB_PFM3 Reset value: 0x0000_0000	Register offset: 0x220

Bits	7	6	5	4	3	2	1	0
31:24	BA_31_30		BA_59_48					
23:16	BA_59_48						SIZE	EN
15:08	BA_47_32							
07:00	BA_47_32							

Bits	Name	Description	Type	Reset value
31:30	BA_31_30	PFM3 Base Address[31:30] These bits are compared with the Bridge ISF address [31:30].	R/W	0
29:18	BA_59_48	PFM3 Base Address[59:48] These bits are compared with the Bridge ISF address bits [59:48].[a]	R/W	0
17	SIZE	PFM3 Window Size 0 = 1 GB 1 = 2 GB	R/W	0
16	EN	PFM3 Window Enable 0 = Disable 1 = Enable Note: Writing a 1 to this location enables the PFAB_PFM3 address window. When enabled all Bridge ISF transactions with addresses that fall within the window get translated as defined by the appropriate remap and mask registers defined in “PFAB_PFM3_REMAP_UPPER Register” through “PFAB_PFM3_MASK Register” onto the PCI bus.	R/W	0
15:00	BA_47_32	PFM3 Base Address[47:32] These bits are compared with internal Bridge ISF address bits [47:32].	R/W	0

a. The BA[63:60] value is hard-wired to 0000.

28.7.25 PFAB_PFM3_REMAP_UPPER Register

This register contains the Prefetchable Memory 3 remap value that translates Bridge ISF addresses to PCI addresses.

Register name: PFAB_PFM3_REMAP_UPPER Reset value: 0x0000_0000	Register offset: 0x224

Bits	7	6	5	4	3	2	1	0
31:24	Remap							
23:16	Remap							
15:08	Remap					Reserved		
07:00	Reserved							

Bits	Name	Description[a]	Type	Reset value
31:12	Remap[63:44]	PFM3 Upper Remap These bits are substituted with the Bridge ISF address bits [63:44] provided the corresponding Mask value is enabled.	R/W	0
11:00	Reserved	Reserved	R	0

a. Bits [11:0] of this register can be used as Remap[43:32]. It is the logical OR of PFAB_PFM3_REMAP_UPPER[11:0] and PFAB_PFM3_REMAP_LOWER[11:0].

28.7.26 PFAB_PFM3_REMAP_LOWER Register

This register contains the lower bits in the Prefetchable Memory 3 remap value that translates Bridge ISF addresses to PCI addresses.

Register name: PFAB_PFM3_REMAP_LOWER Reset value: 0x0000_0000	Register offset: 0x228

Bits	7	6	5	4	3	2	1	0
31:24	Remap_31_12							
23:16	Remap_31_12							
15:08	Remap_31_12				Remap_43_32			
07:00	Remap_43_32							

Bits	Name	Description	Type	Reset value
31:12	Remap_31_12	PFM3 Lower Remap These bits are substituted with the Bridge ISF address bits [31:12] provided the corresponding Mask value is enabled.	R/W	0
11:0	Remap_43_32	PFM3 Lower Remap These bits are substituted with the Bridge ISF address bits [43:32] provided the corresponding Mask value is enabled.	R/W	0

28.7.27 PFAB_PFM3_MASK Register

This register contains the Prefetchable Memory 3 mask value that specifies the window size for translating Bridge ISF addresses to PCI addresses.

Register name: PFAB_PFM3_MASK Reset value: 0x0000_0000	Register offset: 0x22C

Bits	7	6	5	4	3	2	1	0
31:24	Mask_31_12							
23:16	Mask_31_12							
15:08	Mask_31_12				Mask_43_32			
07:00	Mask_43_32							

Bits	Name	Description[a]	Type	Reset value
31:12	Mask_31_12	PFM3 Mask Value These bits when enabled, enable the substitution of the Bridge ISF address bits [31:12] with the PFAB_PFM3 Remap Value Remap[31:12].	R/W	0
11:0	Mask_43_32	PFM3 Mask Value These bits when enabled, enable the substitution of the Bridge ISF address bits [43:32] with the PFAB_PFM3 Remap Value Remap[43:32].	R/W	0

a. Remap[63:44] are substituted for Bridge ISF address bits [63:44] as the Mask bits only provide selectable replacement for bits [43:12] of the original Bridge ISF address.

28.7.28 PFAB_PFM4 Base Address Register

This register specifies the Prefetchable Memory 4 (PFM4) address window used for Bridge ISF to PCI transactions.

Register name: PFAB_PFM4 Reset value: 0x0000_0000	Register offset: 0x230

Bits	7	6	5	4	3	2	1	0
31:24	BA_31_30		BA_59_48					
23:16	BA_59_48						SIZE	EN
15:08	BA_47_32							
07:00	BA_47_32							

Bits	Name	Description[a]	Type	Reset value
31:30	BA_31_30	PFM4 Base Address [31:30]. These bits are compared with the Bridge ISF address bits [31:30].	R/W	0
29:18	BA_59_48	PFM4 Base Address [59:48]. These bits are compared with the internal Bridge ISF address bits [59:48].	R/W	0
17	SIZE	PFM4 window size 0 = 1 GB 1 = 2 GB	R/W	0
16	EN	PFM4 Window Enable 0 = Disable 1 = Enable Note: Writing a 1 to this location enables the PFAB_PFM4 address window. When enabled all Bridge ISF transactions with addresses that fall within the window get translated as defined by the appropriate remap and mask registers defined in “PFAB_PFM4_REMAP_UPPER Register” through “PFAB_PFM4_MASK Register” onto the PCI bus.	R/W	0
15:00	BA_47_32	PFM4 Base Address [47:32]. These bits are compared with the Bridge ISF address bits [47:32].	R/W	0

a. The BA[63:60] value is hard-wired to 0000.

28.7.29 PFAB_PFM4_REMAP_UPPER Register

This register contains the upper bits in the Prefetchable Memory 4 remap value that translates Bridge ISF addresses to PCI addresses.

Register name: PFAB_PFM4_REMAP_UPPER Reset value: 0x0000_0000	Register offset: 0x234

Bits	7	6	5	4	3	2	1	0
31:24	Remap							
23:16	Remap							
15:08	Remap					Reserved		
07:00	Reserved							

Bits	Name	Description[a]	Type	Reset value
31:12	Remap[63:44]	PFM4 Upper Remap These bits are substituted with the Bridge ISF address bits [63:44] provided the corresponding Mask value is enabled.	R/W	0
11:00	Reserved	Reserved	R	0

a. Bits [11:0] of this register can be used as Remap[43:32]. It is the logical OR of PFAB_PFM4_REMAP_UPPER[11:0] and PFAB_PFM4_REMAP_LOWER[11:0].

28.7.30 PFAB_PFM4_REMAP_LOWER Register

This register contains the lower bits in the Prefetchable Memory 4 remap value that translates Bridge ISF addresses to PCI addresses.

Register name: PFAB_PFM4_REMAP_LOWER Reset value: 0x0000_0000	Register offset: 0x238

Bits	7	6	5	4	3	2	1	0
31:24	Remap_31_12							
23:16	Remap_31_12							
15:08	Remap_31_12				Remap_43_32			
07:00	Remap_43_32							

Bits	Name	Description	Type	Reset value
31:12	Remap_31_12	PFM4 Lower Remap These bits are substituted with the Bridge ISF address bits [31:12] provided the corresponding Mask value is enabled.	R/W	0
11:0	Remap_43_32	PFM4 Lower Remap These bits are substituted with the Bridge ISF address bits [43:32] provided the corresponding Mask value is enabled.	R/W	0

28.7.31 PFAB_PFM4_MASK Register

This register contains the Prefetchable Memory 4 mask value that specifies the window size for translating Bridge ISF addresses to PCI addresses.

Register name: PFAB_PFM4_MASK Reset value: 0x0000_0000	Register offset: 0x23C

Bits	7	6	5	4	3	2	1	0
31:24	Mask_31_12							
23:16	Mask_31_12							
15:08	Mask_31_12				Mask_43_32			
07:00	Mask_43_32							

Bits	Name	Description[a]	Type	Reset value
31:12	Mask_31_12	PFM4 Mask Value These bits when enabled, enable the substitution of the Bridge ISF address bits [31:12] with the PFAB_PFM4 Remap Value Remap[31:12].	R/W	0
11:0	Mask_43_32	PFM4 Mask Value These bits when enabled, enable the substitution of the Bridge ISF address bits [43:32] with the PFAB_PFM4 Remap Value Remap[43:32].	R/W	0

a. Remap[63:44] are substituted for Bridge ISF address bits [63:44] as the Mask bits only provide selectable replacement for bits [43:12] of the original Bridge ISF address.

28.7.32 Expansion ROM Map Register

Register name: EROM_MAP Reset value: 0x0000_0000	Register offset: 0x2FC

Bits	7	6	5	4	3	2	1	0
31:24	MAP							Reserved
23:16	Reserved							
15:08	Reserved							
07:00	Reserved				DST			

Bits	Name	Description	Type	Reset value
31:25	MAP	Expansion ROM Map Address These bits replace BA[6:0] of the PE_ROM when using the "PCI Expansion ROM Register".	R/W	0
24:04	Reserved	Reserved	R	0
3:0	DST	Expansion ROM Port Destination Expansion ROM destination port, defaults to RapidIO.	R/W	0

28.7.33 P2O_BAR2_LUT{0..31} Register

This lookup table specifies the lower translated address and destination ID for each of the 31 pages of the PFM BAR (P2O_BAR2) address window.

Register name: P2O_BAR2_LUT{0..31} **Reset value: Undefined**	**Register offset: 0x500, 508,..., 5F8**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	BAR2_PAGE_ADDR							
23:16	BAR2_PAGE_ADDR							
15:08	BAR2_PAGE_ADDR							Reserved
07:00	Reserved					BAR2_DESTID		

Bits	**Name**	**Description**	**Type**	**Reset value**
31:10	BAR2_PAGE_ ADDR[31:10]	PCI BAR2 Translation Address When the BAR2_NOTRAN bit in the P2O_PAGE_SIZES register is a zero, these bits replace the lower PCI bus address bits. The bits replaced depend on the Size of the Address window.	R/W	Undefined
09:04	Reserved	Reserved	R	0
3:00	BAR2_DESTID	PCI BAR2 Destination ID Bridge ISF Destination Port Number. Each Tsi620 external interface is connected to a Bridge ISF port as follows: 0000 = SREP Interface 0001 = PCI Interface	R/W	Undefined

28.7.34 P2O_BAR2_LUT_UPPER{0..31} Register

This lookup table specifies the upper translated address for each of the 31 pages of the PFM BAR (P2O_BAR2) address window.

Register name: P2O_BAR2_LUT_UPPER{0..31} Reset value: Undefined	Register offset: 0x504, 50C,..., 5FC

Bits	7	6	5	4	3	2	1	0
31:24	BAR2_PAGE_ADDR							
23:16	BAR2_PAGE_ADDR							
15:08	BAR2_PAGE_ADDR							
07:00	BAR2_PAGE_ADDR							

Bits	Name	Description	Type	Reset value
31:0	BAR2_PAGE_ADDR[63:32]	PCI BAR2 Translation Address When P2O_PAGE_SIZES[BAR2_NOTRAN] is 0, these bits replace the upper PCI bus address bits. The bits replaced depend on the size of the address window.	R/W	Undefined

28.7.35 P2O_BAR3_LUT{0..31} Register

This lookup table specifies the lower translated address and destination ID for each of the 31 pages of the Memory BAR (P2O_BAR3) address window.

Register name: P2O_BAR3_LUT{0..31} **Reset value: Undefined**	**Register offset: 0x600, 608,..., 6F8**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	BAR3_PAGE_ADDR							
23:16	BAR3_PAGE_ADDR							
15:08	BAR3_PAGE_ADDR							Reserved
07:00	Reserved				BAR3_DESTID			

Bits	**Name**	**Description**	**Type**	**Reset value**
31:10	BAR3_PAGE_ ADDR[31:10]	PCI BAR3 Translation Address When P2O_PAGE_SIZES[BAR3_NOTRAN] is 0, these bits replace the lower PCI bus address bits. The bits replaced depend on the Size of the Address window.	R/W	Undefined
09:04	Reserved	Reserved	R	0
3:0	BAR3_DESTID	PCI BAR3 Destination ID Bridge ISF Destination Port Number. Each Tsi620 external interface is connected to a Bridge ISF port as follows: 0000 = SREP Interface 0001 = PCI Interface	R/W	Undefined

28.7.36 P2O_BAR3_LUT_UPPER{0..31} Register

This lookup table specifies the upper translated address for each of the 31 pages of the Memory BAR (P2O_BAR3) address window.

Register name: P2O_BAR3_LUT_UPPER{0..31} Reset value: Undefined	Register offset: 0x604, 60C,..., 6FC

Bits	7	6	5	4	3	2	1	0
31:24	BAR3_PAGE_ADDR							
23:16	BAR3_PAGE_ADDR							
15:08	BAR3_PAGE_ADDR							
07:00	BAR3_PAGE_ADDR							

Bits	Name	Description	Type	Reset value
31:0	BAR3_PAGE_ADDR[63:32]	PCI BAR3 Translation Address When P2O_PAGE_SIZES[BAR3_NOTRAN] is 0, these bits replace the upper PCI bus address bits. The bits replaced depend on the size of the address window.	R/W	Undefined

29. I²C Registers

Topics discussed include the following:

- "Register Map"
- "Register Descriptions"

29.1 Register Map

The following table lists the register map for the I²C registers.

All registers can be accessed through the internal register bus. A portion of the register space is visible to an external I²C master through the slave interface. The registers in this portion have a *peripheral address* in addition to their internal register bus address, and are called the *externally visible* I²C registers, meaning they can be accessed by the external I²C master using I²C reads and writes (see Table 189). The peripheral address equates to a 4-byte range within a consecutive 256-byte address space. For peripheral addresses, the lowest address maps to the least significant byte of the internal register (LSB), while the highest address maps to the most significant byte of the internal register (MSB).

Table 189: I²C Register Map

Internal Address	Peripheral Address	Register Name	See
0x000– 0x0FC	n/a	Reserved	
0x100	n/a	I2C_DEVID	"I2C Device ID Register"
0x104	n/a	I2C_RESET	"I2C Reset Register"
0x108	n/a	I2C_MST_CFG	"I2C Master Configuration Register"
0x10C	n/a	I2C_MST_CNTRL	"I2C Master Control Register"
0x110	n/a	I2C_MST_RDATA	"I2C Master Receive Data Register"
0x114	n/a	I2C_MST_TDATA	"I2C Master Transmit Data Register"
0x118	n/a	I2C_ACC_STAT	"I2C Access Status Register"
0x11C	n/a	I2C_INT_STAT	"I2C Interrupt Status Register"
0x120	n/a	I2C_INT_ENABLE	"I2C Interrupt Enable Register"
0x124	n/a	I2C_INT_SET	"I2C Interrupt Set Register"
0x12C	n/a	I2C_SLV_CFG	"I2C Slave Configuration Register"

Table 189: I²C Register Map (Continued)

Internal Address	Peripheral Address	Register Name	See
0x130–0x13C	n/a	Reserved	
0x140	n/a	I2C_BOOT_CNTRL	"I2C Boot Control Register"
0x144–0x1FC	n/a	Reserved	
0x200	0x00–0x03	EXI2C_REG_WADDR	"Externally Visible I2C Internal Write Address Register"
0x204	0x04–0x07	EXI2C_REG_WDATA	"Externally Visible I2C Internal Write Data Register"
0x208–0x20C	0x08–0x0F	Reserved	
0x210	0x10–0x13	EXI2C_REG_RADDR	"Externally Visible I2C Internal Read Address Register"
0x214	0x14–0x17	EXI2C_REG_RDATA	"Externally Visible I2C Internal Read Data Register"
0x218–0x21C	0x18–0x1F	Reserved	
0x220	0x20–0x23	EXI2C_ACC_STAT	"Externally Visible I2C Slave Access Status Register"
0x224	0x24–0x27	EXI2C_ACC_CNTRL	"Externally Visible I2C Internal Access Control Register"
0x228–0x27C	0x28–0x7F	Reserved	
0x280	0x80–0x83	EXI2C_STAT	"Externally Visible I2C Status Register"
0x284	0x84–0x87	EXI2C_STAT_ENABLE	"Externally Visible I2C Enable Register"
0x288–0x28C	0x88–0x8F	Reserved	
0x290	0x90–0x93	EXI2C_MBOX_OUT	"Externally Visible I2C Outgoing Mailbox Register"
0x294	0x94–0x97	EXI2C_MBOX_IN	"Externally Visible I2C Incoming Mailbox Register"
0x298–0x2FC	0x98–0xFF	Reserved	
0x300	n/a	I2C_EVENT	"I2C Event and Event Snapshot Registers"
0x304	n/a	I2C_SNAP_EVENT	"I2C Event and Event Snapshot Registers"
0x308	n/a	I2C_NEW_EVENT	"I2C New Event Register"
0x30C	n/a	I2C_EVENT_ENB	"I2C Enable Event Register"

Table 189: I²C Register Map (Continued)

Internal Address	Peripheral Address	Register Name	See
0x310–0x31C	n/a	Reserved	
0x320	n/a	I2C_DIVIDER	"I2C Time Period Divider Register"
0x324–0x33C	n/a	Reserved	
0x340	n/a	I2C_START_SETUP_HOLD	"I2C Start Condition Setup/Hold Timing Register"
0x344	n/a	I2C_STOP_IDLE	"I2C Stop/Idle Timing Register"
0x348	n/a	I2C_SDA_SETUP_HOLD	"I2C_SD Setup and Hold Timing Register"
0x34C	n/a	I2C_SCL_PERIOD	"I2C_SCLK High and Low Timing Register"
0x350	n/a	I2C_SCL_MIN_PERIOD	"I2C_SCLK Minimum High and Low Timing Register"
0x354	n/a	I2C_SCL_ARB_TIMEOUT	"I2C_SCLK Low and Arbitration Timeout Register"
0x358	n/a	I2C_BYTE_TRAN_TIMEOUT	"I2C Byte/Transaction Timeout Register"
0x35C	n/a	I2C_BOOT_DIAG_TIMER	"I2C Boot and Diagnostic Timer"
0x360–0x3B4	n/a	Reserved	
0x3B8	n/a	I2C_BOOT_DIAG_PROGRESS	"I2C Boot Load Diagnostic Progress Register"
0x3BC	n/a	I2C_BOOT_DIAG_CFG	"I2C Boot Load Diagnostic Configuration Register"
0x3C0–0x3FC	n/a	Reserved	

29.2 Register Descriptions

This section describes the I^2C registers. These registers are reset by a chip reset.

29.2.1 I^2C Device ID Register

This register identifies the version of the IDT I^2C block in this device.

Register name: I2C_DEVID **Reset value: 0x0000_0001**	**Register offset: 0x100**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							
24:31	Reserved				REV			

Bits	**Name**	**Description**	**Type**	**Reset Value**
00:27	Reserved	Reserved	R	0x000_0000
28:31	REV	Indicates the revision ID for the I^2C block.	R	0x1

29.2.2 I²C Reset Register

This register completes a reset of the I²C block. This reset returns the logic to its idle, non-transacting state while retaining all configuration registers, such that the block does not have to be reprogrammed. This is provided for exceptional conditions. A reset while the block is involved in a transaction as a master or slave may leave the bus in an unexpected state relative to any external I²C masters or slaves, and thus should be used with caution and only as a last solution if the block seems unresponsive.

I²C registers that are affected by a this reset are indicated in the description of that register. All other registers should be assumed to be unaffected by this reset.

Register name: I2C_RESET **Reset value: 0x0000_0000**	**Register offset: 0x104**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	SRESET	Reserved						
08:15	Reserved							
16:23	Reserved							
24:31	Reserved							

Bits	**Name**	**Description**	**Type**	**Reset Value**
00	SRESET	Reset under Software Control Setting this bit resets the I²C block. The R/W fields of configuration and control registers are not affected (nor is this register affected). While in reset neither the Master nor slave interface will be operational: the I2C_SCLK and I2C_SD signals will be undriven so as to not obstruct the bus. This bit must be written to 0 to bring the block out of reset. Any active bus transactions while reset occurs are aborted. All status and events are returned to the reset state. The boot load sequence will not be invoked upon exit from reset, although the bus idle detect sequence will be invoked. Power-up latch values will not be re-latched, and the fields will remain at their pre-soft-reset value.	R/W	0
01:31	Reserved	Reserved	R	0

29.2.3 I2C Master Configuration Register

This register contains options that apply to master operations initiated through the "I2C Master Control Register". The configuration specifies the properties of the external slave device to which a read or write transaction will be directed.

Register name: I2C_MST_CFG Reset value: Undefined	Register offset: 0x108

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	DORDER	Reserved					PA_SIZE	
16:23	Reserved							
24:31	Reserved	DEV_ADDR						

Bits	Name	Description	Type	Reset Value
00:07	Reserved	Reserved	R	0x00
08	DORDER	Data Order 0 = Data from/to data registers is ordered (processed) from MSB to LSB within an I2C transaction. 1 = Data from/to data registers is ordered (processed) from LSB to MSB within an I2C transaction. Data registers are "I2C Master Transmit Data Register" and "I2C Master Receive Data Register".	R/W	0
09:13	Reserved	Reserved	R	0x00
14:15	PA_SIZE	Peripheral Address Size 00 = No peripheral address used 01 = 8-bit peripheral device addressing using LSB of PADDR 10 = 16-bit peripheral device addressing using MSB and LSB of PADDR (in that order) 11 = Reserved (handled as 00) This field selects the number of bytes in the peripheral address for master transactions. The peripheral address itself is specified in the "I2C Master Control Register".	R/W	Undefined
16:24	Reserved	Reserved	R	0x000
25:31	DEV_ADDR	Device Address Specifies the 7-bit device address to select the I2C device for a read or write transaction initiated through the "I2C Master Control Register".	R/W	Undefined

Do not change this register while a master operation is active. The effect on the transaction cannot be determined.

29.2.4 I²C Master Control Register

This register sets the peripheral address and to start an I²C transaction. The transaction is directed to the device defined in the "I2C Master Configuration Register".

Note: Software must not set the peripheral address and the SIZE parameters such that unintended page wrap-arounds occur in the target device. The Tsi620 does not force repeated start conditions within a single software initiated access.

Register name: I2C_MST_CNTRL **Reset value: 0x0000_0000**	**Register offset: 0x10C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	START	WRITE	Reserved			SIZE		
08:15	Reserved							
16:23	PADDR							
24:31	PADDR							

Bits	**Name**	**Description**	**Type**	**Reset Value**
00	START	Start Operation 0 = I²C operation is not in progress (self clears) 1 = Start of an I²C operation. Clears itself back to zero when the initiated operation has completed. This bit cannot be cleared by software once set, except through a reset under software control. Note: This bit is cleared by a reset initiated by the "I2C Reset Register".	R/W1S	0
01	WRITE	I²C Read or Write 0 = Read from I²C memory 1 = Write to I²C memory For a read, data is returned to the "I2C Master Receive Data Register". For a write, data is taken from the "I2C Master Transmit Data Register".	R/W	0
02:04	Reserved	Reserved	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
05:07	SIZE	Number of bytes in an I^2C operation (read or write) 000 = 0 bytes 001 = 1 bytes 010 = 2 bytes 011 = 3 bytes 100 = 4 bytes 101 = Reserved (equivalent of 0 bytes) 110 = Reserved (equivalent of 0 bytes) 111 = Reserved (equivalent of 0 bytes) A value of 000 should not be normally used for a Read operation, as any device put into read mode assumes at least one byte will be read. An exception would be the SMBus Quick Command protocol to a device that is known to not hold the bus following the slave address phase.	R/W	000
08:15	Reserved	Reserved	R	0
16:31	PADDR	Peripheral address for master operation If PA_SIZE in the "I2C Master Configuration Register" is 10, then all 16 bits are used, with the most significant 8 bits placed on I^2C bus first, followed by the least significant 8 bits. If PA_SIZE is 01, then only least significant 8 bits are used. If PA_SIZE is 00, this field is not used for the transaction.	R/W	0x0000

A master operation is initiated by writing this register and setting the START bit to 1. The same write should set the WRITE, SIZE, and PADDR fields as required by the transaction. Other information for the transaction must have been pre-loaded into the "I2C Master Configuration Register".

Do not change this register while a master operation is active. The effect on the transaction cannot be determined.

The following is the sequence triggered by setting the START bit:

Table 190: Master Operation Sequence

Phase	Description	Outcome
1. Begin transaction	Start arbitration timer.	-
2. Address slave	Detect bus idle. Send START. Send I2C_MST_CFG.[DEV_ADDR]. Send Read or Write (normally Write unless WRITE=0 and PA_SIZE=0). Wait for ACK/NACK. Disable arbitration timer.	Any loss of arbitration repeats this phase. Phase completes with ACK. NACK terminates operation with a MNACK event (issues STOP). Expiration of arbitration timer aborts operation with MARBTO.
3. Peripheral Address	If PA_SIZE = 10, send PADDR[15:8] and wait for ACK. If PA_SIZE = 01 or 10, send PADDR[7:0] and wait for ACK.	If PA_SIZE is 00 or 11, this phase is skipped. Any loss of arbitration will abort with MCOL. Any NACK will abort with MNACK.
4. Send Data (WRITE = 1)	If SIZE > 0, send SIZE bytes from I2C_MST_TDATA based on DORDER. Wait for ACK from each.	This phase is skipped if WRITE=1. Any loss of arbitration will abort with MCOL. Any NACK will abort with MNACK.
5. Read Data Setup (WRITE = 0)	Send RESTART. Repeat the “Address Slave” process with last bit a Read. Wait for ACK/NACK.	This phase is skipped if WRITE=1 or SIZE=0. NACK aborts with MNACK. Loss of arbitration aborts with MCOL (because bus was never released).
6. Read Data (WRITE = 0)	Read SIZE bytes and place in I2C_MST_RDATA based on DORDER. Respond with ACK to each, except for final byte respond with a NACK.	This phase is skipped if WRITE=1 or SIZE=0. Any loss of arbitration aborts with MCOL.
7. Complete	Issue STOP. Clear I2C_MST_CNTRL[START]. Set MSD event.	Except for arbitration loss, master always tries to force a STOP condition.

29.2.5 I2C Master Receive Data Register

This register contains the data read from an external slave device following a read operation initiated using the "I2C Master Control Register".

As bytes are read from the I2C bus, they are placed in this register depending on DORDER in the "I2C Master Configuration Register". If DORDER is 0, bytes are loaded from MSB to LSB, in order: RBYTE3, RBYTE2, RBYTE1, RBYTE0. If DORDER is 1, bytes are loaded from LSB to MSB, in order: RBYTE0, RBYTE1, RBYTE2, RBYTE3. If the transaction size is less than four (4) bytes (that is, SIZE in the "I2C Master Control Register" < 4) then any remaining bytes in the register are left unchanged (that is, they retain the values they had from the prior read operation).

Register name: I2C_MST_RDATA **Reset value: 0x0000_0000**	**Register offset: 0x110**

Bits	0	1	2	3	4	5	6	7
00:07	RBYTE3							
08:15	RBYTE2							
16:23	RBYTE1							
24:31	RBYTE0							

Bits	Name	Description	Type	Reset value
00:07	RBYTE3	Received I2C data — Byte 3 (most significant)	R	0x00
08:15	RBYTE2	Received I2C data — Byte 2	R	0x00
16:23	RBYTE1	Received I2C data — Byte 1	R	0x00
24:31	RBYTE0	Received I2C data — Byte 0 (least significant)	R	0x00

29.2.6 I2C Master Transmit Data Register

This register contains the data to be written (transmitted) to an external slave when a write operation is initiated using the "I2C Master Control Register". This register should be written with data to be sent prior to setting the START bit in that register.

As bytes are written to the I2C bus, they are taken from this register depending on DORDER in the "I2C Master Configuration Register". If DORDER is 0, bytes are taken from MSB to LSB, in order: TBYTE3, TBYTE2, TBYTE1, TBYTE0. If DORDER is 1, bytes are taken from LSB to MSB, in order: TBYTE0, TBYTE1, TBYTE2, TBYTE3. If the transaction size is less than 4 bytes (that is, SIZE in the "I2C Master Control Register" <4) then any remaining bytes in the register are unused. The contents of this register are not affected by the transaction.

Register name: I2C_MST_TDATA **Reset value: 0x0000_0000**	**Register offset: 0x114**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	TBYTE3							
08:15	TBYTE2							
16:23	TBYTE1							
24:31	TBYTE0							

Bits	**Name**	**Description**	**Type**	**Reset value**
00:07	TBYTE3	Transmitted I2C data — Byte 3 (most significant)	R/W	0x00
08:15	TBYTE2	Transmitted I2C data — Byte 2	R/W	0x00
16:23	TBYTE1	Transmitted I2C data — Byte 1	R/W	0x00
24:31	TBYTE0	Transmitted I2C data — Byte 0 (least significant)	R/W	0x00

Do not change this register while a master operation is active. The effect on the transaction cannot be determined.

29.2.7 I²C Access Status Register

This register indicates the status of the I²C block. Fields in this register change dynamically as operations are initiated or progress.

Register name: I2C_ACC_STAT Reset value: 0x0000_0000	Register offset: 0x118

Bits	0	1	2	3	4	5	6	7
00:07	SLV_ ACTIVE	BUS_ ACTIVE	Reserved		SLV_WAIT	SLV_PHASE		SLV_AN
08:15	SLV_PA							
16:23	MST_ ACTIVE	Reserved			MST_PHASE			MST_AN
24:31	Reserved				MST_NBYTES			

Bits	Name	Description	Type	Reset Value
00	SLV_ACTIVE	Slave Active 0 = Slave is not addressed 1 = Slave is addressed by external master and a read or write is active on the bus This bit is set following the slave address phase if the address matched the SLV_ADDR or Alert Response Address and the slave interface was enabled. Note: This bit is zeroed on a reset controlled by the "I2C Reset Register".	R	0
01	BUS_ACTIVE	Bus Active 0 = I²C bus is not active 1 = I²C bus is active: a START bit is seen (and no subsequent STOP) Note: This bit is zeroed on a reset controlled by the "I2C Reset Register", and is not set to 1 until a START condition is seen after reset is de-asserted in the "I2C Reset Register".	R	0
02:03	Reserved	Reserved	R	00

(Continued)

Bits	Name	Description	Type	Reset Value
04	SLV_WAIT	Slave Wait 0 = Slave is not waiting for a STOP or RESTART 1 = Slave is waiting for a STOP or RESTART This bit is clear if the bus is not active or the slave address is being received or the slave is active. This bit is set if the bus is active but the slave is not active and the slave address is not being received. Note: This bit is zeroed on a reset controlled by the "I2C Reset Register".	R	0
05:06	SLV_PHASE	Slave Phase 00 = Slave address being received (even if slave interface is disabled using SLV_EN). 01 = Peripheral address being received 10 = Data incoming (write from external master) 11 = Data outgoing (read by external master) At the end of a slave operation, this field will hold its value until the next START/RESTART. If a slave operation aborts, this field will qualify where in the transaction the error occurred.	R	0x0
07	SLV_AN	Slave Ack/Nack 0 = Slave transaction is not in the ACK/NACK bit of a byte 1 = Slave transaction is in the ACK/NACK bit of a byte This qualifies the SLV_PHASE field.	R	0
08:15	SLV_PA	Slave Peripheral Address This field indicates the current peripheral address that is used when the Tsi620 is accessed by an external master.	R	0x00
16	MST_ACTIVE	Master Active 0 = No master operation in progress 1 = Master operation is in progress This status is the same as the START bit in the "I2C Master Control Register". Note: This bit is zeroed on a reset controlled by the "I2C Reset Register".	R	0
17:19	Reserved	Reserved	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
20:22	MST_ PHASE	Master Phase 000 = START condition being sent 001 = External slave address being transmitted 010 = Peripheral address being transmitted 011 = RESTART condition being sent 100 = Data incoming (read operation) 101 = Data outgoing (write operation) 110 = STOP condition being sent 111 = Reserved At the end of a master operation, this field will hold its value until the next master operation is started. If a master operation aborts, this field will qualify where in the transaction the error occurred.	R	000
23	MST_AN	Master Ack/Nack 0 = Master transaction is not in the ACK/NACK bit of a byte 1 = Master transaction is in the ACK/NACK bit of a byte This qualifies the MST_PHASE field.	R	0
24:27	Reserved	Reserved	R	0x0
28:31	MST_ NBYTES	Master Number of Bytes This is the running count of the number of data bytes transferred in the current master operation (read or write). At the end of an operation, if successfully completed, the field will equal the SIZE field from the "I2C Master Control Register". If an operation aborts prematurely, this field will indicate the number of bytes transferred before the error occurred.	R	0x0

29.2.8 I²C Interrupt Status Register

This register indicates the status of the I²C interrupts. When an interrupt status bit is set, an interrupt is generated to the Interrupt Controller if the corresponding bit is enabled in the "I2C Interrupt Enable Register". If the corresponding enable is not set, the interrupt status bit will still assert but will not result in assertion of an interrupt to the Interrupt Controller. This register can only be accessed through the register bus.

Note: This register is affected by a reset controlled by the "I2C Reset Register". All interrupts will be cleared and no interrupt will assert until an event occurs after SRESET bit in the "I2C Reset Register" is de-asserted.

Register name: I2C_INT_STAT **Reset value: 0x0000_0000**	**Register offset: 0x11C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved						OMB_ EMPTY	IMB_FULL
08:15	Reserved						BL_FAIL	BL_OK
16:23	Reserved				SA_ FAIL	SA_ WRITE	SA_READ	SA_OK
24:31	MA_DIAG	Reserved		MA_COL	MA_TMO	MA_NACK	MA_ATMO	MA_OK

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:5	Reserved	Reserved	R	0x00
6	OMB_EMPTY	Outgoing Mailbox Empty 0 = Interrupt status not asserted 1 = Outgoing mailbox is empty Set when an external I²C master reads data from the mailbox (see "Externally Visible I2C Outgoing Mailbox Register"), if data had been previously been written to the mailbox by software.	R/W1C	0
7	IMB_FULL	Incoming Mailbox Full 0 = Interrupt status not asserted 1 = Incoming mailbox is full Set when an external I²C master writes data into the "Externally Visible I2C Incoming Mailbox Register".	R/W1C	0
8:13	Reserved	Reserved	R	0x00
14	BL_FAIL	Boot Load Failed 0 = Interrupt status not asserted 1 = Boot load sequence failed to complete	R/W1C	0

(Continued)

Bits	Name	Description	Type	Reset Value
15	BL_OK	Boot Load OK 0 = Interrupt status not asserted 1 = Boot load sequence completed successfully This will also be set if the boot loading was disabled at reset.	R/W1C	0
16:19	Reserved	Reserved	R	0x0
20	SA_FAIL	Slave Access Failed 0 = Interrupt status not asserted 1 = Error detected during slave access transaction A slave transaction addressed to the Tsi620 aborted.	R/W1C	0
21	SA_WRITE	Slave Write 0 = Interrupt status not asserted 1 = Internal register write performed The "Externally Visible I2C Internal Write Data Register" was written by an external master, invoking a write to an internal register. This will not assert if slave writes are disabled (WR_EN in the "I2C Slave Configuration Register" = 0).	R/W1C	0
22	SA_READ	Slave Read 0 = Interrupt status not asserted 1 = Internal register read performed The "Externally Visible I2C Internal Read Data Register" was read by an external master, invoking a read to an internal register. This will not assert if slave reads are disabled (RD_EN = 0 in the "I2C Slave Configuration Register").	R/W1C	0
23	SA_OK	Slave Access OK 0 = Interrupt status not asserted 1 = Access completed successfully The Tsi620 was addressed as a slave device and the transaction completed without error.	R/W1C	0
24	MA_DIAG	Master Diagnostic Event 0 = Interrupt status not asserted 1 = Diagnostic event	R/W1C	0
25:26	Reserved	Reserved	R	00

(Continued)

Bits	Name	Description	Type	Reset Value
27	MA_COL	Master Collision 0 = Interrupt status not asserted 1 = Collision (arbitration loss) occurred following the device address phase A transaction initiated using the "I2C Master Control Register" aborted due to loss of arbitration after the slave device address phase. This indicates multiple masters tried to access the same slave. This can also be set at the end of boot load due to BL_FAIL.	R/W1C	0
28	MA_TMO	Master Timeout 0 = Interrupt status not asserted 1 = Transaction aborted due to timeout expiration A transaction initiated using the "I2C Master Control Register" aborted due to expiration of the clock low, byte, or transaction time-outs. This can also be set at the end of boot load due to BL_FAIL.	R/W1C	0
29	MA_NACK	Master NACK 0 = Interrupt status not asserted 1 = NACK received during transaction A transaction initiated through the "I2C Master Control Register" aborted due to receipt of a NACK in response to slave address, peripheral address, or a written byte. This can also be set at the end of boot load due to BL_FAIL.	R/W1C	0
30	MA_ATMO	Master Arbitration Timeout 0 = Interrupt status not asserted 1 = Bus arbitration timeout expired A transaction initiated through the "I2C Master Control Register" aborted due to expiration of the arbitration timeout (see ARB_TO in "I2C_SCLK Low and Arbitration Timeout Register"). This indicates the bus is in use by other masters.	R/W1C	0
31	MA_OK	Master Transaction OK 0 = Interrupt status not asserted 1 = Access completed and successful A transaction initiated through the "I2C Master Control Register" completed without error.	R/W1C	0

The write-1-to-clear (W1C) operation requires that this register first be read to create an event snapshot.

29.2.9 I2C Interrupt Enable Register

This register controls which of the interrupt status bits in the "I2C Interrupt Status Register" will result in an interrupt asserted to the Interrupt Controller. It can only be accessed from the register bus.

Register name: I2C_INT_ENABLE Reset value: 0x0000_0000	Register offset: 0x120

Bits	0	1	2	3	4	5	6	7
00:07	Reserved						OMB_ EMPTY	IMB_FULL
08:15	Reserved						BL_FAIL	BL_OK
16:23	Reserved				SA_ FAIL	SA_ WRITE	SA_READ	SA_OK
24:31	MA_DIAG	Reserved		MA_COL	MA_TMO	MA_NACK	MA_ATMO	MA_OK

Bits	Name	Description	Type	Reset Value
0:5	Reserved	Reserved	R	0x00
6	OMB_ EMPTY	Enable OMB_EMPTY Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0
7	IMB_FULL	Enable IMB_FULL Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0
8:13	Reserved	Reserved	R	0x00
14	BL_FAIL	Enable BL_FAIL Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0
15	BL_OK	Enable BL_OK Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0
16:19	Reserved	Reserved	R	0x0
20	SA_FAIL	Enable SA_FAIL Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
21	SA_WRITE	Enable SA_WRITE Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0
22	SA_READ	Enable SA_READ Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0
23	SA_OK	Enable SA_OK Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0
24	MA_DIAG	Enable MA_DIAG Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0
25:26	Reserved	Reserved	R	00
27	MA_COL	Enable MA_COL Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0
28	MA_TMO	Enable MA_TMO Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0
29	MA_NACK	Enable MA_NACK Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0
30	MA_ATMO	Enable MA_ATMO Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0
31	MA_OK	Enable MA_OK Interrupt 0 = Interrupt is disabled 1 = Interrupt is enabled	R/W	0

29.2.10 I2C Interrupt Set Register

This register sets the status of the I2C blocks interrupts. It can only be accessed from the register bus.

Note: Setting an interrupt sets all related underlying events in the "I2C New Event Register". This is significant in that if all underlying events are disabled for a specific interrupt bit, this register will not appear to work for that bit.

Register name: I2C_INT_SET **Reset value: 0x0000_0000**	**Register offset: 0x124**

Bits	0	1	2	3	4	5	6	7
00:07	Reserved						OMB_ EMPTY	IMB_ FULL
08:15	Reserved						BL_FAIL	BL_OK
16:23	Reserved				SA_FAIL	SA_ WRITE	SA_READ	SA_OK
24:31	MA_DIAG	Reserved		MA_COL	MA_TMO	MA_NACK	MA_ATMO	MA_OK

Bits	Name	Description	Type	Reset Value
0:5	Reserved	Reserved	R	0x00
6	OMB_EMPTY	Set OMB_EMPTY Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0
7	IMB_FULL	Set IMB_FULL Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0
8:13	Reserved	Reserved	R	0x00
14	BL_FAIL	Set BL_FAIL Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0
15	BL_OK	Set BL_OK Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0
16:19	Reserved	Reserved	R	0x0
20	SA_FAIL	Set SA_FAIL Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0

(Continued)

Bits	Name	Description	Type	Reset Value
21	SA_WRITE	Set SA_WRITE Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0
22	SA_READ	Set SA_READ Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0
23	SA_OK	Set SA_OK Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0
24	MA_DIAG	Set MA_DIAG Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0
25:26	Reserved	Reserved	R	00
27	MA_COL	Set MA_COL Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0
28	MA_TMO	Set MA_TMO Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0
29	MA_NACK	Set MA_NACK Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0
30	MA_ATMO	Set MA_ATMO Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0
31	MA_OK	Set MA_OK Interrupt 0 = No effect 1 = Interrupt is set	R/W1S	0

29.2.11 I²C Slave Configuration Register

This register configures the slave interface portion of the I²C block. The slave interface is the logic that responds to transactions from an external master on the I²C bus

Register name: I2C_SLV_CFG Reset value: Undefined	Register offset: 0x12C

Bits	0	1	2	3	4	5	6	7
00:07	RD_EN	WR_EN	ALRT_EN	SLV_EN	Reserved			SLV_UNLK
08:15	Reserved							
16:23	Reserved							
24:31	Reserved	SLV_ADDR						

Bits	Name	Description	Type	Reset Value
0	RD_EN	Register Bus Read Enable This bit controls whether external masters can read registers internal to the Tsi620. The SLV_EN bit must also be set for this option to have any effect. 0 = Transactions that read the "Externally Visible I2C Internal Read Data Register" will not invoke reads of the internal registers. 1 = Transactions that read the "Externally Visible I2C Internal Read Data Register" will trigger reads of the internal register whose address is in the "Externally Visible I2C Internal Read Address Register".	R/W	1
1	WR_EN	Register Bus Write Enable This bit controls whether external masters can write to Tsi620's internal registers. The SLV_EN bit must also be set for this option to have any effect.0 = Transactions that write the "Externally Visible I2C Internal Write Data Register" will not invoke writes of the internal registers. 1 = Transactions that write the "Externally Visible I2C Internal Write Data Register" will trigger writes of the internal register whose address is in the "Externally Visible I2C Internal Write Address Register".	R/W	1

(Continued)

Bits	Name	Description	Type	Reset Value
2	ALRT_EN	Alert Address Enable 0 = Do not respond to read of the Alert Response Address of 0001100 1 = Respond to read of the Alert Response Address of 0001100 if any bits are set in the "Externally Visible I2C Status Register". If enabled, the slave interface will respond to the Alert Response Address for a read transaction if the ALERT_FLAG is set in the "Externally Visible I2C Slave Access Status Register". The response is to return the SLV_ADDR field followed by a 0 to the external master, then clear the ALERT_FLAG. If ALRT_EN is 0, then the Alert Response address can be used as a SLV_ADDR. If ALRT_EN is 1 and SLV_ADDR is also set to the alert response address, then the alert response behavior will take precedence.	R/W	0
3	SLV_EN	Slave Enable 0 = Slave is not enabled; SLV_ADDR is ignored. 1 = Slave interface is enabled; SLV_ADDR is responded to when transaction started by external master. When enabled, the slave interface will acknowledge transactions to the SLV_ADDR from an external master. If not enabled, then all transactions are NACK'd, except the Alert Response Address read, if ALRT_EN is 1. This bit controls access to the peripheral address space of the Tsi620. Access to the internal register space is also controlled by the RD_EN and WR_EN bits. If SLV_EN is 0 then internal register access is also disabled.	R/W	Undefined
4:6	Reserved	Reserved	R	000
7	SLV_UNLK	Slave Address Unlock	R/W	0
8:24	Reserved	Reserved	R	0x0000
25:31	SLV_ADDR	Slave Address This is the device address for the Tsi620 as an I^2C slave. An external master uses this address to access the Tsi620 peripheral register space. For the slave interface to respond to this address, the SLV_EN bit must be set. A SLV_ADDR of 0x00 is never valid, as that value is used for the I2C START_BYTE and General Call functions. Functions that are not supported and are ignored and NACK'd.	R/W	Undefined

29.2.12 I2C Boot Control Register

This register controls the boot load sequence that is initiated following a chip reset of the Tsi620. The initial boot load operation is controlled by the reset state of this register. Some of the fields are also latched from device pins at power-up.

Once boot loading is in progress, the data read from the EEPROM can modify the contents of this register and redirect the loading to another EEPROM, or to another address within the same EEPROM. This process is called “chaining.” The progress of a boot load operation can be monitored using the “I2C Boot Load Diagnostic Progress Register” and “I2C Boot Load Diagnostic Configuration Register”.

This register can be read and written after boot loading is complete, but has no further effect on block operation.

Register name: I2C_BOOT_CNTRL **Reset value: Undefined**	**Register offset: 0x140**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	CHAIN	PSIZE	BINC	BUNLK	Reserved			
08:15	Reserved	BOOT_ADDR						
16:23	PAGE_MODE			PADDR				
24:31	PADDR							

Bits	**Name**	**Description**	**Type**	**Reset Value**
00	CHAIN	Chain During Boot 0 = No chain 1 = Chain to new device This bit is set to invoke a chain operation during boot load. In order to chain, this bit must be set and the register load count must be at zero; that is, the write to this register must be the last one in the boot sequence within this EEPROM if chaining were not continuing the boot. Except for the BUNLK and PAGE_MODE fields, modifications to the remaining fields in this register have no effect unless this bit is set. The fields will change value, but they will not affect the boot load sequence. Once boot load is complete, this register has no further effect on block operation.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
01	PSIZE	Peripheral Address Size 0 = Use 1 byte for peripheral address 1 = Use 2 bytes for peripheral address This selects the number of bytes in the peripheral address. If 0 then only the least significant 5 bits of PADDR are used (+ 3 LSBs of 000). If 1 then all 13 bits of PADDR are used (+ 3 LSBs of 000). For 2-byte addressing, the MSB of the address is transmitted first on the I2C bus (see the PADDR field for an example). This field can be changed during the boot load, in conjunction with setting the CHAIN bit, in order to jump the boot load to a new boot device with different address size.	R/W	Undefined
02	BINC	Boot Address Increment 0 = Do not increment boot address when peripheral address overflows 1 = Increment the least significant 3 bits of the internal boot address when peripheral address overflows, then re-address device This option is valid only when PSIZE is 0, and is used to access devices that use the least significant 3 bits of their device address as a 256-byte page select (typically 2K EEPROMs). When enabled, and the 1-byte peripheral address wraps back to zero, the least significant 3 bits of the device address is incremented, followed by a Restart and a new device address cycle. The device address starts as the value of the BOOT_ADDR field, and is copied internally at boot start or upon a chain operation. It is the internal value that is incremented to simulate addressing a 2K EEPROM. This field can be changed during the boot load, in conjunction with setting the CHAIN bit, in order to jump the boot load to a new boot device with new page properties.	R/W	1
03	BUNLK	Boot Address Unlock	R/W	0
04:08	Reserved	Reserved	R	0x00
09:15	BOOT_ADDR	Boot Device Address This field can be changed during the boot load in conjunction with setting the CHAIN bit, in order to jump the boot load to a new boot device. This starting address is copied internally at boot start or boot load, and it is the internal value that is incremented, as explained in the BINC field.	R/W	Undefined

(Continued)

Bits	Name	Description	Type	Reset Value
16:18	PAGE_MODE	Page Mode 000 = 8 bytes 001 = 32 bytes 010 = 64 bytes 011 = 128 bytes 100 = 256 bytes 101 = 512 bytes 110 = 1024 bytes 111 = Infinite This field modifies the boot load process to adjust the boundary at which the boot device is re-addressed. In the default case, the boot load sequence reads 8 bytes, then does a Restart followed by the device and peripheral address phases. By changing this field, the device will be re-addressed only when the peripheral address crosses the indicated boundary, thus saving a considerable number of clock cycles during the boot. It is up to the programmer of the EEPROM to ensure that the addressed device can support consecutive reads up to the selected boundary. Some devices wrap at certain page boundaries, and this setting must be consistent with such limitations. A setting of 111 causes the entire boot load to be read sequentially, with two exceptions. No matter what the setting of this field, if the boot address is incremented due to the BINC mode being enabled, or if chaining occurs, then the device is readdressed. Changing this field during boot load will immediately affect the boot sequence.	R/W	000
19:31	PADDR	Peripheral Address This is the most significant 5 or 13 bits of the peripheral address (depending on PSIZE setting). The least significant 3 bits are not programmable and are assumed 000; that is, the peripheral address must be aligned to a multiple of 8 address in the EEPROM. To form the peripheral address, this field is shifted left by 3 and then copied internally upon boot start or a chain operation. The internal address is then incremented as the boot load progresses. For 2-byte addressing, the MSB of the peripheral address is sent first. For example, setting this field to 0x0127 gives a peripheral address of (0x0127 << 3) = 0x0938. The first byte sent to the external device is 0x09 and the second byte is 0x38. This field can be changed during the boot load, in conjunction with setting the CHAIN bit, in order to jump the boot load to a new peripheral address.	R/W	0x0000

29.2.13 Externally Visible I2C Internal Write Address Register

This register contains the internal register address set by an external I2C master to be used for internal register writes when the "Externally Visible I2C Internal Write Data Register" is written. The address is forced to be 4-byte aligned (the 2 lowest bits are read-only).

This register is read-only from the register bus, and R/W from the I2C bus through the slave interface. This register corresponds to the I2C peripheral address 0x00 through 0x03.

Note: This register is also used during the boot load process to accumulate the address read from the EEPROM for each address/data pair. Therefore, at the end of the boot load process, this register will contain the last register address read from the EEPROM, or the first four bytes of the register count.

Register name: EXI2C_REG_WADDR **Reset value: 0x0000_0000**	**Register offset: 0x200**

Bits	0	1	2	3	4	5	6	7
00:07	ADDR							
08:15	ADDR							
16:23	ADDR							
24:31	ADDR						Reserved	

Bits	Name	Description	Type	Reset Value
0:29	ADDR	Internal Register Write Address Register address to be used when a write to the "Externally Visible I2C Internal Write Data Register" invokes an internal register write. This address is 4-byte aligned. The specific byte accessed is controlled by the peripheral address within the data register. This address auto-increments by 4 if WINC in the "Externally Visible I2C Internal Access Control Register" is set, and the MSB of the data (peripheral address 0x07) is written.	R	0x0000_0000
30:31	Reserved	Reserved	R	00

29.2.14 Externally Visible I2C Internal Write Data Register

This register contains the internal register data last written by an external I2C master through the slave interface. The register is read-only from the register bus, and R/W from the I2C bus through the slave interface.

This register corresponds to the I2C peripheral addresses 0x04 through 0x07.

Note: This register is also used during the boot load process to accumulate the data read from the EEPROM for each address/data pair. Therefore, at the end of the boot load process, this register will contain the last register data read from the EEPROM, or the last four bytes of the register count.

Register name: EXI2C_REG_WDATA **Reset value: 0x0000_0000**	**Register offset: 0x204**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	WDATA							
08:15	WDATA							
16:23	WDATA							
24:31	WDATA							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:31	WDATA	Internal Register Write Data Note: When the MSB of this register is written (peripheral address 0x07), the slave peripheral address wraps to 0x04 (the LSB of this register) instead of incrementing to 0x08. This allows an external master to write a block of internal registers without having to change the slave peripheral address, assuming WINC in the "Externally Visible I2C Internal Access Control Register" is set to auto-increment the WADDR. When 0x07 is read, the peripheral address increments to 0x08.	R	0

29.2.15 Externally Visible I2C Internal Read Address Register

This register contains the internal register address set by an external I2C master to be used for internal register reads when the "Externally Visible I2C Internal Read Data Register" is read. The address is forced to be 4-byte aligned (the 2 lowest bits are read-only).

This register is read-only from the register bus, and R/W from the I2C bus through the slave interface. This register corresponds to the I2C peripheral address 0x10 through 0x13.

Register name: EXI2C_REG_RADDR **Reset value: 0x0000_0000**	**Register offset: 0x210**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	ADDR							
08:15	ADDR							
16:23	ADDR							
24:31	ADDR						Reserved	

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:29	ADDR	Internal Register Read Address Register address to be used when a read to the "Externally Visible I2C Internal Read Data Register" invokes an internal register read. This address is 4-byte aligned. The specific byte accessed is controlled by the peripheral address within the data register. This address auto-increments by 4 if RINC in the "Externally Visible I2C Internal Access Control Register" is set, and the LSB of the data (peripheral address 0x14) is read.	R	0x0000_0000
30:31	Reserved	Reserved	R	00

29.2.16 Externally Visible I2C Internal Read Data Register

This register contains the internal register data last read by an external I2C master through the slave interface. The register is read-only from both the register bus and the I2C bus through the slave interface.

This register corresponds to the I2C peripheral addresses 0x14 through 0x17.

Register name: EXI2C_REG_RDATA Reset value: 0x0000_0000	Register offset: 0x214

Bits	0	1	2	3	4	5	6	7
00:07	RDATA							
08:15	RDATA							
16:23	RDATA							
24:31	RDATA							

Bits	Name	Description	Type	Reset Value
0:31	RDATA	Internal Register Read Data Note: When the MSB of this register is read (peripheral address 0x17), the slave peripheral address wraps to 0x14 (the LSB of this register) instead of incrementing to 0x18. This allows an external master to read a block of internal registers without having to change the slave peripheral address, assuming RINC in the "Externally Visible I2C Internal Access Control Register" is set to auto-increment the RADDR. When 0x17 is written, the peripheral address increments to 0x18.	R	0

29.2.17 Externally Visible I2C Slave Access Status Register

This register provides status indications to an external I2C master. It is read-only from both the register bus and the I2C bus through the slave interface. This register corresponds to the I2C peripheral addresses 0x20 through 0x23.

Note: This register is affected by a reset controlled by the "I2C Reset Register". All status will be cleared.

Register name: EXI2C_ACC_STAT **Reset value: 0x0000_0000**	**Register offset: 0x220**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							
24:31	ACC_OK	Reserved			OMB_ FLAG	IMB_ FLAG	Reserved	ALERT_ FLAG

Bits	**Name**	**Description**	**Type**	**Reset Value**
00:23	Reserved	Reserved	R	0x00_0000
24	ACC_OK	Internal Register Access OK 0 = No access, or access in progress 1 = Access was successful This bit is set when a slave access successfully reads or writes data to an internal register through the "Externally Visible I2C Internal Write Data Register" or "Externally Visible I2C Internal Read Data Register". Reading this bit returns the last status of the bit. If read through the slave interface (through peripheral address 0x20), the bit is then cleared to 0. The bit is not cleared to 0 when read by a host or indirectly through the EXI2C_REG_RADDR / EXI2C_REG_RDATA function.	R	0
25:27	Reserved	Reserved	R	000

(Continued)

Bits	Name	Description	Type	Reset Value
28	OMB_FLAG	Outgoing Mailbox Flag 0 = Outgoing mailbox empty 1 = New data in the outgoing mailbox This bit is set when data is written to the outgoing mailbox register (“Externally Visible I2C Outgoing Mailbox Register”) by software. This bit remains set (flag up) until an external I^2C master reads the outgoing mailbox register, and the bit is then cleared (flag down). When the mailbox is read, the OMB_EMPTY interrupt is asserted. A mailbox read is considered complete when the external master issues a STOP condition to end the transaction during which any bytes in the mailbox were written.	R	0
29	IMB_FLAG	Incoming Mailbox Flag 0 = Incoming mailbox empty 1 = New data in the incoming mailbox This bit is set when data is written to the incoming mailbox register (“Externally Visible I2C Incoming Mailbox Register”) by an external I^2C master. This bit remains set (flag up) until software reads the incoming mailbox register, and the bit is then cleared (flag down). When the mailbox is written and the flag is set, the IMB_FULL interrupt is asserted. A mailbox read is considered complete when the external master issues a STOP condition to end the transaction during which any bytes in the mailbox were written.	R	0
30	Reserved	Reserved	R	0
31	ALERT_FLAG	Alert Response Flag 0 = No alert 1 = Alert response active This bit is set when the Alert Response would trigger, as defined in the “Externally Visible I2C Status Register”. It is cleared by a successful response to the Alert Response Address or if the global status no longer requires the alert to be asserted. On a reset, this flag will assert immediately due to EXI2C_STAT[RESET] asserting.	R	0

29.2.18 Externally Visible I2C Internal Access Control Register

This register allows an external I2C master to configure the functionality for internal register accesses through the slave interface. This register is read-only from the register bus and R/W from the I2C bus through the slave interface.

The fields in this register control the size and auto-increment functions when internal register accesses are performed by an external master through the slave interface.

This register corresponds to the I2C peripheral addresses 0x24 through 0x27.

Register name: EXI2C_ACC_CNTRL **Reset value: 0x0000_00A0**	**Register offset: 0x224**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							
24:31	RSIZE		WSIZE		RINC	WINC	Reserved	

Bits	**Name**	**Description**	**Type**	**Reset Value**
00:23	Reserved	Reserved	R	0x00_0000
24:25	RSIZE	Internal Register Read Access Size 00 = 1 byte 01 = 2 bytes (Reserved) 10 = 4 bytes – An internal register read is invoked once for each internal register, loading all 4 bytes in the "Externally Visible I2C Internal Read Data Register". The read is performed when the LSB of the data register is read (peripheral address 0x14). 11 = 8 bytes (Reserved) All Reserved settings will result in internal read accesses being disabled.	R	10

(Continued)

Bits	Name	Description	Type	Reset Value
26:27	WSIZE	Internal Register Write Access Size 00 = 1 byte 01 = 2 bytes (Reserved) 10 = 4 bytes – An internal register write is invoked once for each internal register, writing all 4 bytes from the "Externally Visible I2C Internal Write Data Register". The write is performed when the MSB of the data register is written (peripheral address 0x07). 11 = 8 bytes (Reserved) All Reserved settings will result in internal writes accesses being disabled.	R	10
28	RINC	Enable Auto-Incrementing on Internal Register Reads 0 = The "Externally Visible I2C Internal Read Address Register" is unchanged after reads performed to the "Externally Visible I2C Internal Read Data Register" 1 = The "Externally Visible I2C Internal Read Address Register" is incremented by 4 after reads performed to the LSB of the "Externally Visible I2C Internal Read Data Register" (peripheral address 0x14) so that the address points to the next internal register address. When auto-incrementing is on, consecutive internal registers can be read in one I^2C transaction without the need to reset the peripheral address because the peripheral address wraps from 0x17 back to 0x14. If auto-incrementing is off, then the same internal register can be read multiple times in a single I^2C transaction. The latter could be useful for polling a status register.	R	0
29	WINC	Enable Auto-Incrementing on Internal Register Writes 0 = The "Externally Visible I2C Internal Write Address Register" is unchanged after reads performed to the "Externally Visible I2C Internal Write Data Register". 1 = The "Externally Visible I2C Internal Write Address Register" is incremented by 4 after writes performed to the MSB of the "Externally Visible I2C Internal Write Data Register" (peripheral address 0x07) so that the address points to the next internal register address. When auto-incrementing is on, consecutive internal registers can be written in one I^2C transaction with the need to reset the peripheral address because the peripheral address wraps from 0x07 back to 0x04. If auto-incrementing is off, then the same internal register can be written multiple times in a single I^2C transaction.	R	0
30:31	Reserved	Reserved	R	00

29.2.19 Externally Visible I2C Status Register

This register provides a summary view of status of the Tsi620. It can be polled by an external system management device. Any bit masked by its related enable, changing from 0 to 1, will cause ALERT_FLAG to be set in the "Externally Visible I2C Slave Access Status Register", and the Tsi620 to respond to the Alert Response Address if the ALRT_EN bit is set in the "I2C Slave Configuration Register". The related enables are present in the "Externally Visible I2C Enable Register". If all masked status bits are 0, then the ALERT_FLAG clears. The ALERT_FLAG also clears when the slave responds to the Alert Response Address, and not set again until there is a change in the status.

Bits [0] are read only from the register bus, but R/W1C from the I^2C bus through the slave interface. They are set when the corresponding event occurs within the Tsi620, and held asserted until an external I^2C master writes a 1 to that position to clear the event. If an event is still asserting at the time the W1C occurs, the bit remains set.

The software status bits [1:3] are R/W from the register bus. They can be set or cleared by software, and thereby used for any system purpose. An external I^2C master can write 1 to those bits to clear them. If the W1C occurs at the same time as software is writing the bit, the software written value will take precedence.

This register corresponds to the I^2C peripheral addresses 0x80 through 0x83. This register is affected by a reset controlled by the "I2C Reset Register". All status will be cleared, including the software status bits. Chip status will re-assert after a SRESET is released in the "I2C Reset Register" only if that chip event occurs again.

Register name: EXI2C_STAT **Reset value: 0x8000_0000**	**Register offset: 0x280**

Bits	0	1	2	3	4	5	6	7
00:07	RESET	SW_ STAT2	SW_ STAT1	SW_ STAT0	OMBW	IMBR	I2C	TEA
08:15	RCS	MCS	PCI_INT	BISF_INT	SREP_INT	Reserved	MC_LAT	MCE
16:23	GPIO1	GPIO0	CLK_GEN	Reserved				PORT8
24:31	PORT7	PORT6	PORT5	PORT4	PORT3	PORT2	PORT1	PORT0

Bits	Name	Description	Type	Reset Value
0	RESET	Reset Status 0 = No reset has occurred since the last time this bit was cleared. 1 = A reset has occurred since the last time this bit was cleared. This indicates the reset of the Tsi620. This bit is set to 1 on a Block or Chip reset, and cleared to 0 on a when the I^2C block is reset using the SRESET bit of the "I2C Reset Register".	R	1

(Continued)

Bits	Name	Description	Type	Reset Value
1	SW_STAT2	Software Status Bit 2 0 = Status value 0 1 = Status value 1 This bit can be set or cleared by software using a register write for any system specific purpose.	R/W	0
2	SW_STAT1	Software Status Bit 1 0 = Status value 0 1 = Status value 1 This bit can be set or cleared by software using a register write for any system specific purpose.	R/W	0
3	SW_STAT0	Software Status Bit 0 0 = Status value 0 1 = Status value 1 This bit can be set or cleared by software using a register write for any system specific purpose.	R/W	0
4	OMBW	Outgoing Mailbox Written 0 = Outgoing mailbox not filled since last clear 1 = Outgoing mailbox is filled This bit asserted indicates that software has written to the outgoing mailbox since this bit was last cleared.	R	0
5	IMBR	Incoming Mailbox Read 0 = Incoming mailbox not read since last clear 1 = Incoming mailbox is emptied This bit asserted indicates that software has read the incoming mailbox, when the mailbox was full, since this bit was last cleared.	R	0
6	I2C	I^2C Interrupt 0 = I^2C is not asserting an interrupt to processor 1 = I^2C is asserting an interrupt to the processor	R	0
7	TEA	Switch ISF Transaction Error Acknowledge (TEA) 0 = No TEA 1 = TEA asserted by one or more ports TEA occurred in the Switch ISF (to determine which port(s) incurred the TEA, see the “Switch ISF Interrupt Status Register”).	R	0
8	RCS	Reset Control Symbol Status 0 = No status asserted 1 = Status asserted Combined Reset Control Symbol interrupt status from all RapidIO ports. Note that this does not include the SREP.	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
9	MCS	Multicast Event Control Symbol Status 0 = No status asserted 1 = Status asserted Combined multicast event control system interrupt status from all ports, excluding the SREP.	R	0
10	PCI_INT	Interrupt from the PCI Interface 0 = No interrupt 1 = PCI has asserted an interrupt to the processor PCI_INT is set if any of the following bits in the "Block Event Status Register" are set: • PCI_ERR • PCI_RESET_RX	R	0
11	BISF_INT	Interrupt from the Bridge ISF 0 = No interrupt 1 = Bridge ISF has asserted an interrupt to the processor	R	0
12	SREP_INT	Interrupt from the SREP 0 = No interrupt 1 = SREP has asserted an interrupt to the processor SREP_INT is set if any of the following bits in the "Block Event Status Register" are set: • SREP_RESET_RX • SREP_MCS_RX • SREP_PW_RX • SREP_DB_RX • SREP_ERR	R	0
13	Reserved	N/A	R	0
14	MC_LAT	Multicast Latency Timeout 0 = No timeout 1 = Multicast timeout on one or more ports Indicates that a multicast request for a specific port could not be sent in the required time. To determine which port(s) have had their multicast maximum latency timeout period expire, see "Switch ISF Broadcast Buffer Maximum Latency Expired Error Register".	R	0
15	MCE	Multicast Work Queue Dropped Packet Interrupt 0 = No interrupt 1 = Multicast logic dropped a packet	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
16	GPIO1	General Purpose I/O Interrupt Input 1 At least one of the GPIO pins 16-31 have an interrupt asserted. 0 = No interrupt 1 = Interrupt asserted	R	0
17	GPIO0	General Purpose I/O Interrupt Input 0 At least one of the GPIO pins 0-15 have an interrupt asserted. 0 = No interrupt 1 = Interrupt asserted	R	0
18	CLK_GEN	Clock Generator Interrupt The clock generator PLLs have slipped. 0 = No interrupt 1 = Interrupt asserted	R	0
19:22	Reserved	Reserved	R	000
23	PORT8	Port 8 Interrupt 0 = No interrupt 1 = Port 8 has asserted an interrupt to the processor	R	0
24	PORT7	Port 7 Interrupt 0 = No interrupt 1 = Port 7 has asserted an interrupt to the processor Note: This bit is never asserted in the Tsi620.	R	0
25	PORT6	Port 6 Interrupt 0 = No interrupt 1 = Port 6 has asserted an interrupt to the processor	R	0
26	PORT5	Port 5 Interrupt 0 = No interrupt 1 = Port 5 has asserted an interrupt to the processor	R	0
27	PORT4	Port 4 Interrupt 0 = No interrupt 1 = Port 4 has asserted an interrupt to the processor	R	0
28	PORT3	Port 3 Interrupt 0 = No interrupt 1 = Port 3 has asserted an interrupt to the processor	R	0
29	PORT2	Port 2 Interrupt 0 = No interrupt 1 = Port 2 has asserted an interrupt to the processor	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
30	PORT1	Port 1 Interrupt 0 = No interrupt 1 = Port 1 has asserted an interrupt to the processor	R	0
31	PORT0	Port 0 Interrupt 0 = No interrupt 1 = Port 0 has asserted an interrupt to the processor	R	0

29.2.20 Externally Visible I2C Enable Register

Any bit set in this register will enable the equivalent bit in the "Externally Visible I2C Status Register" to set the ALERT_FLAG. These enables do not affect whether events are set in the global status register, only whether the asserted events are allowed to set the ALERT_FLAG when changing from 0 to 1. If an event is already asserted in the status when the related enable is changed from 0 to 1, this is equivalent to the event asserting, and the ALERT_FLAG will be set.

This register is R/W from either the register bus or from the I2C bus through the slave interface. If the register is written by both at the same time, the register bus interface will take precedence.

This register corresponds to the I2C peripheral addresses 0x84 through 0x87.

Register name: EXI2C_STAT_ENABLE **Reset value: 0xFFFF_FFFF**	**Register offset: 0x284**

Bits	0	1	2	3	4	5	6	7
00:07	RESET	SW_ STAT2	SW_ STAT1	SW_ STAT0	OMBW	IMBR	I2C	TEA
08:15	RCS	MCS	PCI_INT	BISF_INT	SREP_INT	Reserved	MC_LAT	MCE
16:23	GPIO1	GPIO0	CLK_GEN	Reserved				PORT8
24:31	PORT7	PORT6	PORT5	PORT4	PORT3	PORT2	PORT1	PORT0

Bits	Name	Description	Type	Reset Value
0	RESET	Enable RESET Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
1	SW_STAT2	Enable SW_STAT2 Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
2	SW_STAT1	Enable SW_STAT1 Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
3	SW_STAT0	Enable SW_STAT0 Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
4	OMBW	Enable Outgoing Mailbox Written 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1

(Continued)

Bits	Name	Description	Type	Reset Value
5	IMBR	Enable Incoming Mailbox Read 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
6	I2C	Enable I2C Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
7	TEA	Enable TEA Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
8	RCS	Enable RCS Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
9	MCS	Enable MCS Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
10	PCI_INT	Enable PCI Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
11	BISF_INT	Enable Bridge ISF Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
12	SREP_INT	Enable SREP Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
13	Reserved	N/A	R/W	1
14	MC_LAT	Enable MC_LAT Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
15	MCE	Enable MCE Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
16	GPIO1	Enable GPIO1 Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1

(Continued)

Bits	Name	Description	Type	Reset Value
17	GPIO0	Enable GPIO0 Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
18	CLK_GEN	Enable CLK_GEN Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
19:22	Reserved	Reserved These bits are unused in the Tsi620. The enables can be changed, but have no effect.	R	0xF
23	PORT8	Enable PORT8 Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
24	PORT7	Enable PORT7 Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
25	PORT6	Enable PORT6 Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
26	PORT5	Enable PORT5 Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
27	PORT4	Enable PORT4 Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
28	PORT3	Enable PORT3 Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
29	PORT2	Enable PORT2 Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
30	PORT1	Enable PORT1 Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1
31	PORT0	Enable PORT0 Alert Response 0 = Status asserted will not enable setting ALERT_FLAG 1 = Status asserted will enable setting ALERT_FLAG	R/W	1

29.2.21 Externally Visible I2C Outgoing Mailbox Register

This register is the outgoing mailbox, allowing the processor to communicate data to an external I2C master. The register is R/W from the register bus, and read-only from the I2C bus through the slave interface.

This register corresponds to the I2C peripheral addresses 0x90 through 0x93.

Register name: EXI2C_MBOX_OUT Reset value: 0x0000_0000	Register offset: 0x290

Bits	0	1	2	3	4	5	6	7
00:07	DATA							
08:15	DATA							
16:23	DATA							
24:31	DATA							

Bits	Name	Description	Type	Reset Value
0:31	DATA	Mailbox data to be transferred to an external I2C master. Every write to this register by software sets the OMB_FLAG bit in the "Externally Visible I2C Slave Access Status Register", indicating data is available in the outgoing mailbox. When this register is read by an external master, the OMB_FLAG bit is cleared, and an OMB_EMPTY interrupt is asserted if the OMB_FLAG bit was set. This register is read-only through the I2C slave interface. **Note**: A read is considered complete when the STOP condition is seen on the I2C bus, and one or more bytes in this register were read by the external master since the preceding START condition.	R/W	0

29.2.22 Externally Visible I2C Incoming Mailbox Register

This register is the incoming mailbox, allowing an external I2C master to communicate data to the host processor. The register is read-only from the register bus, and R/W from the I2C bus through the slave interface.

This register corresponds to the I2C peripheral addresses 0x94 through 0x97.

Register name: EXI2C_MBOX_IN Reset value: 0x0000_0000	Register offset: 0x294

Bits	0	1	2	3	4	5	6	7
00:07	DATA							
08:15	DATA							
16:23	DATA							
24:31	DATA							

Bits	Name	Description	Type	Reset value
0:31	DATA	Mailbox data transferred from an external I2C master. Every write to this register sets the IMB_FLAG in the "Externally Visible I2C Slave Access Status Register", and asserts an IMB_FULL interrupt. When software reads this register, the IMB_FLAG is cleared.	R	0

29.2.23 I2C Event and Event Snapshot Registers

These registers indicate events that occur within the I2C block. For the I2C_EVENT register, each bit is an "or" of the corresponding bit in the "I2C New Event Register" and the I2C_SNAP_EVENT register. The I2C_SNAP_EVENT register contains those events that were asserted prior to the last snapshot. A snapshot is taken when the "I2C Interrupt Status Register" is read.

Each bit in these registers are write-one-to-clear. Writing a 1 to a bit position in the I2C_EVENT register will clear the event in both the snapshot and new event registers. Writing a 1 to a bit position in the I2C_SNAP_EVENT register will clear the bit only in that register. Writing a 1 to a bit position in the I2C_INT_STAT register will clear all related event bits in the I2C_SNAP_EVENT register, provided those events are enabled in the I2C_EVENT_ENB register. Bits from the I2C_EVENT register are masked (enabled) by the corresponding bits in the "I2C Enable Event Register", and then determine whether a related bit in the I2C_INT_STAT register is set.

Note: These registers are affected by a reset controlled by the "I2C Reset Register". All events will be cleared and will not assert while SRESET is asserted in the "I2C Reset Register".

Register name: I2C_{EVENT, SNAP_EVENT} **Reset value: 0x0000_0000**	**Register offset: 0x300, 304**

Bits	0	1	2	3	4	5	6	7
00:07	Reserved	SDW	SDR	SD	Reserved	DTIMER	DHIST	DCMDD
08:15	IMBW	OMBR	Reserved	SCOL	STRTO	SBTTO	SSCLTO	Reserved
16:23	Reserved	MTD	Reserved	BLTO	BLERR	BLSZ	BLNOD	BLOK
24:31	Reserved		MNACK	MCOL	MTRTO	MBTTO	MSCLTO	MARBTO

Bits	Name	Description	Type	Reset Value
00	Reserved	Reserved	R	0
01	SDW	Slave Internal Register Write Done Event 0 = Event not asserted 1 = Slave interface completed a transaction for an external master that resulted in a write to an internal register	R/W1C	0
02	SDR	Slave Internal Register Read Done Event 0 = Event not asserted 1 = Slave interface completed a transaction for an external master that resulted in a read to an internal register	R/W1C	0
03	SD	Slave Transaction Done Event 0 = Event not asserted 1 = Slave interface completed an I2C transaction for an external master with no detectable error	R/W1C	0

(Continued)

Bits	Name	Description	Type	Reset Value
04	Reserved	Reserved	R	0
05	DTIMER	Diagnostic Timer Expired Event 0 = Event not asserted 1 = Diagnostic timer has expired This event does not assert during the boot load sequence. The BLTO will assert instead.	R/W1C	0
06	DHIST	Diagnostic History Filling Event 0 = Event not asserted 1 = Diagnostic history recorded the 8th event	R/W1C	0
07	DCMDD	Diagnostic Command Done Event 0 = Event not asserted 1 = Master interface completed the diagnostic command	R/W1C	0
08	IMBW	Incoming Mailbox Write Event 0 = Event not asserted 1 = Slave interface completed a write transaction to the incoming mailbox	R/W1C	0
09	OMBR	Outgoing Mailbox Read Event 0 = Event not asserted 1 = Slave interface completed a read transaction to the outgoing mailbox when the OMB_FLAG was set The event is asserted only if the mailbox was full.	R/W1C	0
10	Reserved	Reserved	R	0
11	SCOL	Slave Collision Detect Event 0 = Event not asserted 1 = Slave interface detected a bit collision on the I^2C bus during a slave transaction initiated by an external master. The slave interface was not able to successfully assert a 1 for a data bit or for a NACK.	R/W1C	0
12	STRTO	Slave Transaction Timeout Event 0 = Event not asserted 1 = Transaction timer expired during a slave transaction initiated by an external master	R/W1C	0
13	SBTTO	Slave Byte Timeout Event 0 = Event not asserted 1 = Byte timer expired during a slave transaction initiated by an external master	R/W1C	0

(Continued)

Bits	Name	Description	Type	Reset Value
14	SSCLTO	Slave I2C_SCLK Low Timeout Event 0 = Event not asserted 1 = I2C_SCLK low timer expired during a slave transaction initiated by an external master	R/W1C	0
15:16	Reserved	Reserved	R	00
17	MTD	Master Transaction Done Event 0 = Event not asserted 1 = Master interface completed the I^2C transaction initiated through the "I2C Master Control Register"	R/W1C	0
18	Reserved	Reserved	R	0
19	BLTO	Boot Load Timeout Event 0 = Event not asserted 1 = Boot load timer expired This event only asserts during the boot load sequence if the Boot/Diagnostic timer expires. During normal operation, the DTIMER event will assert.	R/W1C	0
20	BLERR	Boot Load Error Event 0 = Event not asserted 1 = The boot load sequence failed due to an error during register reading: a protocol error (NACK when ACK expected), an I2C_SCLK low timer, collision after the device addressing phase, or the last six bytes of a register count field not being 0xFF. This error will be qualified by the MNACK, MCOL or MSCLTO event. The last data read from the EEPROM is visible in the EXI2C_REG_WADDR and EXI2C_REG_WDATA registers.	R/W1C	0
21	BLSZ	Boot Load Size Error Event 0 = Event not asserted 1 = The boot load sequence aborted because the count field is incorrect, indicating an improperly loaded boot EEPROM. The register count is visible in the EXI2C_REG_WADDR register.	R/W1C	0
22	BLNOD	Boot Load No Device Event 0 = Event not asserted 1 = The boot load sequencer received a NACK 6 times when trying to address the slave device. No device is responding to the boot load device address.	R/W1C	0

(Continued)

Bits	Name	Description	Type	Reset Value
23	BLOK	Boot Load OK Event 0 = Event not asserted 1 = The boot load sequence completed with no detectable errors. This bit is also asserted if boot load is disabled upon power up.	R/W1C	0
24:25	Reserved	Reserved	R	00
26	MNACK	Master NACK Received Event 0 = Event not asserted 1 = Master interface received a NACK from a slave device during a transaction initiated through the "I2C Master Control Register". This event can also assert during boot load, and provides more information on the source of a BLERR event.	R/W1C	0
27	MCOL	Master Collision Detect Event 0 = Event not asserted 1 = Master interface lost arbitration after it addressed the slave device during a transaction initiated through the "I2C Master Control Register". Another master is competing for access to the same slave device. This event can also assert during boot load, and provides more information on the source of a BLERR event.	R/W1C	0
28	MTRTO	Master Transaction Timeout Event 0 = Event not asserted 1 = Transaction timeout timer expired during a transaction initiated through the "I2C Master Control Register"	R/W1C	0
29	MBTTO	Master Byte Timeout Event 0 = Event not asserted 1 = Byte timeout timer expired during a transaction initiated through the "I2C Master Control Register"	R/W1C	0
30	MSCLTO	Master I2C_SCLK Low Timeout Event 0 = Event not asserted 1 = SCL_TO timeout timer expired during a transaction initiated through the "I2C Master Control Register". Another device is holding the I2C_SCLK signal low. This event can also assert during boot load, and provides more information on the source of a BLERR event.	R/W1C	0
31	MARBTO	Master Arbitration Timeout Event 0 = Event not asserted 1 = Arbitration timeout timer expired during a transaction initiated through the "I2C Master Control Register". Another master has control of the I^2C bus.	R/W1C	0

29.2.24 I2C New Event Register

This register indicates events that occurred since the last snapshot. This register is write-one-to-set. Writing a 1 to a bit position will set the event for diagnostic purposes. The register is cleared by writing to the I2C_EVENT register (see "I2C Event and Event Snapshot Registers") or by creating a snapshot by reading the "I2C Interrupt Status Register". For individual event descriptions, see the I2C_EVENT register.

Note: This register is affected by a reset controlled by the "I2C Reset Register". All events will be cleared and will not assert while SRESET is asserted in the "I2C Reset Register".

Register name: I2C_NEW_EVENT **Reset value: 0x0000_0000**	**Register offset: 0x308**

Bits	0	1	2	3	4	5	6	7
00:07	Reserved	SDW	SDR	SD	Reserved	DTIMER	DHIST	DCMDD
08:15	IMBW	OMBR	Reserved	SCOL	STRTO	SBTTO	SSCLTO	Reserved
16:23	Reserved	MTD	Reserved	BLTO	BLERR	BLSZ	BLNOD	BLOK
24:31	Reserved		MNACK	MCOL	MTRTO	MBTTO	MSCLTO	MARBTO

Bits	Name	Description	Type	Reset Value
00	Reserved	Reserved	R	0
01	SDW	Slave Internal Register Write Done Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
02	SDR	Slave Internal Register Read Done Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
03	SD	Slave Transaction Done Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
04	Reserved	Reserved	R	0
05	DTIMER	Diagnostic Timer Expired Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
06	DHIST	Diagnostic History Filling Event 0 = Event not asserted 1 = Event asserted	R/W1S	0

(Continued)

Bits	Name	Description	Type	Reset Value
07	DCMDD	Diagnostic Command Done Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
08	IMBW	Incoming Mailbox Write Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
09	OMBR	Outgoing Mailbox Read Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
10	Reserved	Reserved	R	0
11	SCOL	Slave Collision Detect Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
12	STRTO	Slave Transaction Timeout Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
13	SBTTO	Slave Byte Timeout Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
14	SSCLTO	Slave I2C_SCLK Low Timeout Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
15:16	Reserved	Reserved	R	00
17	MTD	Master Transaction Done Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
18	Reserved	Reserved	R	0
19	BLTO	Boot Load Timeout Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
20	BLERR	Boot Load Error Event 0 = Event not asserted 1 = Event asserted	R/W1S	0

(Continued)

Bits	Name	Description	Type	Reset Value
21	BLSZ	Boot Load Size Error Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
22	BLNOD	Boot Load No Device Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
23	BLOK	Boot Load OK Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
24:25	Reserved	Reserved	R	00
26	MNACK	Master NACK Received Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
27	MCOL	Master Collision Detect Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
28	MTRTO	Master Transaction Timeout Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
29	MBTTO	Master Byte Timeout Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
30	MSCLTO	Master I2C_SCLK Low Timeout Event 0 = Event not asserted 1 = Event asserted	R/W1S	0
31	MARBTO	Master Arbitration Timeout Event 0 = Event not asserted 1 = Event asserted	R/W1S	0

29.2.25 I2C Enable Event Register

This register modifies the function of the I2C_EVENT register (see "I2C Event and Event Snapshot Registers"). Each bit in this register enables (1) or disables (0) the corresponding event in the I2C_EVENT register from asserting in the "I2C Interrupt Status Register".

Register name: I2C_EVENT_ENB Reset value: 0x74DE_5F3F	Register offset: 0x30C

Bits	0	1	2	3	4	5	6	7
00:07	Reserved	SDW	SDR	SD	Reserved	DTIMER	DHIST	DCMDD
08:15	IMBW	OMBR	Reserved	SCOL	STRTO	SBTTO	SSCLTO	Reserved
16:23	Reserved	MTD	Reserved	BLTO	BLERR	BLSZ	BLNOD	BLOK
24:31	Reserved		MNACK	MCOL	MTRTO	MBTTO	MSCLTO	MARBTO

Bits	Name	Description	Type	Reset Value
00	Reserved	Reserved	R	0
01	SDW	Slave Internal Register Write Done Enable 0 = Event does not assert to interrupt status 1 = Event will assert in interrupt status	R/W	1
02	SDR	Slave Internal Register Read Done Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
03	SD	Slave Transaction Done Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
04	Reserved	Reserved	R	0
05	DTIMER	Diagnostic Timer Expired Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
06	DHIST	Diagnostic History Filling Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	0
07	DCMDD	Diagnostic Command Done Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
08	IMBW	Incoming Mailbox Write Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
09	OMBR	Outgoing Mailbox Read Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
10	Reserved	Reserved	R	0
11	SCOL	Slave Collision Detect Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
12	STRTO	Slave Transaction Timeout Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
13	SBTTO	Slave Byte Timeout Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
14	SSCLTO	Slave I2C_SCLK Low Timeout Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
15:16	Reserved	Reserved	R	00
17	MTD	Master Transaction Done Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
18	Reserved	Reserved	R	0
19	BLTO	Boot Load Timeout Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
20	BLERR	Boot Load Error Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
21	BLSZ	Boot Load Size Error Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1

(Continued)

Bits	Name	Description	Type	Reset Value
22	BLNOD	Boot Load No Device Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
23	BLOK	Boot Load OK Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
24:25	Reserved	Reserved	R	00
26	MNACK	Master NACK Received Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
27	MCOL	Master Collision Detect Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
28	MTRTO	Master Transaction Timeout Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
29	MBTTO	Master Byte Timeout Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
30	MSCLTO	Master I2C_SCLK Low Timeout Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1
31	MARBTO	Master Arbitration Timeout Enable 0 = Event does not assert to interrupt status 1 = Event will assert in the interrupt status	R/W	1

29.2.26 I2C Time Period Divider Register

This register provides programmable extension of the reference clock period into longer periods used by the timeout and idle detect timers.

Register name: I2C_DIVIDER **Reset value:** 0x004D_03E9	**Register offset: 0x320**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved				USDIV			
08:15	USDIV							
16:23	Reserved				MSDIV			
24:31	MSDIV							

Bits	**Name**	**Description**	**Type**	**Reset Value**
00:03	Reserved	Reserved	R	0x0
04:15	USDIV	Period Divider for Micro-Second Based Timers This field divides the reference clock down for use by the Idle Detect Timer, the Byte Timeout Timer, the I2C_SCLK Low Timeout Timer, and the Milli-Second Period Divider. Period(USDIV) = Period(Ref Clock) * (USDIV + 1), where Ref Clock is 12.8 ns. Reset period is 1 microsecond.	R/W	0x004D
16:19	Reserved	Reserved	R	0x0
20:31	MSDIV	Period Divider for Milli-Second Based Timers This field divides the USDIV period down further for use by the Arbitration Timeout Timer, the Transaction Timeout Timer, and the Boot/Diag Timeout Timer. Period(MSDIV) = Period(USDIV) * (MSDIV + 1). Reset period is 1 millisecond.	R/W	0x03E9

29.2.27 I²C Start Condition Setup/Hold Timing Register

This register programs the setup and hold timing for the Start condition when generated by the master control logic. The timer periods are relative to the reference clock. This register is shadowed during boot loading, and can be reprogrammed prior to a chain operation without affecting the bus timing for the current EEPROM.

Register name: I2C_START_SETUP_HOLD **Reset value:** 0x0170_0x013A	**Register offset: 0x340**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	START_SETUP							
08:15	START_SETUP							
16:23	START_HOLD							
24:31	START_HOLD							

Bits	**Name**	**Description**	**Type**	**Reset Value**
00:15	START_SETUP	Count for the START Condition Setup Period Defines the minimum setup time for the START condition; that is, both I2C_SCLK and I2C_SD seen high prior to I2C_SD pulled low. This is a master-only timing parameter. This value also doubles as the effective Stop Hold time. Period(START_SETUP) = (START_SETUP * Period(Ref Clock)), where Ref Clock is 12.8 ns. Reset time is 4.71 microseconds.	R/W	0x0170
16:31	START_HOLD	Count for the START Condition Hold Period Defines the minimum hold time for the START condition; that is, from I2C_SD seen low to I2C_SCLK pulled low. This is a master only timing parameter. Period(START_HOLD) = (START_HOLD * Period(Ref Clock)), where Ref Clock is 12.8 ns. Reset time is 4.01 microseconds.	R/W	0x013A

29.2.28 I2C Stop/Idle Timing Register

This register programs the setup timing for the Stop condition when generated by the master control logic, and the Idle Detect timer. The Start Setup time doubles as the Stop Hold. The timer period for the Stop setup is relative to the reference clock. The timer period for the Idle Detect is relative to the USDIV period. The STOP setup time is shadowed during boot loading, and can be reprogrammed prior to a chain operation without affecting the bus timing for the current EEPROM.

Register name: I2C_STOP_IDLE **Reset value:** 0x013A_0033	**Register offset: 0x344**

Bits	0	1	2	3	4	5	6	7
00:07	STOP_SETUP							
08:15	STOP_SETUP							
16:23	IDLE_DET							
24:31	IDLE_DET							

Bits	Name	Description	Type	Reset Value
00:15	STOP_SETUP	Count for STOP Condition Setup Period Defines the minimum setup time for the STOP condition; that is, both I2C_SCLK seen high and I2C_SD seen low prior to I2C_SD released high. This is a master-only timing parameter. Period(STOP_SETUP) = (STOP_SETUP * Period(Ref Clock)), where Ref Clock is 12.8 ns. Reset time is 4.01 microseconds.	R/W	0x013A
16:31	IDLE_DET	Count for Idle Detect Period Used in two cases. First, defines the period after reset during which the I2C_SCLK signal must be seen high to call the bus idle. This period is needed to avoid interfering with an ongoing transaction after reset. Second, defines the period before a master transaction during which the I2C_SCLK and I2C_SD signals must both be seen high to call the bus idle. This period is a protection against external master devices not correctly idling the bus. Period(IDLE_DET) = (IDLE_DET * Period(USDIV)), where USDIV is the microsecond time defined in the "I2C Time Period Divider Register". A value of zero results in no idle detect period, meaning the bus will be sensed as idle immediately. Reset time is 51 microseconds.	R/W	0x0033

29.2.29 I2C_SD Setup and Hold Timing Register

This register programs the setup and hold times for the I2C_SD signal when output by either the master or slave interface. It is shadowed during boot loading, and can be reprogrammed prior to a chain operation without affecting the bus timing for the current EEPROM.

Register name: I2C_SDA_SETUP_HOLD **Reset value:** 0x0063_0018	**Register offset: 0x348**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	SDA_SETUP							
08:15	SDA_SETUP							
16:23	SDA_HOLD							
24:31	SDA_HOLD							

Bits	**Name**	**Description**	**Type**	**Reset Value**
00:15	SDA_SETUP	Count for the I2C_SD Setup Period Defines the minimum setup time for the I2C_SD signal; that is, I2C_SD set to desired value prior to rising edge of I2C_SCLK. This applies to both slave and master interface. Note: This value should be set to the sum of the I2C_SD setup time and the maximum rise/fall time of the I2C_SD signal to ensure that the signal is valid on the output at the correct time. This time is different than the raw I2C_SD setup time in the *I^2C Specification*. Period(SDA_SETUP) = (SDA_SETUP * Period(Ref Clock)), where Ref Clock is 12.8 ns. Reset time is 1260 nanoseconds.	R/W	0x0063
16:31	SDA_HOLD	Count for I2C_SD Hold Period Defines the minimum hold time for the I2C_SD signal; that is, I2C_SD valid past the falling edge of I2C_SCLK. This applies to both slave and master interface. Period(SDA_HOLD) = (SDA_HOLD * Period(Ref Clock)), where Ref Clock is 12.8 ns. Reset time is 310 nanoseconds.	R/W	0x0018

29.2.30 I2C_SCLK High and Low Timing Register

This register programs the nominal high and low periods of the I2C_SCLK signal when generated by the master interface. It is shadowed during boot loading, and can be reprogrammed prior to a chain operation without affecting the bus timing for the current EEPROM.

Register name: I2C_SCL_PERIOD **Reset value:** 0x0187_0187	**Register offset: 0x34C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	SCL_HIGH							
08:15	SCL_HIGH							
16:23	SCL_LOW							
24:31	SCL_LOW							

Bits	**Name**	**Description**	**Type**	**Reset Value**
00:15	SCL_HIGH	Count for I2C_SCLK High Period Defines the nominal high period of the clock, from rising edge to falling edge of I2C_SCLK. This is a master-only parameter. The observed period may be shorter if other devices pull the clock low. Period(SCL_HIGH) = (SCL_HIGH * Period(Ref Clock)), where Ref Clock is 12.8 ns. Reset time is 5.00 microseconds (100 kHz).	R/W	0x0187
16:31	SCL_LOW	Count for I2C_SCLK Low Period Defines the nominal low period of the clock, from falling edge to rising edge of I2C_SCLK. This is a master-only parameter. The observed period may be longer if other devices pull the clock low. Period(SCL_LOW) = (SCL_LOW * Period(Ref Clock)), where Ref Clock is 12.8 ns. Reset time is 5.00 microseconds (100 kHz).	R/W	0x0187

29.2.31 I2C_SCLK Minimum High and Low Timing Register

This register programs the minimum high and low periods of the I2C_SCLK signal when generated by the master interface. It is shadowed during boot loading, and can be reprogrammed prior to a chain operation without affecting the bus timing for the current EEPROM.

Register name: I2C_SCL_MIN_PERIOD **Reset value:** 0x0139_016F	**Register offset: 0x350**

Bits	0	1	2	3	4	5	6	7
00:07	SCL_MINH							
08:15	SCL_MINH							
16:23	SCL_MINL							
24:31	SCL_MINL							

Bits	Name	Description	Type	Reset Value
00:15	SCL_MINH	Count for I2C_SCLK High Minimum Period Defines the minimum high period of the clock, from rising edge seen high to falling edge of I2C_SCLK. This is a master-only parameter. The observed period may be shorter if other devices pull the clock low. Period(SCL_MINH) = (SCL_MINH * Period(Ref Clock)), where Ref Clock is 12.8 ns. Reset time is 4.01 microseconds.	R/W	0x0139
16:31	SCL_MINL	Count for I2C_SCLK Low Minimum Period Defines the minimum low period of the clock, from falling edge seen low to rising edge of I2C_SCLK. This is a master-only parameter. The observed period may be longer if other devices pull the clock low. Period(SCL_MINL) = (SCL_MINL * Period(Ref Clock)), where Ref Clock is 12.8 ns. Reset time is 4.71 microseconds.	R/W	0x016F

29.2.32 I2C_SCLK Low and Arbitration Timeout Register

This register programs the I2C_SCLK low timeout and the Arbitration timeout. The arbitration timer period is relative to the MSDIV period, and the I2C_SCLK low timeout period is relative to the USDIV period.

Register name: I2C_SCL_ARB_TIMEOUT **Reset value: 0x65BB_0033**	**Register offset: 0x354**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	SCL_TO							
08:15	SCL_TO							
16:23	ARB_TO							
24:31	ARB_TO							

Bits	**Name**	**Description**	**Type**	**Reset Value**
00:15	SCL_TO	Count for I2C_SCLK Low Timeout Period Defines the maximum amount of time for a slave device holding the I2C_SCLK signal low. This timeout covers the period from I2C_SCLK falling edge to the next I2C_SCLK rising edge. Value 0x0 disables the timeout. Period(SCL_TO) = (SCL_TO * Period(USDIV)) where USDIV is the microsecond time defined in the "I2C Time Period Divider Register". The reset value of this timeout is 26,000 microseconds (26 milliseconds).	R/W	0x65BB
16:31	ARB_TO	Count for Arbitration Timeout Period Defines the maximum amount of time for the master interface to arbitrate for the bus before aborting the transaction. This timeout covers the period from master operation start (setting the START bit in the "I2C Master Control Register") until the ACK/NACK is received from the external slave for the slave device address. A value of 0 disables the timeout. Period(ARB_TO) = (ARB_TO * Period(MSDIV)) where MSDIV is the millisecond time defined in "I2C Time Period Divider Register". The reset value of this timeout is 51 milliseconds. However, this timeout is not active during the boot load sequence.	R/W	0x0033

29.2.33 I2C Byte/Transaction Timeout Register

This register programs the Transaction and Byte timeouts. The timer periods are relative to the USDIV period for the byte timeout, and relative to the MSDIV period for the transaction timeout.

Register name: I2C_BYTE_TRAN_TIMEOUT Reset value: 0x0000_0000	Register offset: 0x358

Bits	0	1	2	3	4	5	6	7
00:07	BYTE_TO							
08:15	BYTE_TO							
16:23	TRAN_TO							
24:31	TRAN_TO							

Bits	Name	Description	Type	Reset Value
00:15	BYTE_TO	Count for Byte Timeout Period Defines the maximum amount of time for a byte to be transferred on the I2C bus. This covers the period from Start condition to next ACK/NACK, between two successive ACK/NACK bits, or from ACK/NACK to Stop/Restart condition. A value of 0 disables the timeout. Period(BYTE_TO) = (BYTE_TO * Period(USDIV)) where USDIV is the microsecond time defined in "I2C Time Period Divider Register". This timeout is disabled on reset, and is not used during boot load.	R/W	0x0000
16:31	TRAN_TO	Count for Transaction Timeout Period Defines the maximum amount of time for a transaction on the I2C bus. This covers the period from Start to Stop. A value of 0 disables the timeout. Period(TRAN_TO) = (TRAN_TO * Period(MSDIV)) where MSDIV is the millisecond time defined in "I2C Time Period Divider Register". This timeout is disabled on reset, and is not used during boot load.	R/W	0x0000

29.2.34 I2C Boot and Diagnostic Timer

This register programs a timer that times out the boot load sequence, and can be used after boot load as a general purpose timer.

Register name: I2C_BOOT_DIAG_TIMER Reset value: 0x0000_0F9E	Register offset: 0x35C

Bits	0	1	2	3	4	5	6	7
00:07	FREERUN	Reserved						
08:15	Reserved							
16:23	COUNT							
24:31	COUNT							

Bits	Name	Description	Type	Reset Value
00	FREERUN	Free Running Timer 0 = Timer is not automatically restarted when expires 1 = When timer expires, timer is restarted with same COUNT	R/W	0
01:15	Reserved	Reserved	R	0x0000
16:31	COUNT	Count for Timer Period Defines period for timer. Initial reset value is used for overall boot load timeout. During normal operation, this timer can be used for any general purpose timing. A value of 0 disables the timeout. Period(DTIMER) = (COUNT * Period(MSDIV)), where MSDIV is the millisecond period define in "I2C Time Period Divider Register". Timer begins counting when this register is written. If this register is written while the counter is running, the timer is immediately restarted with the new COUNT, and the DTIMER/BLTO event is not generated. When the timer expires, either the BLTO or DTIMER event is generated, depending on whether the boot load sequence is active. If FREERUN is set to 1 when timer expires, then the timer is restarted immediately (the event is still generated), providing a periodic interrupt capability. The reset value for the boot load timeout is 4 seconds. If the boot load completes before the timer expires, the timer is set to zero (disabled).	R/W	0x0F9E

29.2.35 I²C Boot Load Diagnostic Progress Register

This register provides visibility of the register count and peripheral address during the boot load sequence.

Register name: I2C_BOOT_DIAG_PROGRESS Reset value: 0x0000_0000	Register offset: 0x3B8

Bits	0	1	2	3	4	5	6	7
00:07	REGCNT							
08:15	REGCNT							
16:23	PADDR							
24:31	PADDR							

Bits	Name	Description	Type	Reset Value
00:15	REGCNT	Register Count The number of registers remaining to load from the current EEPROM during the boot load sequence. This register is initialized to the count read from the first two bytes of the EEPROM after reset, or to the first two byte read after a boot chaining operation. The field counts down as each register address/data pair is read.	R	0x0000_0000
16:31	PADDR	Peripheral Address Value of current peripheral address used by the boot load sequence. This field is initialized to zero at reset, and increments as the boot load sequence progresses. If a chain operation is performed, this field is loaded with the new peripheral address from the "I2C Boot Control Register" (with the 3 LSBs set to zero).	R	0x0000_0000

This is a diagnostic register. Documentation is provided for reference purposes only. The function of this register is not guaranteed in future versions and usage thereof is not supported.

29.2.36 I²C Boot Load Diagnostic Configuration Register

This register provides visibility of boot sequence information.

Register name: I2C_BOOT_DIAG_CFG Reset value: Undefined	Register offset: 0x3BC

Bits	0	1	2	3	4	5	6	7
00:07	BOOTING	BDIS	PASIZE	PINC	Reserved			
08:15	Reserved							
16:23	Reserved							
24:31	Reserved	BOOT_ADDR						

Bits	Name	Description	Type	Reset Value
00	BOOTING	Booting 0 = Boot sequence not active 1 = Boot sequence in progress	R	0
01	BDIS	Boot Disabled 0 = Boot enabled 1 = Boot disabled	R	Undefined
02	PASIZE	Peripheral Address Size 0 = 1-byte peripheral address 1 = 2-byte peripheral address	R	Undefined
03	PINC	Page Increment 0 = Page increment disabled 1 = Page increment enabled	R	0
04:24	Reserved	Reserved	R	0x000
25:31	BOOT_ADDR	Boot Device Address Current value of the boot device address in use by the boot load sequence. This value is incremented during the bootload if the page increment feature is enabled.	R	0x00

This is a diagnostic register. Documentation is provided for reference purposes only. The function of this register is not guaranteed in future versions and usage thereof is not supported.

30. GPIO Registers

Topics discussed include the following:

30.1 Register Map

The following table lists the register map for the GPIO Interface. For information about Tsi620's device register map, see "Overview of Device Register Map".

Table 191: GPIO Register Map

Offset[a]	Register Name	See
0x000	GPIO0_DATA	"GPIO 0 Data Register"
0x004	GPIO0_CNTRL	"GPIO 0 Control Register"
0x008	GPIO0_EDGE_CNTRL	"GPIO 0 Edge Control Register"
0x00C	GPIO0_INT_STATUS	"GPIO 0 Interrupt Status Register"
0x010	GPIO1_DATA	"GPIO 1 Data Register"
0x014	GPIO1_CNTRL	"GPIO 1 Control Register"
0x018	GPIO1_EDGE_CNTRL	"GPIO 1 Edge Control Register"
0x01C	GPIO1_INT_STATUS	"GPIO 1 Interrupt Status Register"

a. It is recommended that GPIO input signals be grouped together, and be separate from outputs. The base offset for GPIO registers within the Tsi620 is 0x1ACE0. The GPIO registers support 32 GPIO signals.

30.2 Register Descriptions

This section describes the GPIO registers. These registers are reset by a chip reset.

30.2.1 GPIO 0 Data Register

This register presents the data on the GPIO Interface when configured as outputs, or defines the data on the GPIO Interface when configured as inputs.

Register name: GPIO0_DATA **Reset value: Undefined**	**Register offset: 0x000**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	GPIO_DATA_IN[15:8]							
08:15	GPIO_DATA_IN[7:0]							
16:23	GPIO_DATA_OUT[15:8]							
24:31	GPIO_DATA_OUT[7:0]							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:15	GPIO_DATA_IN[15:0]	Value from the input port. This field indicates the data currently on the GPIO Interface. This may be the same data that is being driven on the output port. Note: GPIO_DATA_IN[0] represents GPIO[0] data.	R	Undefined
16:31	GPIO_DATA_OUT[15:0]	Value driven on the output port. This field defines each bit of the output port. If the port is configured as an output with open-drain capability, when these bits are set to one the output is tristated (open-drain); when set to 0 the output is driven low. If the port is configured as an output, the value written to this field is driven on the output port. Note: GPIO_DATA_OUT[0] controls GPIO[0] data.	R/W	0xFFFF

30.2.2 GPIO 0 Control Register

This register controls the direction and output configuration of the GPIO Interface pins. The direction and configuration of each GPIO pin can be configured individually.

Register name: GPIO0_CNTRL Reset value: 0x0000_0000	Register offset: 0x004

Bits	0	1	2	3	4	5	6	7
00:07	GPIO_DIR[15:8]							
08:15	GPIO_DIR[7:0]							
16:23	GPIO_CFG[15:8]							
24:31	GPIO_CFG[7:0]							

Bits	Name	Description	Type	Reset Value
0:15	GPIO_DIR [15:0]	Direction of the GPIO pins 0 = Input, tri-stated output 1 = Output or Output with open-drain depending on configuration	R/W	0x0000
16:31	GPIO_CFG [15:0]	0 = Standard output 1 = Open-drain output	R/W	0x0000

30.2.3 GPIO 0 Edge Control Register

This register controls when an interrupt is generated by the state of a GPIO pin, or on the transition of a GPIO pin's state.

Register name: GPIO0_EDGE_CNTRL Reset value: 0x0000_0000	Register offset: 0x008

Bits	0	1	2	3	4	5	6	7
00:07	STATE[15:8]							
08:15	STATE[7:0]							
16:23	EDGE[15:8]							
24:31	EDGE[7:0]							

Bits	Name	Description	Type	Reset Value
0:15	STATE[15:0]	Specifies the state of the input pin that generates an interrupt. 0 = An interrupt is generated when the pin is low, or transitions to low as specified by the EDGE field on a per GPIO pin basis. 1 = An interrupt is generated when the pin is high, or transitions to high as specified by the EDGE field on a per GPIO pin basis.	R/W	0x0000
16:31	EDGE[15:0]	Specifies that an interrupt is generated 0 = On the transition of the GPIO pin to the state specified by STATE (that is, edge triggered) 1 = When the pin is in the state specified by STATE (that is, level triggered) Note: Edge/Level is specified on a per GPIO basis.	R/W	0x0000

30.2.4 GPIO 0 Interrupt Status Register

This register controls the interrupt status, and enables or disables interrupt generation on a per GPIO pin basis.

Register name: GPIO0_INT_STATUS Reset value: 0x0000_0000	Register offset: 0x00C

Bits	0	1	2	3	4	5	6	7
00:07	STATUS[15:8]							
08:15	STATUS[7:0]							
16:23	ENABLE[15:8]							
24:31	ENABLE[7:0]							

Bits	Name	Description	Type	Reset Value
0:15	STATUS [15:0]	Indicates the interrupt status on a per GPIO pin basis. When read, 0 = No Interrupt 1 = An interrupt has occurred When written, a 1 value will clear the corresponding interrupt detection	R/W1C	0x0000
16:31	ENABLE [15:0]	Enables interrupt generation on a per GPIO pin basis. 0 = Disables interrupt generation 1 = Enables interrupt generation	R/W	0x0000

30.2.5 GPIO 1 Data Register

This register presents the data on the GPIO Interface when configured as outputs, or defines the data on the GPIO Interface when configured as inputs.

Register name: GPIO1_DATA **Reset value: Undefined**	**Register offset: 0x010**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	GPIO_DATA_IN[31:24]							
08:15	GPIO_DATA_IN[23:16]							
16:23	GPIO_DATA_OUT[31:24]							
24:31	GPIO_DATA_OUT[23:16]							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:15	GPIO_DATA_IN[31:16]	Value from the input port. This field indicates the data currently on the GPIO Interface. This may be the same data that is being driven on the output port.	R	Undefined
16:31	GPIO_DATA_OUT[31:16]	Value driven on the output port. This field defines each bit of the output port. If the port is configured as an output with open-drain capability, when these bits are set to one the output is tristated (open-drain); when set to 0 the output is driven low. If the port is configured as an output, the value written to this field is driven on the output port.	R/W	0xFFFF

30.2.6 GPIO 1 Control Register

This register controls the direction and output configuration of the GPIO Interface pins. The direction and configuration of each GPIO pin can be configured individually.

Register name: GPIO1_CNTRL Reset value: 0x0000_0000	Register offset: 0x014

Bits	0	1	2	3	4	5	6	7
00:07	GPIO_DIR[31:24]							
08:15	GPIO_DIR[23:16]							
16:23	GPIO_CFG[31:24]							
24:31	GPIO_CFG[23:16]							

Bits	Name	Description	Type	Reset Value
0:15	GPIO_DIR [31:16]	Direction of the GPIO pins 0 = Input, tri-stated output 1 = Output or Output with open-drain depending on configuration	R/W	0x0000
16:31	GPIO_CFG [31:16]	0 = Standard output 1 = Open-drain output	R/W	0x0000

30.2.7 GPIO 1 Edge Control Register

This register controls when an interrupt is generated by the state of a GPIO pin, or on the transition of a GPIO pin's state.

Register name: GPIO1_EDGE_CNTRL Reset value: 0x0000_0000	Register offset: 0x018

Bits	0	1	2	3	4	5	6	7
00:07	STATE[31:24]							
08:15	STATE[23:16]							
16:23	EDGE[31:24]							
24:31	EDGE[23:16]							

Bits	Name	Description	Type	Reset Value
0:15	STATE[31:16]	Specifies the state of the input pin that generates an interrupt. 0 = An interrupt is generated when the pin is low, or transitions to low as specified by the EDGE field on a per GPIO pin basis. 1 = An interrupt is generated when the pin is high, or transitions to high as specified by the EDGE field on a per GPIO pin basis.	R/W	0x0000
16:31	EDGE[31:16]	Specifies that an interrupt is generated 0 = On the transition of the GPIO pin to the state specified by STATE (that is, edge triggered) 1 = When the pin is in the state specified by STATE (that is, level triggered) Note: Edge/Level is specified on a per GPIO basis.	R/W	0x0000

30.2.8 GPIO 1 Interrupt Status Register

This register controls the interrupt status, and enables or disables interrupt generation on a per GPIO pin basis.

Register name: GPIO1_INT_STATUS Reset value: 0x0000_0000	Register offset: 0x01C

Bits	0	1	2	3	4	5	6	7
00:07	STATUS[31:24]							
08:15	STATUS[23:16]							
16:23	ENABLE[31:24]							
24:31	ENABLE[23:16]							

Bits	Name	Description	Type	Reset Value
0:15	STATUS [31:16]	Indicates the interrupt status on a per GPIO pin basis. When read, 0 = No Interrupt 1 = An interrupt has occurred When written, a 1 value will clear the corresponding interrupt detection	R/W1C	0x0000
16:31	ENABLE [31:16]	Enables interrupt generation on a per GPIO pin basis. 0 = Disables interrupt generation 1 = Enables interrupt generation	R/W	0x0000

31. Reset Control Registers

31.1 Register Descriptions

This section describes the Tsi620 reset control register. This register is reset by a chip reset. The base offset for reset control registers within the Tsi620 is 0x1AC80. For information about Tsi620's device register map, see "Overview of Device Register Map".

31.1.1 Block Reset Control Register

This register controls the various aspects of a Block Reset within the Tsi620. It is only reset by a chip reset (power-up) of the Tsi620.

Register name: BLK_RST_CTL **Reset value: ECF8_0000**	**Register offset: 000**

Bits	0	1	2	3	4	5	6	7
00:7	SRIO	SREP	PCI	DO_ RESET	BISF	SISF	Reserved	CHIP_ RESET
08:15	PGTBR	PGTSW	I2C	I2C_BOOT	PCI_SELF _RST	Reserved		
16:23	Reserved							
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0	SRIO	When this bit is set all Switch Ports are reset when the BLK_RST_n pin is asserted. Note: SRIO and SISF must be reset together. When a Switch reset occurs, SRIO and SISF must both be set.	R/W	1
1	SREP	When this bit is set the SREP block is reset when the BLK_RST_n pin is asserted. The sticky bits in this block are not cleared. Note: SREP, PCI, and BISF must be reset together. When a Bridge reset occurs, SREP, PCI, and BISF must all be set.	R/W	1
2	PCI	When this bit is set the PCI Interface is reset when the BLK_RST_n pin is asserted. Note: SREP, PCI, and BISF must be reset together. When a Bridge reset occurs, SREP, PCI, and BISF must all be set.	R/W	1

(Continued)

Bits	Name	Description	Type	Reset Value
3	DO_RESET	When this bit is set, a reset is triggered. If the CHIP_RESET bit is set, then a chip reset occurs. If the CHIP_RESET bit is cleared, then a block reset occurs and the other bits in this register control which blocks are reset (see "Chip Reset").	R/W1S	0
4	BISF	When this bit is set the Bridge ISF is reset when the BLK_RST_b pin is asserted. Note: SREP, PCI, and BISF must be reset together. When a Bridge reset occurs, SREP, PCI, and BISF must all be set.	R/W	1
5	SISF	When this bit is set the Switch ISF and the Multicast Engine are reset when the BLK_RST_n pin is asserted. Note: SRIO and SISF must be reset together. When a Switch reset occurs, SRIO and SISF must both be set.	R/W	1
6	Reserved	N/A	R	0
7	CHIP_RESET	When this bit is set and a Block (BLK_RST_b assertion, or DO_RESET is set to 1), Switch, or Bridge reset is performed, all of the Tsi620 blocks are reset regardless of the other settings in this register. This is the same as asserting the CHIP_RST_b pin. Note: All sticky bits are cleared.	R/W	0
8	PGTBR	Propagate Bridge Reset to the Switch. When this bit is set and a Bridge reset occurs, a Switch reset also occurs.	R/W	1
9	PGTSW	Propagate Switch Reset to the Bridge. When this bit is set and a Switch reset occurs, a Bridge reset also occurs.	R/W	1
10	I2C	When this bit is set the I^2C Interface is reset when • the BLK_RST_n pin is asserted • a Switch Reset is triggered • a Bridge Reset is propagated to a Switch Reset This bit can only be set if all other blocks are reset. This means that the SRIO, SREP, BISF, SISF, and PCI bits must be set when this bit is set. When performing a Switch reset, if this bit is set, then the PGTSW bit must be set. When performing a Bridge reset, if this bit is set, then the PGTBR bit must be set.	R/W	1
11	I2C_BOOT	When this bit is set, and the I^2C Interface is reset, the I^2C Interface attempts to initialize the Tsi620's registers from an EEPROM device. This bit has no effect if a Chip Reset is performed (either CHIP_RESET is set in this register when DO_RESET is set to 1, or the CHIP_RST_b pin is asserted). This bit can only be set if the I2C bit is set. Setting this bit when the I2C bit is not set has no effect.	R/W	1

(Continued)

Bits	Name	Description	Type	Reset Value
12	PCI_SELF_RST	When this bit is set, and the PCI_RST input is asserted and is an input, a reset occurs. Which blocks are reset, and the type of reset, are controlled by the bits within this register. If this bit is not set, then the PCI_RESET_RX bit is set and the RST_IRQ_b pin is asserted (see "Reset Event Sources").	R/W	1
13:31	Reserved	N/A	R	0

32. Clock Generator Registers

Topics discussed include the following:

- "Register Map"
- "Register Descriptions"

32.1 Register Map

Table 192 lists the register map for the Clock Generator. The base offset for clock generator registers within the Tsi620 is 0x1AD00. For information about Tsi620's device register map, see "Overview of Device Register Map".

Table 192: Clock Generator Register Map

Offset	Register name	See
0x000	CG_INT_STATUS	"Clock Generator Interrupt Status Register"
0x004	CG_INT_SET	"Clock Generator Interrupt Set Register"
0x008	CG_INT_CTRL	"Clock Generator Interrupt Control Register"
0x00C	Reserved	
0x010	CG_PLL0_CTRL0	"Clock Generator PLL0 Control Register 0"
0x014–01C	Reserved	
0x020	CG_PLL0_CTRL1	"Clock Generator PLL0 Control Register 1"
0x024–02C	Reserved	
0x030	CG_IO_CTRL	"Clock Generator Output Control Register"
0x034	CG_RIO_PWRUP_STATUS	"Clock Generator RapidIO Power-up Status Register"
0x038	CG_PWRUP_STATUS	"Clock Generator Tsi620 Power-up Status Register"

32.2 Register Descriptions

The following section describes the Clock Generator registers. These registers are reset by a chip reset.

32.2.1 Clock Generator Interrupt Status Register

This register monitors the lock status of Tsi620's PLLs.

Register name: CG_INT_STATUS **Reset value: 0x0000_0000**	**Register offset: 0x000**

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
31:24	Reserved							
23:16	Reserved						LOL	
15:08	Reserved							FB_SLIP
07:00	Reserved							R_SLIP

Bits	**Name**	**Description**	**Type**	**Reset value**
31:18	Reserved	Reserved	R	0
17:16	LOL	Loss of Lock x1 = Clock Generator 1x = PCI clock de-skew PLL Each bit indicates that the corresponding PLL has lost, and possibly regained, lock. Each bit is an independent interrupt. This means that all combinations of bits [17:16] are valid values for this field. Multiple bits can be set if more than one PLL loses lock.	R/W1C	0
15:9	Reserved	Reserved	R	0
8	FB_SLIP	Feedback Slip Indication 0 = No slip 1 = Slip detected. A slip indicates that the PLL's VCO was operating too fast.	R/W1C	0
7:1	Reserved	Reserved	R	0
0	R_SLIP	Reference Slip Indication 0 = No slip 1 = Slip detected. A slip indicates that the PLL's VCO was operating too fast.	R/W1C	0

32.2.2 Clock Generator Interrupt Set Register

This register allows software to set interrupts based on the lock status of Tsi620's PLLs.

Register name: CG_INT_SET Reset value: 0x0000_0000	Register offset: 0x004

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved						LOL_SET	
15:08	Reserved							FB_SLIP_SET
07:00	Reserved							R_SLIP_SET

Bits	Name	Description	Type	Reset value
31:18	Reserved	Reserved	R	0
17:16	LOL_SET	Loss of Lock Set 0 = No effect 1 = Set the corresponding bit in the "Clock Generator Interrupt Status Register"	R/W1S	0
15:9	Reserved	Reserved	R	0
8	FB_SLIP_SET	Feedback Slip Indication Set 0 = No effect 1 = Set the corresponding bit in the "Clock Generator Interrupt Status Register"	R/W1S	0
7:1	Reserved	Reserved	R	0
0	R_SLIP_SET	Reference Slip Indication Set 0 = No effect 1 = Set the corresponding bit in the "Clock Generator Interrupt Status Register"	R/W1S	0

32.2.3 Clock Generator Interrupt Control Register

This register enables interrupts on the lock status of Tsi620's PLLs.

Register name: CG_INT_CTRL Reset value: 0x0000_0000	Register offset: 0x008

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved						LOL_EN	
15:08	Reserved							FB_ SLIP_EN
07:00	Reserved							R_ SLIP_EN

Bits	Name	Description	Type	Reset value
31:18	Reserved	Reserved	R	0
17:16	LOL_EN	Loss of Lock Interrupt Enable 0 = Interrupt disabled 1 = Enable the corresponding interrupt in the "Clock Generator Interrupt Status Register"	R/W	0
15:9	Reserved	Reserved	R	0
8	FB_SLIP_EN	Feedback Slip Interrupt Enable 0 = Interrupt disabled 1 = Enable the corresponding interrupt in the "Clock Generator Interrupt Status Register"	R/W	0
7:1	Reserved	Reserved	R	0
0	R_SLIP_EN	Reference Slip Interrupt Enable 0 = Interrupt disabled 1 = Enable the corresponding interrupt in the "Clock Generator Interrupt Status Register"	R/W	0

32.2.4 Clock Generator PLL0 Control Register 0

This register controls generation of clocks by the Clock Generator PLL.

Register name: CG_PLL0_CTRL0 Reset value: 0x0000_0000	Register offset: 0x010

Bits	7	6	5	4	3	2	1	0
31:24	Reserved	PWRDWN	Reserved					Reserved
23:16	Reserved							
15:08	Reserved							
07:00	Reserved							

Bits	Name	Description	Type	Reset value
31	Reserved	N/A	R	0
30	PWRDWN	Power Down 0 = Normal operation 1 = Power down the PLL Note: Setting this bit powers down the analog section of the PLL and puts the PLL in bypass mode. The output of the PLL is a copy of the reference clock.	R/W	0
29:0	Reserved	Reserved	R	00000

32.2.5 Clock Generator PLL0 Control Register 1

This register controls the programming setup of the Clock Generator PLLs.

Register name: CG_PLL0_CTRL1 Reset value: 0x0000_0007	Register offset: 0x020

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved							
15:08	Reserved							BWADJ
07:00	BWADJ							

Bits	Name	Description	Type	Reset value
31:9	Reserved	Reserved	R	0
8:0	BWADJ	Bandwidth Adjustment This field adjusts the loop bandwidth of the PLL. By reducing the bandwidth, the PLL will filter out progressively lower frequency input jitter. However, lowering the bandwidth will also increase long-term jitter. Not all values of BWADJ will give desirable results. It is recommended to keep this field value at 7 or higher.	R/W	0x7

32.2.6 Clock Generator Output Control Register

This register enables and disables the clock outputs from the Clock Generator.

Register name: CG_IO_CTRL Reset value: 0x0000_1F00	Register offset: 0x030

Bits	7	6	5	4	3	2	1	0
31:24	Reserved							
23:16	Reserved			PCI_VAL				
15:08	Reserved			PCI_EN				
07:00	Reserved							

Bits	Name	Description	Type	Reset value
31:21	Reserved	Reserved	R	0
20:16	PCI_VAL	Value driven for disabled PCI output clocks	R/W	0b00000
15:13	Reserved	Reserved	R	0
12:8	PCI_EN	PCI output clock enable Setting these bits high enables the corresponding PCI output clock. When a clock output is disabled, its output is set to the value set by the corresponding bit in PCI_VAL.	R/W	0b11111
7:0	Reserved	Reserved	R	0

32.2.7 Clock Generator RapidIO Power-up Status Register

This register shows the status of the RapidIO-related Tsi620 power-up configuration.

Register name: CG_RIO_PWRUP_STATUS Reset value: Undefined	Register offset: 0x034

Bits	7	6	5	4	3	2	1	0
31:24	SP_MAST_EN	SP_HOST	SP_TX_SWAP	SP_RX_SWAP	SP_REF_CLK		SP_IO_SPEED	
23:16	Reserved				SP6_MODE	SP4_MODE	SP2_MODE	SP0_MODE
15:08	Reserved							
07:00	Reserved	SP6_PWRDN	SP5_PWRDN	SP4_PWRDN	SP3_PWRDN	SP2_PWRDN	SP1_PWRDN	Reserved

Bits	Name	Description[a]	Type	Reset value
31	SP_MAST_EN	Configuration pin name.	R	Undefined
30	SP_HOST	Configuration pin name.	R	Undefined
29	SP_TX_SWAP	Configuration pin name.	R	Undefined
28	SP_RX_SWAP	Configuration pin name.	R	Undefined
27:26	SP_REF_CLK	Configuration pin name.	R	Undefined
25:24	SP_IO_SPEED	Configuration pin name.	R	Undefined
23:20	Reserved	N/A	R	Undefined
19	SP6_MODE	Configuration pin name.	R	Undefined
18	SP4_MODE	Configuration pin name.	R	Undefined
17	SP2_MODE	Configuration pin name.	R	Undefined
16	SP0_MODE	Configuration pin name.	R	Undefined
15:07	Reserved	N/A	R	Undefined
6	SP6_PWRDN	Configuration pin name.	R	Undefined
5	SP5_PWRDN	Configuration pin name.	R	Undefined
4	SP4_PWRDN	Configuration pin name.	R	Undefined
3	SP3_PWRDN	Configuration pin name.	R	Undefined
2	SP2_PWRDN	Configuration pin name.	R	Undefined

(Continued)

Bits	Name	Description[a]	Type	Reset value
1	SP1_PWRDN	Configuration pin name.	R	Undefined
0	Reserved	N/A	R	Undefined

a. For information on the pins referenced in this register, see “Power-up”.

32.2.8 Clock Generator Tsi620 Power-up Status Register

This register shows the status of the non-RapidIO Tsi620 power-up configuration.

Register name: CG_PWRUP_STATUS Reset value: Undefined	Register offset: 0x038

Bits	7	6	5	4	3	2	1	0
31:24	Reserved						VARIANT	
23:16	Reserved			PCI_HOLD_BOOT	PCI_ARBEN	PCI_RSTDIR	PCI_M66EN	PCI_PLL_BYPASS
15:08	Reserved				I2C_SEL	I2C_DISABLE	I2C_SLAVE	I2C_MA
07:00	Reserved	I2C_SA[6:0]						

Bits	Name	Description[a]	Type	Reset value
31:26	Reserved	N/A	R	0
25:24	VARIANT	Configuration pin name.	R	Undefined
23:21	Reserved	N/A	R	0
20	PCI_HOLD_BOOT	Configuration pin name.	R	Undefined
19	PCI_ARBEN	Configuration pin name.	R	Undefined
18	PCI_RSTDIR	Configuration pin name.	R	Undefined
17	PCI_M66EN	Configuration pin name.	R	Undefined
16	PCI_PLL_BYPASS	Configuration pin name.	R	Undefined
15:12	Reserved	N/A	R	0
11	I2C_SEL	Configuration pin name.	R	Undefined
10	I2C_DISABLE	Configuration pin name.	R	Undefined
9	I2C_SLAVE	Configuration pin name.	R	Undefined
8	I2C_MA	Configuration pin name.	R	Undefined
7	Reserved	N/A	R	0
6:0	I2C_SA	Configuration pin name.	R	Undefined

a. For information on the pins referenced in this register, see “Power-up”.

33. Event Management Registers

Topics discussed include the following:

- "Register Map"
- "Register Descriptions"

33.1 Register Map

The base offset for the Tsi620's Event Management registers is 0x1ACC0.

The following table lists the register map for the Event Management registers. For information about Tsi620's device register map, see "Overview of Device Register Map".

Table 193: Event Management Register Map

Offset	Register Name	See
0x000	EVENT_ROUTE_1	"Event Routing Register 1"
0x004	EVENT_ROUTE_2	"Event Routing Register 2"
0x010	BLOCK_STATUS	"Block Event Status Register"

33.2 Register Descriptions

33.2.1 Event Routing Register 1

This register controls how Bridge ISF, I^2C, Tsi620 Switch, and SREP event notification is performed by the Tsi620.

Register name: EVENT_ROUTE_1 **Reset value: 0x1110_1111**	**Register offset: 0x000**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	Reserved	BISF_ERR			Reserved	I2C		
08:15	Reserved	SWITCH_ERR			Reserved			
16:23	Reserved	SREP_MCS_RX			Reserved	SREP_PW_RX		
24:31	Reserved	SREP_DB_RX			Reserved	SREP_ERR		

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	Reserved	N/A	R	0
1:3	BISF_ERR	Route the Bridge ISF error event according to the following: 0 = No notification for the Bridge ISF error 1 = Assert the Tsi620 interrupt output 2 = Force the SREP to send a port-write 3–6 = Reserved 7 = Map to PCI INTA This bit corresponds to the BISF_ERR bit in the "Block Event Status Register".	R/W	1
4	Reserved	N/A	R	0
5:7	I2C	Route the I^2C event according to the following: 0 = No notification for the I2C event 1 = Assert the Tsi620 interrupt output 2 = Force the SREP to send a port-write 3–6 = Reserved 7 = Map to PCI INTA This bit corresponds to the I2C bit in the "Block Event Status Register".	R/W	1
8	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
9:11	SWITCH_ERR	Route the Switch event according to the following: 0 = No notification for the Switch event 1 = Assert the Tsi620 interrupt output 2–6 = Reserved 7 = Map to PCI INTA This bit corresponds to the SWITCH_ERR bit in the "Block Event Status Register".	R/W	1
12:16	Reserved	N/A	R	0
17:19	SREP_MCS_RX	Route the SREP MCS event according to the following: 0 = No notification for the SREP MCS event 1 = Assert the Tsi620 interrupt output 2–6 = Reserved 7 = Map to PCI INTA This bit corresponds to the SREP_MCS_RX bit in the "Block Event Status Register".	R/W	1
20	Reserved	N/A	R	0
21:23	SREP_PW_RX	Route the SREP Port-Write Received event according to the following: 0 = No notification for the SREP Port-Write Received event 1 = Assert the Tsi620 interrupt output 2–6 = Reserved 7 = Map to PCI INTA This bit corresponds to the SREP_PW_RX bit in the "Block Event Status Register".	R/W	1
24	Reserved	N/A	R	0
25:27	SREP_DB_RX	Route the SREP Doorbell Received event according to the following: 0 = No notification for the SREP Doorbell Received event 1 = Assert the Tsi620 interrupt output 2–6 = Reserved 7 = Map to PCI INTA This bit corresponds to the SREP_DB_RX bit in the "Block Event Status Register".	R/W	1
28	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
29:31	SREP_ERR	Route the SREP Error event according to the following: 0 = No notification for the SREP Error event 1 = Assert the Tsi620 interrupt output 2–6 = Reserved 7 = Map to PCI INTA This bit corresponds to the SREP_ERR bit in the "Block Event Status Register". Note that events within the SREP can be routed directly to port-write events, or to the SREP Error event. By default, all SREP Error events are mapped to the SREP_ERR interrupt. For more information on the routing of SREP events, see "Event Notification and Register Hierarchy".	R/W	1

33.2.2 Event Routing Register 2

This register controls how PCI, GPIO, and Clock Generator event notification is performed by the Tsi620.

Register name: EVENT_ROUTE_2 Reset value: 0x0111_0100	Register offset: 0x004

Bits	0	1	2	3	4	5	6	7
00:7	Reserved					PCI_ERR		
08:15	Reserved	GPIO1			Reserved	GPIO0		
16:23	Reserved					CLK_GEN		
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0:4	Reserved	N/A	R	0
5:7	PCI_ERR	Route the PCI error event according to the following: 0 = No notification for the PCI error 1 = Assert the Tsi620 interrupt output 2 = Force the SREP to send a port-write 3–6 = Reserved 7 = Map to PCI INTA This corresponds to the PCI_ERR bit in the “Block Event Status Register”.	R/W	1
8	Reserved	N/A	R	0
9:11	GPIO1	Route the GPIO1 error event according to the following: 0 = No notification for the GPIO1 event 1 = Assert the Tsi620 interrupt output 2 = Force the SREP to send a port-write 3–6 = Reserved 7 = Map to PCI INTA This corresponds to the GPIO1 bit in the “Block Event Status Register”.	R/W	1
12	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
13:15	GPIO0	Route the GPIO0 error event according to the following: 0 = No notification for the GPIO0 event 1 = Assert the Tsi620 interrupt output 2 = Force the SREP to send a port-write 3–6 = Reserved 7 = Map to PCI INTA This corresponds to the GPIO0 bit in the "Block Event Status Register".	R/W	1
16:20	Reserved	N/A	R	0
21:23	CLK_GEN	Route the Clock Generator interrupt event according to the following: 0 = No notification for the CLK_GEN event 1 = Assert the Tsi620 interrupt output 2 = Force the SREP to send a port-write 3–6 = Reserved 7 = Map to PCI INTA This corresponds to the CLK_GEN bit in the "Block Event Status Register".	R/W	1
24:31	Reserved	N/A	R	0

33.2.3 Block Event Status Register

This register contains the event status of all the Tsi620's blocks. For information on the sources of all of these events, see "Reset Event Sources".

Register name: BLOCK_STATUS Reset value: 0x0000_0000	Register offset: 0x010

Bits	0	1	2	3	4	5	6	7
00:7	Reserved							
08:15	Reserved							
16:23	Reserved		SW_RESET_RX	PCI_RESET_RX	CLK_GEN	GPIO1	GPIO0	PCI_ERR
24:31	BISF_ERR	I2C	SWITCH_ERR	SREP_RESET_RX	SREP_MCS_RX	SREP_PW_RX	SREP_DB_RX	SREP_ERR

Bits	Name	Description	Type	Reset Value
0:17	Reserved	N/A	R	0
18	SW_RESET_RX	Switch Reset Request event status 0 = Not asserted 1 = Asserted This event is caused when a RapidIO port receives a reset request, and the RapidIO port is configured to assert RST_IRQ_n (see "Switch Port Reset Request Event").	R	0
19	PCI_RESET_RX	PCI Reset Request event status 0 = Not asserted 1 = Asserted This event is caused by the assertion of the PCI_RSTn. Notification of a PCI reset request is controlled by the setting of PCI_SELF_RST in the "Block Reset Control Register".	R	0
20	CLK_GEN	Clock Generator event status 0 = Not asserted 1 = Asserted Notification of this event is controlled by CLK_GEN in the "Event Routing Register 2". For more information on the causes of this event, see "Clock Generator Interrupt Status Register".	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
21	GPIO1	GPIO1 event status 0 = Not asserted 1 = Asserted Notification of this event is controlled by GPIO1 in the "Event Routing Register 2". For more information on the causes of this event, see "GPIO 0 Interrupt Status Register".	R	0
22	GPIO0	GPIO0 event status 0 = Not asserted 1 = Asserted Notification of this event is controlled by GPIO0 in the "Event Routing Register 2". For more information on the causes of this event, see "GPIO 0 Interrupt Status Register".	R	0
23	PCI_ERR	PCI error event status 0 = Not asserted 1 = Asserted Notification of this event is controlled by PCI_ERR in the "Event Routing Register 2". For more information on the causes of this event, see "PCI Interrupt Status Register".	R	0
24	BISF_ERR	Bridge ISF event status 0 = Not asserted 1 = Asserted Notification of this event is controlled by BISF_ERR in the "Event Routing Register 1". For more information on the causes of this event, see "Bridge ISF TEA Interrupt Status Register", "Bridge ISF ECC CE Interrupt Status Register", and "Bridge ISF ECC UE Interrupt Status Register".	R	0
25	I2C	I2C event status 0 = Not asserted 1 = Asserted Notification of this event is controlled by I2C in the "Event Routing Register 1". For more information on the causes of this event, see "I2C Interrupt Status Register".	R	0
26	SWITCH_ERR	Switch event status 0 = Not asserted 1 = Asserted Notification of this event is controlled by SWITCH_ERR in the "Event Routing Register 1". For more information on the causes of this event, see "Switch Interrupt Status Register".	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
27	SREP_RESET_RX	SREP Reset Request event status 0 = Not asserted 1 = Asserted For more information on the causes of this event, see "Reset Control Symbol Processing".	R	0
28	SREP_MCS_RX	SREP MCS Received event status 0 = Not asserted 1 = Asserted Notification of this event is controlled by SREP_MCS_RX in the "Event Routing Register 1". For more information on the causes of this event, see "SREP Interrupt Status Register".	R	0
29	SREP_PW_RX	SREP Port-Write Received event status 0 = Not asserted 1 = Asserted Notification of this event is controlled by SREP_PW_RX in the "Event Routing Register 1". For more information on the causes of this event, see "SREP Interrupt Status Register".	R	0
30	SREP_DB_RX	SREP Doorbell Received event status 0 = Not asserted 1 = Asserted. Notification of this event is controlled by SREP_DB_RX in the "Event Routing Register 1". For more information on the causes of this event, see "SREP Interrupt Status Register".	R	0
31	SREP_ERR	SREP error event status 0 = Not asserted 1 = Asserted Notification of this event is controlled by SREP_ERR in the "Event Routing Register 1". For more information on the causes of this event, see the ISF_LOG, RIO_LOG, and IMPL_PHY_ERRin the "SREP Interrupt Status Register".	R	0

Index

Numerics

A

B

C

D

E

R

S

T

V

W

CORPORATE HEADQUARTERS
6024 Silver Creek Valley Road
San Jose, CA 95138

for SALES:
800-345-7015 or 408-284-8200
www.idt.com

for Tech Support:
email: srio@idt.com

June 4, 2013