IDT® Tsi564A
Serial RapidIO Switch

User Reference Manual

80B802A_MA001_05

August 18, 2009

6024 Silver Creek Valley Road San Jose, California 95138
Telephone: (408) 284-8200 • FAX: (408) 284-3572
Printed in U.S.A.

GENERAL DISCLAIMER

Integrated Device Technology, Inc. ("IDT") reserves the right to make changes to its products or specifications at any time, without notice, in order to improve design or performance. IDT does not assume responsibility for use of any circuitry described herein other than the circuitry embodied in an IDT product. Disclosure of the information herein does not convey a license or any other right, by implication or otherwise, in any patent, trademark, or other intellectual property right of IDT. IDT products may contain errata which can affect product performance to a minor or immaterial degree. Current characterized errata will be made available upon request. Items identified herein as "reserved" or "undefined" are reserved for future definition. IDT does not assume responsibility for conflicts or incompatibilities arising from the future definition of such items. IDT products have not been designed, tested, or manufactured for use in, and thus are not warranted for, applications where the failure, malfunction, or any inaccuracy in the application carries a risk of death, serious bodily injury, or damage to tangible property. Code examples provided herein by IDT are for illustrative purposes only and should not be relied upon for developing applications. Any use of such code examples shall be at the user's sole risk.

The IDT logo is registered to Integrated Device Technology, Inc. IDT and CPS are trademarks of Integrated Device Technology, Inc.

.

“Accelerated Thinking” is a service mark of Integrated Device Technology, Inc.

Contents

Figures

Tables

About this Document

This section discusses the following topics:

Scope

The *Tsi564A Serial RapidIO Switch User Manual* discusses the features, capabilities, and configuration requirements for the Tsi564A. It is intended for hardware and software engineers who are designing system interconnect applications with the device.

Document 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 underscore and lowercase "b". 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 illustrates the non-differential signal naming convention.

State	Single-line signal	Multi-line signal
Active low	NAME_b	NAME_b[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 illustrates 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

80B802A_MA001_05, Final, August 2009

This is the current release of the *Tsi564A Serial RapidIO Switch User Manual.* There have been no technical changes to this document. The formatting has been updated to reflect IDT.

80B802A_MA001_04, Final, June 2007

Changes occurred throughout this document. This revision represents the production version of the device.

80B802A_MA001_03, Final, March 2006

An error was corrected in the industrial part number (see Section B on page 249). No other information has been edited in this document.

80B802A_MA001_02, Final, March 2006

This was the production version of the *Tsi564A Serial RapidIO Switch User Manual.* Part numbers were changed in the ordering information to reflect the production status of the device.

80B802A_MA001_01, Advance, April 2005

This was the first release of the *Tsi564A Serial RapidIO Switch User Manual.*

1. Functional Overview

This chapter describes the main features and functions of the Tsi564A. This chapter includes the following information:

1.1 Tsi564A Overview

The IDT Tsi564A™ is an industry leading 40 Gbits/s Serial RapidIO switch. The Serial RapidIO Switch is part of a family of switches that enable customers to develop systems with robust features and high performance at low cost.

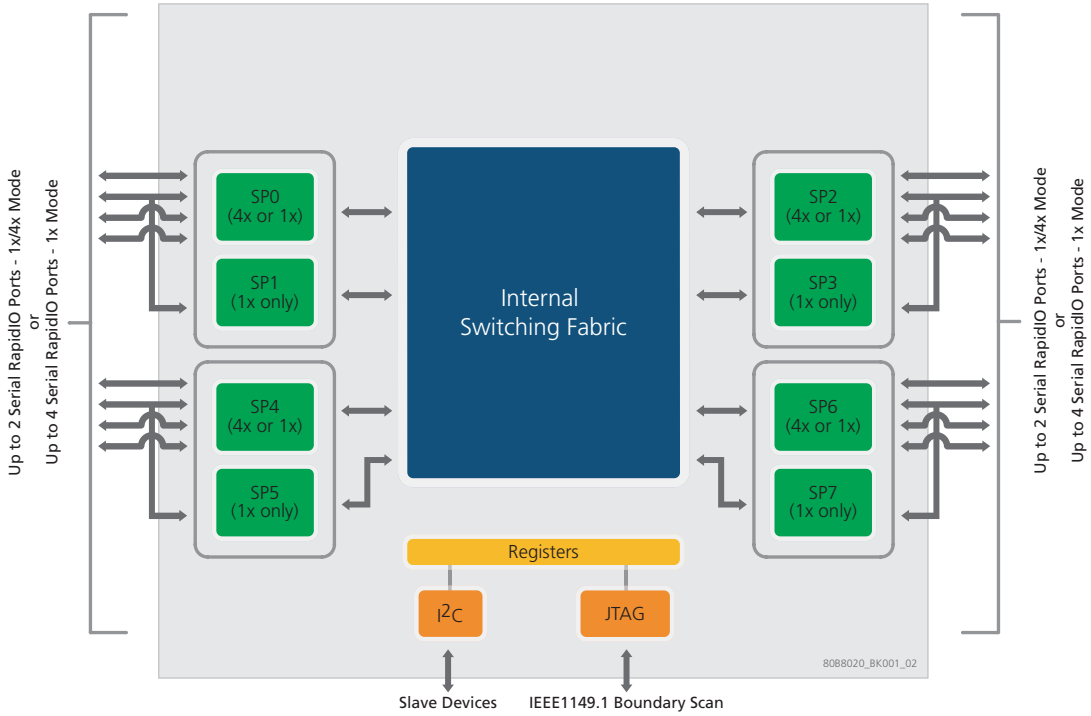
The Tsi564A provides designers and architects with maximum scalability in order to design the device in a wide range of applications. Flexible port configurations can be selected through multiple port width and frequency options.

Based on the *Serial RapidIO Specification,* the Tsi564A incorporates SerDes functionality, error recovery, priority-based fabric routing, high payload efficiency, and table-based fabric packet routing. In addition, the Tsi564A supports RapidFabric extensions including data streaming packet switching for interworking and encapsulation.

The device goes beyond standard specification requirements to solve system level issues with performance optimization through port statistics gathering, low power consumption, and hot swappable I/Os. The extensive buffering and traffic management architecture is specifically designed for line rate termination and the prevention of head-of-line blocking.

Figure 1 shows the main components of the device.

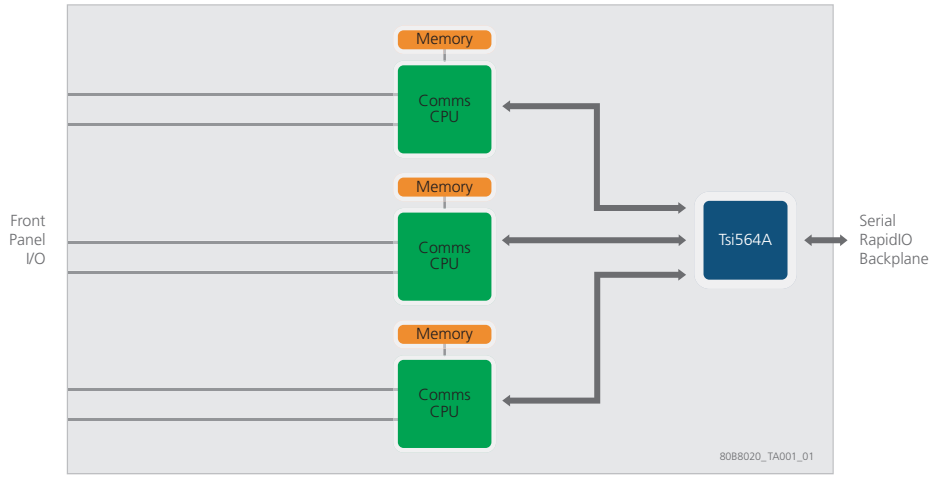
Figure 1: Tsi564A Block Diagram



1.1.1 Typical Applications

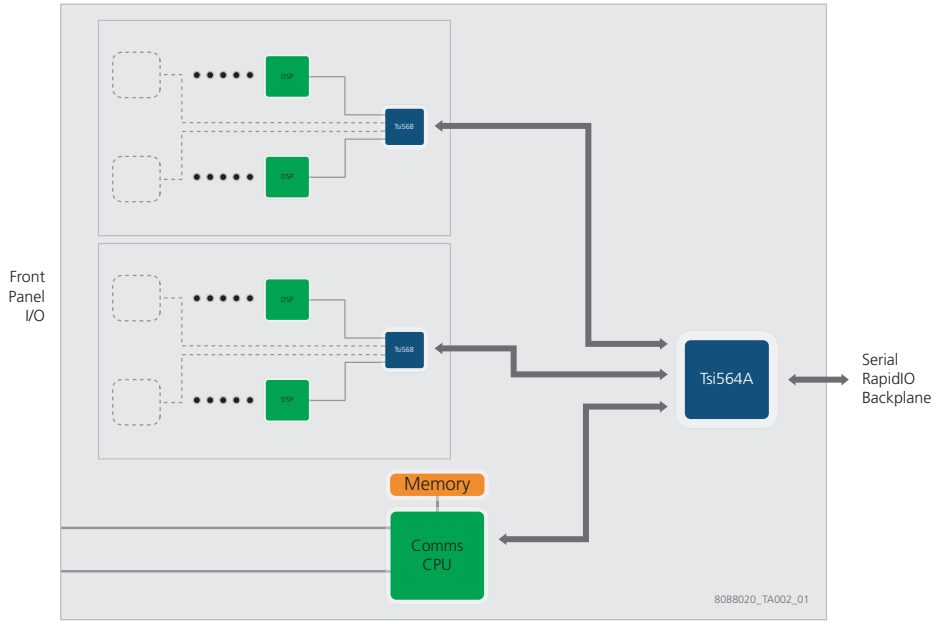
The Tsi564A can be used in many embedded communication applications. It provides chip-to-chip interconnect between I/O devices and can replace existing proprietary backplane fabrics for board-to-board interconnect which improves system cost and product time-to-market.

Figure 2: Processor Farm Mezzanine



The Tsi564A provides traffic aggregation through packet prioritization when it is used with RapidIO-enabled I/O devices. When it is in a system with multiple RapidIO-enabled processors it provides high performance peer-to-peer communication through its non-blocking switch fabric.

Figure 3: Distributed Processing Blade



1.1.2 Features

The Tsi564A contains the following features:

- *RapidIO Interconnect Specification (Revision 1.2)* compliant

Electrical Layer Serial RapidIO Features

- Up to 4 ports in 4x serial mode
- Up to 8 ports in 1x serial mode (each 4x port can be configured independently as two 1x ports)
- Operating baud rate per data lane: 1.25 Gbit/s, 2.5 Gbit/s, or 3.125Gbit/s
- Full duplex bandwidth:
  - 12.5 Gbit/s inbound and 12.5 Gbit/s outbound bandwidth at 3.125 GHz for a port configured for 4x mode[1]
  - 3.125 Gbit/s inbound and 3.125 Gbit/s outbound bandwidth at 3.125 GHz for a port configured for 1x mode[2]

1. Usable data rate is 10 Gbit/s rather than 12.5 Gbit/s due to 8B/10B physical layer encoding.

- Programmable serial transmit current with pre-emphasis equalization
- Loopback support for system testing
- Hot-insertion capable I/Os and hardware support
- Per-port power down modes to reduce power consumption

Transport Layer RapidIO Features

- Dedicated destination ID lookup table per port, used to direct packets through the switch
- Supports 512 destination IDs per lookup table
- Error management capability
- Reset-system interrupt support
- Debug packet generation in debug mode

Other Device Interfaces

- Master-only I^2C port, supports up to eight EEPROMs
- IEEE 1149.1 boundary scan, with register access

Internal switching fabric (ISF)

- Full-duplex, line rate, non-blocking switching fabric
- Prevents head-of-line blocking on each port
- Eight packet buffers per ingress port
- Eight packet buffers per egress port

Register Access

- Registers can be accessed from any RapidIO interface and the JTAG interfaces
- Optionally loads default configuration from ROMs during boot-up, through I^2C
- Supports one outstanding maintenance transaction per port
- Supports 32-bit wide (4 byte) register access

2. Usable data rate is 2.5 Gbit/s rather than 3.125 Gbit/s due to 8B/10B physical layer encoding.

1.2 Serial RapidIO Interface

The Tsi564A provides high-performance serial RapidIO interfaces that are used to provide connectivity for control plane and data plane applications. All RapidIO interfaces are compliant with the *RapidIO Interconnect Specification (Revision 1.2)*.

This section describes the transport layer features common to all Tsi564A RapidIO Interfaces. For information on the electrical layer characteristics of the serial RapidIO ports see "Serial RapidIO Electrical Interface" on page 47.

The RapidIO interface has the following capabilities:

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

1.2.1 Features

The following features are supported:

- Up to four 4x-mode or up to eight 1x-mode serial RapidIO ports operating at up to 3.125 Gbit/s.
- Per-port destination ID lookup table, used to direct packets through the switch
- RapidIO error management extensions described in *RapidIO Interconnect Specification (Revision 1.2) Part 8*, including both hardware and software error recovery
- Both cut-through and store-and-forward modes for performance tuning
- Debug packet generation and capture

1.2.2 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 see "Serial RapidIO Electrical Interface" on page 47). Received packets have their integrity verified by means of a Cyclic Redundancy Code (CRC). Once the packet's integrity has been verified, the destination ID of the packet is used to access the routing lookup table to determine what port the packet should be forwarded to. The packet is buffered by the Internal Switch Fabric (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 from the ISF. The integrity of packets forwarded through the ISF is retained by sending the CRC code received with the packet.

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.

1.2.3 Maintenance Requests

Maintenance requests are the only packet type whose forwarding is dependent on the hop count value. 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 look up table.

1.2.4 Control Symbols

Control symbols received from the Serial MAC have their CRC validated, and their field values checked. If either the CRC is incorrect or the control symbol field values are incorrect, the control symbol is discarded. Otherwise, the control symbol is used by the port for purposes of packet management in the transmit port or link maintenance.

1.3 Serial RapidIO Electrical Interface

The logic block of the Tsi564A that supports the electrical interface is called the Media Access Controller (MAC). The Tsi564A has four MACs and has 8 Serial RapidIO ports. The 8 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.

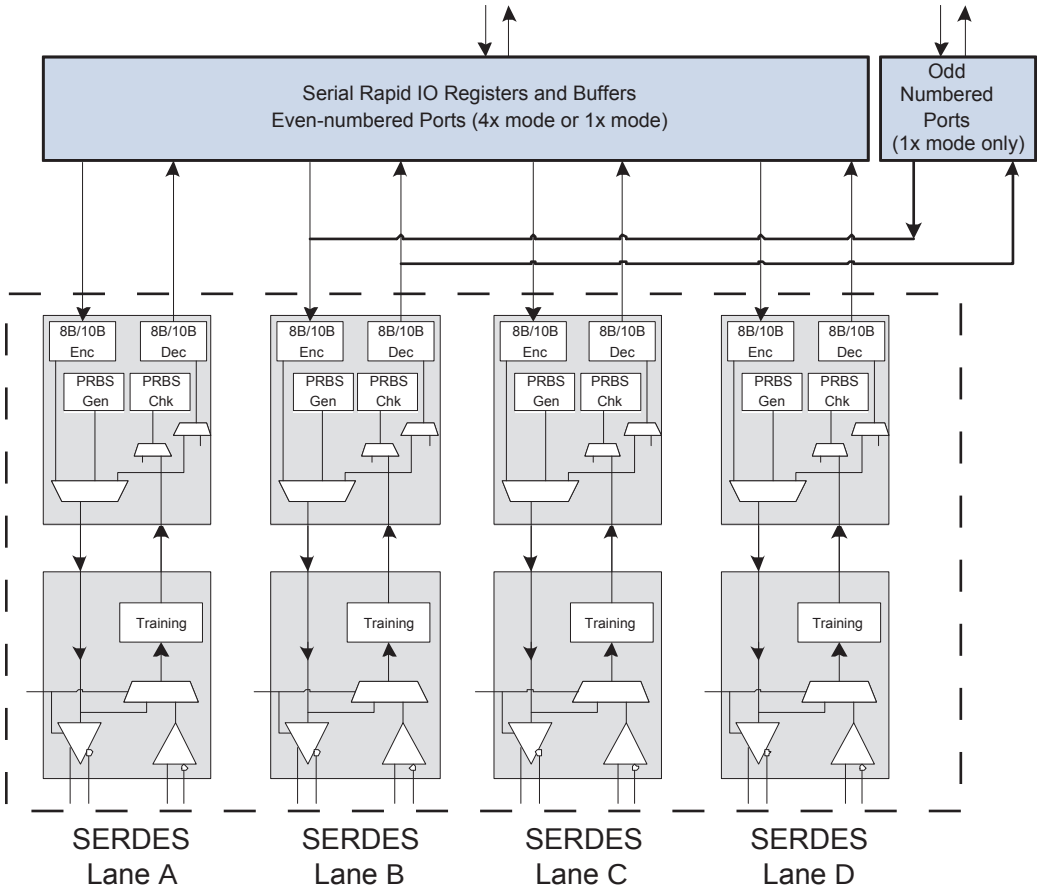
The even and odd number ports have different capabilities. Even numbered ports can operate in either 4x or 1x mode, while odd numbered ports can only operate in 1x mode. When the even numbered port is operating in 4x mode, it has control over all four differential pairs. In 4x mode, the default state of the odd numbered port is powered on. 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 is enabled to also operate in 1x mode.

When a MAC is operating in 4x mode, all registers in the MAC are accessible but the 1x port does not have access to the physical layer.

Each port has flexible testing features including loop back modes and bit error rate test support.

The Tsi564A MAC block is shown Figure 4.

Figure 4: Tsi564A Serial MAC Block Diagram



The serial RapidIO ports include the following features:

- Up to 4 ports in 4x serial mode
- Up to 8 ports in 1x serial mode (each 4x port can be configured as two 1x ports)
- Operating baud rate per data lane: 1.25 Gbit/s, 2.5 Gbit/s, or 3.125 Gbit/s
  - 12.5 Gbit/s inbound and 12.5 Gbit/s outbound bandwidth at 3.125 GHz for a port configured for 4x mode[1]
  - 3.125 Gbit/s inbound and 3.125 Gbit/s outbound bandwidth at 3.125 GHz for a port configured for 1x mode[2]
- Programmable serial transmit current with pre-emphasis equalization
- Serial loop back with a built-in testability

1. Usable data rate is 10 Gbit/s.
2. Usable data rate is 2.5 Gbit/s.

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

1.4 Internal Switching Fabric (ISF)

The Internal Switching Fabric (ISF) is the crossbar-switching matrix at the core of the Tsi564A. It transfers packets from ingress ports to egress ports and prioritizes traffic based on the RapidIO priority associated with packet and port congestion.

The ISF has the following features:

- Full-duplex, 8-port, line rate, non-blocking, crossbar-based switching fabric
- 10 Gbit/s fabric ports allow up to 10x internal speedup

The maximum bandwidth stated above is based on the maximum allowable ISF clock speed of 156.25 MHz, and summing the bandwidth for all ingress and egress ports.

- Manages head-of-line blocking on each port
- Buffers hold eight packets per ingress RapidIO port
- Buffers hold eight packets per egress RapidIO port
- Cut-through and store-and-forward switching of variable-length packets

1.5 I^2C Interface

The Tsi564A has a single, master-only, I^2C compatible interface that supports up to eight I^2C slave devices. The interface is used to load the post-reset register configuration from external EEPROMs.

The primary function of the Tsi564A's I^2C Interface is to provide the default configuration after a system boot. The Tsi564A retrieves configuration data from an I^2C-compatible serial EEPROM and then writes this to its own system configuration registers.

The block has the following features:

- I^2C controller is master only with respect to the I^2C bus
- Master register port for the system boot procedure
- General purpose I^2C interface operated through four 32-bit internal control/data registers
- Standard mode I^2C interface, up to 100 KHz operation

- Supports up to 2K byte of address space in 8 EEPROMs and up to 255 address/data pairs for register configuration during system boot.

1.6 JTAG Interface

The JTAG Interface includes dedicated user-accessible test logic that is compliant with the *IEEE1149.1 Standard Test Access Port and Boundary Scan Architecture*. The IEEE 1149.1 compliant JTAG interface is used for device testing, boundary scan, and to read and write registers.

The Joint Test Action Group (JTAG) created the boundary-scan testing standard (documented in the *IEEE 1149.1 Standard*) for testing printed circuit boards (PCBs).

The boundary-scan approach involves designing boundary-scan circuitry into the integrated circuit. PCBs populated with 1149.1 compliant devices can be tested for connectivity, correct device orientation, correct device location, and device identification. All the pins on compliant devices can be controlled and observed using (typically) five pins that are routed to the board edge connector. Board designers can develop a standard test for all 1149.1 compliant devices regardless of device manufacturer, package type, technology, or device speed.

The SerDes pins are not on the boundary scan chain.

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 Tsi564A's configuration registers
- Supports the following instruction opcodes:
  - Sample/Preload
  - Extest
  - Bypass
  - Hi-Z
  - IDCODE

— Clamp

— User data select

2. RapidIO Interface

This chapter describes the serial RapidIO interface of the Tsi564A. It includes the following information:

2.1 Overview

The Tsi564A provides high-performance serial RapidIO interfaces that are used to provide connectivity for control plane and data plane applications. All RapidIO interfaces are compliant with the *RapidIO Interconnect Specification (Revision 1.2)*.

This section describes the transport layer features common to all Tsi564A RapidIO Interfaces. For information on the electrical layer characteristics of the serial RapidIO ports see "Serial RapidIO Electrical Interface" on page 47.

The RapidIO interface has the following capabilities:

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

2.2 Transaction Flow

The Tsi564A receives a RapidIO packet on one of its serial RapidIO interfaces. After performing integrity checks, such as validating a CRC, the interface logic locates the destination identifier in the packet. The Tsi564A uses this information to determine to which egress port the packet must be sent. It consults a user-configurable lookup table, which maps destination identifiers into egress port numbers.

The serial port transfers the packet to the internal switching fabric (ISF) where it is buffered and delivered to an egress port. The 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 ISF manages head-of-line blocking, which means that when a packet cannot be moved to an egress port (because multiple ingress ports are trying to send to the same egress port for example), the ISF selects another packet to service from the same ingress port.

In cut-through mode, the egress port transmits the packet as it receives it from the ISF. The egress port stores the packet so that it can be retransmitted should an error occur.

Packet transfers between RapidIO ports within the Tsi564A can be *cut-through*. This means that when there is no switching congestion, packet data is moved from one point to another before the packet has arrived entirely. For example, the start of a packet can be sent out of the egress port before the end of the packet has arrived at the ingress port. RapidIO provides a *stomp* mechanism to abort partially transmitted packets that are later determined to have data integrity errors or similar errors. If the Tsi564A finds that a packet that is being cut-through has an error it sends a stomp control symbol to notify the receiver that the packet was in error and all received data should be dropped.

2.3 Lookup Tables

Lookup tables (LUTs) are used to 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 switch. The port does this by using the destination ID as an index to a lookup table containing user-defined egress port numbers.

Each RapidIO Interface has its own, uniquely configurable lookup table. Configuration and maintenance of the lookup tables is compliant with the *RapidIO Interconnect Specification (Revision 1.2)*. These registers write all lookup tables simultaneously. Additionally, each port's look up table can be accessed using device specific registers.

The LUTs support flat mode. Flat mode supports destination IDs in the range of 0 to 511, with a default port for destination IDs outside this range.

2.3.1 Flat Mode Lookup Table 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 "RIO Route LUT Attributes CSR" on page 150.

2.3.2 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 1). 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.

For test purposes, it is possible to corrupt the parity of an entry by setting the PAR_INVERT field in the RIO Port x Route Config DestID CSR to one and writing one or more LUT entries. This is useful for testing software and verifying Tsi564A functionality. All LUT entries written while this bit is one are corrupt. Each time an afflicted LUT entry is read, a parity error is detected.

2.3.3 Lookup Table Error Summary

Table 1 summarizes error conditions and resulting behaviors associated with the LUTs.

Table 1: Error Summary

Event	Behavior
Packet routed to a shut down port	Packet discarded and no record of packet is kept.
Packet routed to disabled port	Packet discarded and no record of packet is kept.
Packet routed to unconnected port	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 recorded in error capture registers and the packet is discarded • ,IMP_SPEC_ERR bit is set in the "RIO Port x Error Detect CSR" on page 172 • Port write can be generated (if enabled) • Interrupt can be generated (if enabled).

Table 1: Error Summary *(Continued)*

Event	Behavior
Parity error on LUT entry	Packet recorded in error capture registers, packet discarded • ,IMP_SPEC_ERR bit is set in the "RIO Port x Error Detect CSR" on page 172 • Port write is generated • Interrupt is generated.
Writes to LUT entries through the broadcast LUT registers to unconnected or shutdown ports	Silently ignored
Access to LUT with a destination ID which exceeds LUT size	Writes silently ignored, reads return 0.

2.3.4 Lookup Table Entry States

A lookup table entry can be in one of three states: mapped, unmapped, and parity error. After any reset all lookup table entries are undefined (an unknown state). All lookup table entries must be programmed to a known value after reset to achieve predictable operation. A lookup table entry that routes packets to a port that exists within the Tsi564A is *mapped*. A lookup table entry that routes packets to a port that does not exist with the Tsi564A *unmapped*. When a lookup table entry's parity is incorrect, the lookup table entry is in a parity error state. Table 2 shows the possible lookup table states.

Table 2: Lookup Table States

Lookup Table Entry State	Action on Packet Arrival
Mapped	Packet is routed to the specified output port
Unmapped port value	Default port is used for routing the packet. **Note:** The default port is defined in RIO Route LUT Attributes CSR (see Section 10.4.13 on page 150).

Table 2: Lookup Table States

Lookup Table Entry State	Action on Packet Arrival
Unmapped default port value	• Packet recorded in error capture registers • Packet discarded • IMP_SPEC_ERR bit is set in the "RIO Port x Error Detect CSR" on page 172 • Port write can be generated (if enabled) • Interrupt can be generated (if enabled).
Parity Error	• Packet recorded in error capture registers • Packet is discarded • IMP_SPEC_ERR bit is set in the "RIO Port x Error Detect CSR" on page 172 • Port write is generated • Interrupt is generated.
UnProgrammed (Undefined)	Non-deterministic operation may occur.

When a port value for a lookup table entry is unmapped, the default port is used for routing the packet as defined in RIO Route LUT Attributes CSR register. If the default port value is unmapped, packets routed using the default port value are discarded and the IMP_SPEC_ERR bit is set in the "RIO Port x Error Detect CSR" on page 172.

Lookup table entries can be programmed through the standard RapidIO compliant interface or through a IDT-specific interfaceTsi564A specific set of registers which can auto-increment entries reducing the number of maintenance cycles required. The Tsi564A specific set of registers include the "RIO Port x Route Config DestID CSR" on page 186 and the "RIO Port x Route Config Output Port CSR" on page 187.

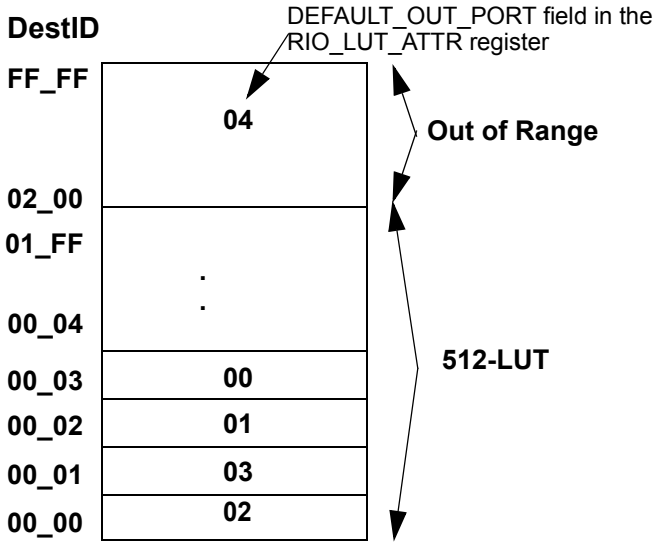
2.3.5 Configuring the Lookup Tables

Each of the ports on the Tsi564A 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.

LUT entries can be reprogrammed at any time during normal system operation. If deterministic behavior for a routing change is required, software must control when packets use the LUT entries that are being changed.

Figure 5 shows an example of a lookup table. In this example, a destination ID of 0x0002, or 0x02, is routed by the 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: Lookup Table Example



2.3.5.1 Registers Used in Lookup Table Configuration

The Tsi564A's RapidIO interface is compliant with the *RapidIO Interconnect Specification (Revision 1.2)*. The following RapidIO standard registers are used by the Tsi564A for programming the lookup tables:

- RIO Route Config DestID CSR
- RIO Route Config Output Port CSR
- RIO Route LUT Size CAR
- RIO Route LUT Attributes CSR

2.3.5.2 Routing Destination IDs Based on the Lookup Table Entries

The Tsi564A routes packets with the varying destination ID possibilities in the following way:

- If a packet's destination ID is in the range of 0 - 511, and its lookup table entry has been programmed, the packet is routed out the specified port.
- If the table entry is programmed to the unmapped state (output port is set to 0xFF), the packet is routed using the default egress port defined in the "RIO Route LUT Attributes CSR" on page 150.

- If a packet's destination ID is greater than 511, the packet is also routed using the default egress port defined in the "RIO Route LUT Attributes CSR" on page 150.
- If the default egress port in RIO Route LUT Attributes CSR is unmapped, the Tsi564A discards the packet and sets the IMP_SPEC_ERR bit in the "RIO Port x Error Detect CSR" on page 172.

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 the value 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.

- Packets with 16-bit destination IDs from 0x0000 through 0x00FF are routed using the same lookup table entries as those for the 8-bit destination IDs 0x00 through 0xFF.

2.3.5.3 Lookup Table Configuration Examples

The Tsi564A lookup tables can be configured through an external EEPROM or through software maintenance writes to the Tsi564A 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 is used to program the lookup tables:

1. Write the destination ID to be configured or queried to the RIO Route Config DestID CSR register.
2. Read the RIO Route Config Output Port CSR register 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 entry, perform the following steps:

1. Write to the RIO Route Config DestID CSR with a value of 0x00000098. This makes the Destination ID 0x98.
2. Write to the RIO Route Config 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 entry, perform the following steps:

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

In this example, if a further write to RIO Route Config Output Port CSR (offset 0x11274) 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 RIO Route Config DestID CSR with a value of 0x00000054. This programs the Destination ID 0x54.
2. Read to the value in RIO Route Config 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.

2.4 Maintenance Packets

The Tsi564A treats maintenance packets differently than other packets. In a system it is possible that the Tsi564A is 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 look up table, and other values programmed in the intervening devices.

If a maintenance packet has a hop count greater than zero, the Tsi564A decrements the hop count, recalculates the CRC, and routes the packet out the port selected in the look up tables.

If a maintenance read or maintenance write request packet has a hop count of 0, the port must process the maintenance request and send 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 composed by the Tsi564A 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 request is being processed will be retried by the RapidIO port.

The Tsi564A supports 4 byte maintenance requests only. Maintenance requests for more than 4 bytes, as well as maintenance packets which are not read or write requests, are dropped and an error is noted in the IMP_SPEC_ERR bit of the "RIO Port x Error Detect CSR" on page 172. Examples of maintenance packets which are dropped are maintenance response and port-write packets received with a hop count of 0.

2.5 Reset Control Symbol Processing

One of the functions which can be performed by control symbols is requesting that the link partner reset itself. The Tsi564A can generate link-request/reset control symbols using the standard RapidIO registers defined for the purpose.

The Tsi564A generates four link-request/reset control symbols with only one register access to the RIO Serial Port x Link Maintenance Request CSR (see "RIO Serial Port x Link Maintenance Request CSR" on page 155).

The reset control symbol entirely resets the Tsi564A.

For more information on reset control symbol handling, see "Resets" on page 108.

2.6 Data Integrity Checking

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

2.6.1 Packet Data Integrity Checking

Packets have two locations where CRC may occur. The first location is 80 bytes into the packet. The second location is at the end of the packet. Packets 80 bytes or fewer in size have only one CRC, packets larger than 80 bytes have two 16 bit CRC codes. With one exception (maintenance transactions), the Tsi564A does not (re)compute CRC codes for packets. The CRC code is forwarded with the packet across the ISF, and the packet is transmitted with the same CRC code it was received with. This ensures that packet corruption within the Tsi564A is detected.

Maintenance packets have a hop count field, covered by CRC, which must be altered by the Tsi564A if the packet is to be forwarded. Therefore the CRC is recomputed for maintenance packets for each link they traverse.

2.6.2 Control Symbol Data Integrity Checking

Control symbols are made up of 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.

2.7 Error Management

The Tsi564A supports the Software Assisted Error Recovery registers as defined by the RapidIO specification. For the complete list of registers supported for Software Assisted Error Recovery, see “RapidIO Physical Layer Registers” on page 151.

2.8 Hot Insertion and Hot Extraction

Hot insertion and hot extraction functionality enables reliable systems to safely add, remove, and replace components while the system continues to operate.

The system host can use the Tsi564A’s capability to restrict the access of a newly inserted component to prevent a faulty component from negatively affecting the system. The following bit fields in “RIO Serial Port x Control CSR” on page 164 are used to control access to the system:

- PORT_DIS: Ports on which hot insertion/extraction events can occur must have the PORT_DIS bit set to 0, to allow link initialization to take place.
- PORT_LOCKOUT: When the PORT_LOCKOUT bit is set, only link-request control symbols may be exchanged. When the PORT_LOCKOUT bit is cleared, access is controlled by OUTPUT_PORT_ENABLE and INPUT_PORT_ENABLE.
- OUTPUT_PORT_ENABLE: Controls whether packets other than maintenance requests/responses may be sent by the Tsi564A.
- INPUT_PORT_ENABLE: Controls whether packets other than maintenance requests/responses may be received by the Tsi564A.

The lookup tables (LUTs) are used to ensure that no traffic is being routed to the component being inserted/removed. For more information on lookup table functionality, see “Lookup Tables” on page 28.

The system host can be notified by the Tsi564A when a component has been inserted through the use of the port write mechanism. The registers and bit fields relevant to port write transmission for purposes of hot insertion and hot extraction are:

- RIO Serial Port X Control CSR and the PORT_LOCKOUT bit: When the PORT_LOCKOUT bit is set, and the link has successfully initialized, a port write can be sent periodically to notify the system host that a new component has been added to the system.
- RIO Port X Control Independent register and the LINK_INIT_NOTIFICATION bit: When the LINK_INIT_NOTIFICATION bit is set, the PORT_LOCKOUT bit is set, and the link as successfully initialized, port writes are sent and interrupts are asserted to notify a system host that a new component has been inserted.

- RIO Port x Interrupt Status register and the LINK_INIT_NOTIFICATIONbit: When the PORT_LOCKOUT bit is set, this bit indicates that the link has successfully initialized.
- RIO Port x Interrupt Generate register and the LINK_INIT_NOTIFICATION bit: This bit causes the LINK_INIT_NOTIFICATION bit of the RIO Port x Interrupt Status Register to be set. This function is useful for testing software.

The remainder of the hot insertion/extraction section describes the use of Tsi564A functionality to safely insert and extract components.

2.8.1 Hot Insertion

Tsi564A 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 must be set to allow the LINK_INIT_NOTIFICATION bit in the RIO 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 RIO Port x Interrupt Status Register, or of setting the LINK_INIT_NOTIFICATION bit in the “RIO Port x Control Independent Register” on page 188 to assert an interrupt or send port write transactions.

The OVER_PWIDTH field of the “RIO Serial Port x Control CSR” on page 164 should be set to 0 (no overrides) to make sure the port detects the insertion of a link partner.

Once a link partner has been detected, it is up to maintenance software to ensure that the port is configured in the correct mode for the link partner (either 4x/1x or 1x+1x) before completing the remainder of the hot insertion procedure. For information on how to change the mode of the ports which share a MAC, see “Changing Port Mode” on page 51.

Once the system host has been notified that a new component has been inserted, the LINK_INIT_NOTIFICATION bit should be cleared in the RIO Port x Interrupt Status Register to stop the assertion of interrupts and/or transmission of port writes.

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 rest of the system. The OUTPUT_PORT_ENABLE and INPUT_PORT_ENABLE 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 the RIO 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 programmed to 0, since the component just inserted has been reset. Similarly, the OUTBOUND value for the component that was just inserted must be programmed to match the INBOUND value of the Tsi564A's port, contained in the RIO Serial Port x Local ackID Status CSR. Just as with a controlled reset of a link partner (see "Generating a RapidIO Reset Request to a Peer Device" on page 110), the writes of the two OUTBOUND values must occur in the order given (Tsi564A followed by the link partner).

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

2.8.1.1 Link Partner and Unsupported Error Recovery

In the event that the link partner does not support the software assisted error recovery registers, the equivalent of the "RIO Serial Port x Local AckID Status CSR" on page 159 will not exist in the link partner. Since it will be impossible to set the link partner's OUTBOUND value in this case, the Tsi564A INBOUND value must become zero. The method for setting the Tsi564A INBOUND value to 0 is to power down and power up the Tsi564A port using the PWDN_X4 or PWDN_X1 bit in the SRIO MAC x Clock Selection Register. For the proper use of the PWDN_X4 and PWDN_X1 bits, see Table 11 on page 55.

2.8.2 Hot Extraction

Tsi564A ports on which a hot extraction event occurs should not have any transactions flowing through them in preparation for the extraction. The LUT entries for all ports in the Tsi564A must be configured to not route any packets to the port on which the hot extraction occurs. The PORT_LOCKOUT bit must be set on the port on which the hot extraction event occurs in order to drop all packets arriving from the 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.

2.8.3 Hot Extraction System Notification

System designers may require confirmation of when a component has been extracted.There are a number of confirmation methods supported by the Tsi564A.

2.8.3.1 Polling

The system can poll the PORT_OK and PORT_UNINIT bits in the "RIO Serial Port x Control CSR" on page 164 for evidence that the link partner is no longer present.

2.8.3.2 Interrupts and Port Writes

This design is useful in applications that need to monitor the continued presence of a component that is to be extracted.

Interrupts and/or port writes can be implemented as part of the hot insertion and hot extraction system. For example, while the PORT_LOCKOUT bit is set and the LINK_INIT_NOTIFICATION bit in the "RIO Port x Control Independent Register" on page 188 is set, port write transactions are sent and interrupts are asserted until the component is extracted.

2.8.3.3 Link Errors

This design is useful in applications that are only concerned with when a component has been extracted.

Another notification implementation makes use of the 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 the "RIO Port x Control Independent Register" on page 188 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.

2.9 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. The remainder of this section deals with LOLS recovery related to high error rates on a link. For a discussion of LOLS handling due to reset of a link partner, see "Generating a RapidIO Reset Request to a Peer Device" on page 110. For a discussion of LOLS handling due to hot insertion or hot extraction of a link partner, see "Hot Insertion and Hot Extraction" on page 37.

When the Tsi564A 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 1.2)*.

When a Tsi564A port detects LOLS, it restarts its synchronization state machine and stops its Time-out Interval Value (TVAL) timer for expected packet and control-symbol acknowledgements. The Tsi564A port is in input-error stopped state due to errors seen on the link. For more information on the duration of this timer, see the TVAL field in the "RIO Switch Port Link Time Out Control CSR" on page 153.

Once synchronization is re-acquired, the Tsi564A 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 treats the LOLS event as a temporary interruption that is completely 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 Tsi564A are not acknowledged because the port is in the input-error stopped state. The link partner times out waiting for a packet acknowledge control symbol, and enters the output-error stopped state. To recover, the link partner sends a link-request/input-status control symbol to the Tsi564A port. This clears the input-error stopped state on the Tsi564A.

The Tsi564A 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 Tsi564A.

2.9.1 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 a uncontrolled extraction of the link partner, and a hardware fault on the link partner. Packets will continue to be directed to the non-functional (dead) link, but will not be able to make forward progress. This has the potential to eventually block every traffic path in the system.

To allow systems to robustly deal with dead links, the Tsi564A has a *dead link timer* feature. This is a proprietary function that is outside of the *RapidIO Specification*. The DLT_EN and DLT_THRESH field (in the "SRIO MAC x SERDES Configuration — Channel 0" on page 196) are used to enable/disable the dead link timer, and specify the duration of the dead link timer, for each MAC. Because DLT_EN and DLT_THRESH apply to each MAC, each pair of ports shares the dead link timer duration and enable/disable control values.

Each port has a dead link timer. If the dead link timer expires, indicating that the link is no longer able to transmit or receive, then the port begins to act in a manner designed to remove 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 ISF are accepted and dropped. Packets received by the port from RapidIO may still be forwarded to the ISF.

The dead link timer functionality is controlled by the register "SRIO MAC x SERDES Configuration — Channel 0" on page 196. The dead link timer can be disabled for the pair of ports sharing a MAC by setting the DLT_EN bit to 0. The range of the timer can be set by writing the DLT_THRESH field.

2.10 Debug Packet and Control Symbol Generator

In Debug mode, debug packets are inserted in the inbound flow. A packet generated by the debug packet generation feature is routed through the ISF, and can be transmitted by any port on the Txi56x. Selection of the port the packet is transmitted on is the function of the Lookup Tables (LUTs). The LUT on the port where the packet is generated must be set up correctly in order for the packet to be transmitted on the intended port. For more information on programming the lookup table, see "Lookup Tables" on page 28.

Generated control symbols are sent out the port on which they are generated.

The debug packets must be 32 bits aligned and 5x32 bits in size.

The contents of all the fields of the packet, including the CRC field, are defined by the registers described in Table 3.

Table 3: Debug Packet Content

Data	Word 0	Word 1
0x0	RIO Port x Debug 0	RIO Port x Packet Debug 1
0x8	RIO Port x Packet Debug 2	RIO Port x Packet Debug 3
0x10	RIO Port x Packet Debug 4	-

2.10.1 Debug Operations

The debug mode supports the following operations:

- Any type of packet — Up to 20 bytes in size (including CRC), the first 5 bits are overwritten by ackID
- Embedded and non-embedded control symbols

2.10.2 Debug Registers

The following registers control operations in Debug mode:

- "RIO Port x Control Independent Register" on page 188
  - — DEBUG: Defines the Debug mode of operation. In the Debug mode the debug registers are unlocked and used for writing the content of the debug packet.
  - — CS_EMBED: Forces logic to insert a control symbol into the debug packet. The control symbol is defined in the register in RIO Serial Port x Control Symbol Transmit register.

— SEND_DBG_PKT: Send debug packet bit which triggers the start of debug packet generation.

- Capture registers defined in Table 3 on page 42
  - These registers are used to capture error information in normal mode. In Debug mode they are unlocked and used to write the packet content.
- "RIO Port x Control Symbol Transmit" on page 191)
  - This register is used to define the 24 bits of the control symbol fields.

2.10.2.1 How to Use Debug Mode

The first step in all operations is to set up debug mode by setting the DEBUG bit to 1 in the "RIO Port x Control Independent Register" on page 188. After the bit is set, the following steps show how the debug functionality operates:

1. Send a packet
   - Write the contents of the packet into the registers listed in Table 3 on page 42. The content contains the information shown in "Packet Content" on page 44.
     - The CRC for maintenance packets is generated by the Tsi564A automatically. All other packets require software to compute the CRC to be sent with the packet.
   - Write 1 to the SEND_DBG_PKT field in the "RIO Port x Control Independent Register" on page 188
2. Send a control symbol
   - Write the contents of the control symbol into the registers listed in Table 3 on page 42.
   - Write the contents of the control symbol by setting CS_EMB to 0 in the "RIO Port x Control Symbol Transmit" on page 191. The content includes information of stype0, stype1 and other fields of a control symbol.

For information on the construction of a control symbol, see Table 34 on page 245.

   - The CRC is generated by internal logic.
   - The control symbol is sent automatically.
3. Send a packet with an embedded control symbol
   - Write the contents of the packet in registers listed in Table 3 on page 42.
   - Enable the insertion of a control symbol by setting CS_EMBED to 1 in the "RIO Port x Control Symbol Transmit" on page 191.

— Write 1 to the SEND_DBG_PKT field in the "RIO Port x Control Independent Register" on page 188.

— The control symbol is embedded in the beginning of packet after the Start-Of-packet control symbol

When a debug packet is generated, the recipient of the packet can generate a response. The system must be initialized to allow for the handling of this response. If handling the response is not possible, the response can be sent to a destination ID that is unmapped. Doing so, while the default port destination ID is also unmapped, means the response packet will be discarded. For more information on the look up tables, and other alternatives for routing responses, see "Lookup Tables" on page 28.

2.10.2.2 Packet Content

The following tables show the content of specific packet types.

Table 4 shows the construction of a nread request packet.

Table 4: Nread Request Packet Construction (Base Size: 12bytes)

Nread Request Packet												
ackID	Rsvd	CRF	Priority	TT 2	Ftype	destID	source ID	Trans type	Rd/ wrsize	source TID	address + wdptr+ xamsbs	CRC 16
5 bits	2 bits	1 bit	2 bits	2 bits	4 bits	8 or 16 bits	8 or 16 bits	4 bits	4 bits	8 bits	32 bits	16 bits

Table 5 shows the construction of a nwrite request packet.

Table 5: Nwrite Request Packet Construction (Base Size: 16bytes)

Nwrite Request Packet													
ackID	Rsvd	CRF	Pri	TT 2	Ftype	destID	source ID	Trans type	Rd/wrs ize	source TID	address + wdptr/ xamsbs	Pay load	CRC 16
5 bits	2 bits	1 bit	2 bits	2 bits	4 bits	8 or 16 bits	8 or 16 bits	4 bits	4 bits	8 bits	32 bits	32 bits	16 bits

Table 6 shows the construction of a maintenance read packet.

Table 6: Maintenance Read Packet Construction (Base Size: 11 bytes)

Maintenance Read packet													
ackID	Rsvd	CRF	pri	TT 2	ftype	destID	source ID	trans type	Rd/ wrsize	source TID	hop count	address+ wdptr+ rsvd	CRC 16
5 bits	2 bits	1 bit	2 bits	2 bits	4 bits	8 or 16 bits	8 or 16 bits	4 bits	4 bits	8 bits	8 bits	24 bits	16 bits

Table 7 shows the construction of a maintenance write packet.

Table 7: Maintenance Write Packet Construction (Base Size:15 bytes)

Maintenance Write packet														
ackID	Rsvd	CRF	Pri	TT 2	Ftype	dest ID	sourceID	Trans type	Rd/ wrsize	source TID	hop count	address+ wdptr+ rsvd	Pay load	CRC 16
5 bits	2 bits	1 bit	2 bits	2 bits	4 bits	8 or 16 bits	8 or 16 bits	4 bits	4 bits	8 bits	8 bits	24 bits	32 bits	16 bits

3. Serial RapidIO Electrical Interface

This chapter describes the IDT-specific electrical layer features of the Tsi564A Serial RapidIO Electrical Interface. See Section 2 on page 27 for a description of the standards-defined RapidIO features common to all RapidIO ports.

This chapter includes the following information:

3.1 Overview

The logic block of the Tsi564A that supports the electrical interface is called the Media Access Controller (MAC). The Tsi564A has four MACs and has 8 Serial RapidIO ports. The 8 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.

The 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. 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 is enabled to also operate in 1x mode.

When a MAC is operating in 4x mode, all registers in the MAC are accessible but the 1x port does not have access to the physical layer.

Each port has flexible testing features including loop back modes and bit error rate test support.

3.2 RapidIO Port Numbering

The RapidIO ports on the Tsi564A are numbered sequentially from 0 to 7. Table 8 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 8: Tsi564A 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)	1x or 4x
7	Serial Port 7 (SP7)	1x

3.3 Port Aggregation: 1x and 4x Link Modes

The serial RapidIO MAC provides the PMA/PCS encoding/decoding layers, as well as the RapidIO physical, transport, and logical layer functionality required in a switch device in order to support packet routing and maintenance transactions.

The MACs are numbered in even numbers beginning at zero so, serial ports 0 and 1 use MAC 0, ports 2 and 3 use MAC 2, etc. 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 Tsi564A MAC has an external pin called SPx_MODESEL. This pin can be pulled high or low to configure the MAC for either 1x + 1x mode or 4x + 0x mode (see "Signals and Pinlist" on page 115 for information on these pins). 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 reflected in the PORT_WIDTH field of the "RIO Serial Port x Control CSR" on page 164.

If the SPx_MODESEL pin values are changed after they have been sampled at reset release, the port's operation is not affected.

The actual port width currently in use can be different from the pin-selected width. An even port with the capability to run in either 1x or 4x mode port can downgrade 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 more information, see "RIO Serial Port x Control CSR" on page 164 for status and control fields for port width and "4x Configuration" on page 50 for downgraded port configuration.

3.3.1 1x + 1x Configuration

When the even-numbered port in a Tsi564A MAC is configured to operate in 1x mode, the odd-numbered port in a MAC can also be used in 1x mode. In this configuration, the even-numbered port always uses SerDes lane A and the odd-numbered port always uses SerDes lane B. The remaining two SerDes lanes may be powered down to conserve power. For information on powering down a port, see "Port Power Down" on page 53.

The two ports that share the same MAC also share the same transmit clock, which means the two ports must have the same bit rate. The initial clock rate is selected by the global power-up option for all ports. To select the bit rate independent of the SPEED[1:0] pins write the SCLK_SEL field in the "SRIO MAC x Clock Selection Register" on page 211, as described in "Port Power Down" on page 53.

3.3.2 4x Configuration

When the even-numbered port in a Tsi564A MAC is configured to operate in 4x mode, the odd-numbered port in a MAC cannot be used. To save power, the odd-numbered port can be powered down (see "Port Power Down" on page 53).

The unusable, odd-numbered port is still a part of the Tsi564A'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 Tsi564A's behavior is undefined. For more information on register behavior under power down conditions, see "Port Power Down" on page 53.

The even-numbered port configured for 4x mode follows the link-width negotiation rules outlined in the *RapidIO Interconnect Specification (Revision 1.2)*. 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.

System software can force a downgrade in port mode by writing the OVER_PWIDTH field in the "RIO Serial Port x Control CSR" on page 164 on either the Tsi564A or in its link partner. The current operating link width is available in the INIT_PWDITH field in the "RIO Serial Port x Control CSR" on page 164. Software may be required to manage ackID recovery for the link partner when changing the active lane between lanes A and C.

The link partner must have the capability to continue to communicate when changing the active lane between Lanes A and C. For more information on determining the capability of the link partner, see "RIO Serial Port x Control CSR" on page 164.

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 "RIO Port x Control Independent Register" on page 188.

3.3.3 Changing Port Mode

It is possible to change the mode of operation of a pair of ports ont he same MAC through the following steps:

- Power down the odd and even ports by setting the PWDN_X1 and PWDN_X4 bits in the "SRIO MAC x Clock Selection Register" on page 211 to 1.
- Change the SP(n)_MODESEL input pin value for that pair of ports to reflect the required value. This can be accomplished by the actions of an external controller, or manual intervention.
- Power up the even port and, if the port is in 1x + 1x mode, power up the odd port in the port pair by setting the PWDN_X1 and PWDN_X4 bits in the"SRIO MAC x Clock Selection Register" on page 211 to 0.
- Reprogram all registers for the ports which were just powered up. This includes those registers which support broadcast addresses in the address range 0x10000 through 0x100FF.

When Port 0 is powered down, registers accessed through the broadcast address range (0x10000 through 0x100FF) returns 0s until port 0 is powered up and its registers are reprogrammed.

3.4 Clocking

Serial RapidIO ports use source clocked transmission; the clock is embedded in the data stream using 8B/10B encoding. The Tsi564A recovers the embedded clock in the received data stream and generates a separate clock to transmit its own data.

The Tsi564A supports three different signaling rates that are generated from two different external clock sources. Table 9 shows the clock speeds. Multiplication is performed by the SERDES PLL. The division by two of the S_CLK_2 input is performed at the input. For more information on clocking, see "Clocks, Resets and Power-up Options" on page 103.

Table 9: Valid Serial RapidIO Clock Speeds

Clock Source	Clock Rate	Comments	Bit Rate	User Bandwidth (1x mode)	User Bandwidth (4x mode)
Pin S_CLK_2	250 MHz	Clock is divided by two then multiplied by 10	1.25 Gbit/s	1.0 Gbit/s	4.0 Gbit/s
Pin S_CLK_2	250 MHz	Clock is multiplied by 10	2.50 Gbit/s	2.0 Gbit/s	8.0 Gbit/s
Pin S_CLK_1	312.5 MHz	Clock is multiplied by 10	3.125 Gbit/s	2.5 Gbit/s	10 Gbit/s

A port's initial signaling rate can be one of the three options listed in Table 9. The initial signaling rate is determined by the setting of the SP_IO_SPEED[1:0] pins (see "Signals and Pinlist" on page 115). There is one pair of SP_IO_SPEED pins for the entire device, which means all serial RapidIO ports operate at the same bit rate by default.

When a pair of ports on the same MAC are both operating in 1x mode, both ports have the same rate.

Table 10: SP_IO_SPEED[1:0] Settings

Setting	SP_IO_SPEED1	SP_IO_SPEED0
S_CLK_2 divided by 2	0	0
S_CLK_2	0	1
S_CLK_1	1	0
Reserved	1	1

As required by the *RapidIO Interconnect Specification (Revision 1.2)*, receive and transmit signals must occur at the same clock speed. This means a port must always transmit at the same clock rate that it receives. Additionally, the difference in speed between the transmit and receive clocks must be within 200 parts per million.

3.4.1 Changing the Clock Speed

The following procedure changes the signaling rate of a port:

1. Set the PWDN_X4 bit in the "SRIO MAC x Clock Selection Register" on page 211 to 1
2. Select the new clock speed using the SCLK_SEL field in the "SRIO MAC x Clock Selection Register" on page 211
3. Set the PWDN_X4 bit to 0

Powering down the port to change the clock speed results in loss of per-port configuration. All SPx_xxx registers revert to their default values and must be reconfigured. Lookup tables for the port must also be reprogrammed after powering down the port.

3.5 Port Power Down

Almost any of the Tsi564A RapidIO ports can be powered down to minimize power consumption when the port is not required. The port that cannot be powered-down is Port 0 because global device registers are accessed through the port.

Port 0 must never be powered-down in the Tsi564A.

When a port is powered down, all registers that begin with 'SP' in their name in the register chapter return 0 and all writes to these registers are ignored. The exception to this rule are the SPx_ERR_STATUS (RIO Port x Error and Status CSR) and SPx_CTL registers (RIO Serial Port x Control CSR), both of which return 0x00000001 when read. Together, these values indicate that the port is an uninitialized RapidIO port.

Register values are maintained when a port is disabled. To disable a port, set PORT_DIS to 1 in the "RIO Serial Port x Control CSR" on page 164.

The following register types are read only and return zero when a port is powered-down:

- RapidIO Logical Layer and Transport Layer Registers (see Section 10.4 on page 135)
- RapidIO Physical Layer Registers (see Section 10.5 on page 151)

- RapidIO Error Management Extension Registers (see Section 10.6 on page 168)
- IDT-Specific RapidIO Registers (see Section 10.7 on page 179)

Both the RIO Port x Error and Status CSR (SPx_ERR_STATUS) and RIO Serial Port x Control CSR (SPx_CTL) registers return 0x00000001 when read instead of 0s.

When an individual port is powered down and powered up, all IDT-Specific RapidIO Registers must be reprogrammed. This includes those registers which support broadcast addresses in the address range 0x10000 through 0x100FF.

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

- Serial Port Electrical Layer Registers (see Section 10.8 on page 195)
- Internal Switching Fabric (ISF) Registers (see Section 10.9 on page 228)
- Utility Unit Registers (see Section 10.11 on page 237)

3.5.1 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 SP{0..7}_PWRDN pins.
- The default configuration provided by the pins can be altered through the SMACx_CLK_SEL registers. This occurs during boot up through an EEPROM on the I^2C bus, or during normal operation through a register write.
- The individual I/O bit lanes can be disabled even when the port logic is powered up.

3.5.2 Configuration and Operation Through Power Down

The IOs for the individual bit lanes can be powered down when they are not used. For example, two bit lanes are not used if the pair of ports in a MAC are configured for 1x + 1x mode. The SMACx_CFG_CH registers (see Section 10.8.1 on page 196) are used to power down the two lanes.

If only the IOs are powered down, the clocks to the internal logic remain active. Resources in the internal switching fabric are not released which means an I/O power down can affect service on other ports. When just the I/Os are powered down, the switching fabric is not aware that the lanes are not able to transmit data buffered to them. It is important to use I/O power down only in situations listed in Table 11.

When the port logic is powered down, the clocks to the internal logic for that port are stopped. When a port is powered down during normal operation, no packets must be routed to the port. All packets sent to the powered down port are silently discarded when the internal switching fabric attempts to deliver them to the port.The LUTs of the other ports must be reconfigured to not route packets to the powered down port or packets will be lost. This ensures that resources in the internal switching fabric are released when the port is powered down.

All valid power-down scenarios are shown in Table 11.

Table 11: Serial Port Power Down Procedure

Mode for Serial Port *n*	Mode for Serial Port *n+1*	Required Power Down Configuration
4x	n/a	• De-assert the SP*n*_PWRDN pin and/or set the SMAC*n*_CLK_SEL[PWDN_X4] register bit to 0. • To save power, assert the SP*n+1*_PWRDN pin and/or set the SMAC*n*_CLK_SEL[PWDN_X1] register bit to 1. Otherwise, Port n+1 consumes power unnecessarily.
1x	1x	• De-assert the SP*n*_PWRDN pin and/or set the SMAC*n*_CLK_SEL[PWDN_X4] register bit to 0. • De-assert the SP*n+1*_PWRDN pin and/or set the SMAC*n*_CLK_SEL[PWDN_X1] register bit to 0. • To save power, set the SMAC*n*_CFG_CH{2,3}.PWDN register field to 1. This powers down the unused SerDes lanes C and D (see Section 10.8.2 on page 200).
1x	Port Not Used	• De-assert the SP*n*_PWRDN pin and/or set the SMAC*n*_CLK_SEL[PWDN_X4] register bit to 0. • To save power, assert the SP*n+1*_PWRDN pin and/or set the SMAC*n*_CLK_SEL.PWDN_X1 register bit to 1. • And to additional save power, set the appropriate SMAC*n*_CFG_CH{1, 2,3}.PWDN register field to 1. This powers down SerDes lanes B, C, and D (see Section 10.8.2 on page 200).
Port Not Used	Port Not Used	• To save power, assert the SP*n*_PWRDN pin and/or set the SMAC*n*_CLK_SEL.PWDN_X4 register bit to 1. • To save power, assert the SP*n+1*_PWRDN pin and/or set the SMAC*n*_CLK_SEL.PWDN_X1 register bit to 1. • And to additional save power, set the SMAC*n*_CFG_CH{0..3}.PWDN register field to 1. This powers down the SerDes lanes.

3.5.2.1 Signals Sampled on Reset

After hardware reset is de-asserted, the Tsi564A samples the state of the power down pins and only powers up the ports that are enabled. Each RapidIO port has a unique pin, SP{0..7}_PWRDN.

The sampled state of the pins is available immediately in the SMACx_CLK_SEL 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 interfaces.

3.6 Lane Synchronization and Alignment

As a port comes out of reset, it must begin to synchronize its receiver bit sampling and transmit a pattern to allow the other end of the link to synchronize its receiver bit sampling. The port monitors its receive port for /K28.5/ code groups on each lane. Bit errors result in the adjustment of the sampling window. Once a /K28.5/ code group is clearly detected, 127 /K28.5/ code groups must be received error free before the ports attempt to transmit anything else. Receive clocking is recovered from the reception of the /K28.5/ code groups. An error during the sampling of 127 error free code groups causes the code group counter to re-start.

The port continuously transmits /K28.5/ code groups on each lane to assist its link partner in gaining link synchronization.

For a 4x port, after lane synchronization is complete, lane alignment starts. The port transmits /K27.7/ alignment characters /A/'s (||A||) on all four lanes, according to the *RapidIO Interconnect Specification (Revision 1.2)* idle sequence generation rules,. Reception of four ||A||'s without the intervening reception of a misaligned column is the condition for achieving lane alignment. When 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 and the alignment process repeats itself. Bit errors, or receptions of rows without all /A/'s, result in sampling/buffering adjustments.

For more information, see the *RapidIO Interconnect Specification (Revision 1.2) Part VI: Physical Layer 1x/4x LP-Serial Specification*.

3.7 Programmable Driver Current and Equalization

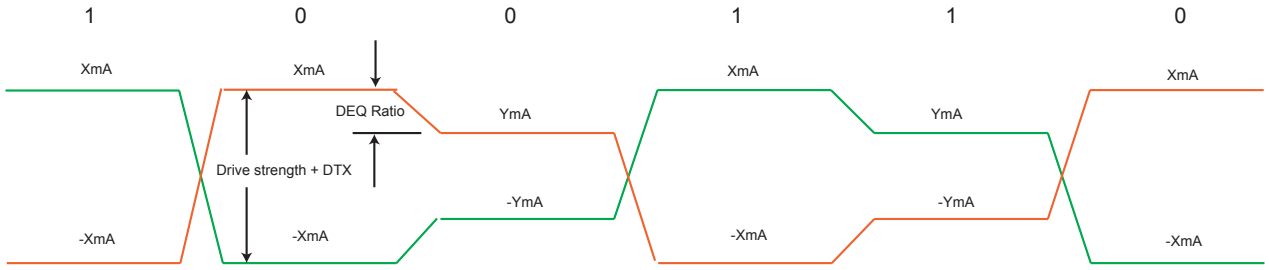
Drive strength can be adjusted depending on the track length and electrical characteristics of the path connecting devices to the Tsi564A, to reduce power consumption or increase the quality of communication. The driver current on each lane can be controlled through the DTX and DRV_STRENGTH fields in the "SRIO MAC x SERDES Configuration — Channel 0" on page 196.

The nominal drive strength for each lane has three possible values (10, 20, and 28 mA) selected by the DRV_STRENGTH field. The actual drive strength is between 60% and 135% of nominal, controllable in 5% increments, depending on the setting of the DTX field.

Additionally, a first order pre-emphasis function is available, controlled by the DEQ field of these same registers. The DEQ field controls the amount that current is decreased after the first bit of a sequence of 1's or 0's is transmitted. The amount of pre-emphasis is specified as a ratio of the pre-emphasis drive strength to the actual drive strength.

Figure 6 shows the drive current and equalization settings and where they impact a transmission waveform.

Figure 6: Drive Strength and Equalization Waveform



3.7.1 Settings

The DTX and DRV_STRENGTH fields can be configured to values outside the *RapidIO Interconnect Specification (Revision 1.2)* limits. IDT has only tested the Tsi564A to the RapidIO limits, any programming outside this range is not recommended.

The SerDes has been tested to meet the RapidIO specification, however operation beyond this range is possible.

For more information on setting the drive strength, please visit www.IDT.com and use the request for support form.

3.7.1.1 Drive Strength Programming Example

If the initial value of the DRV_STRENGTH and DTX are set to 0b10 and 0b0111, an drive strength of 28 mA * 1.35 = 37.8 mA is produced. However, if DEQ is set to 0b1010, the ratio of pre-emphasis drive strength to actual drive strength is 0.40, or 37.8 * 0.40 = 15.12 mA. This means that after the first 1 in a sequence of 1s is transmitted, the drive strength drops from 37.8 mA to 22.68 mA.

Electrical current requirements for SP{n}_Vtt pins are dependent upon this calculation. Each lane may be set independently of the others.

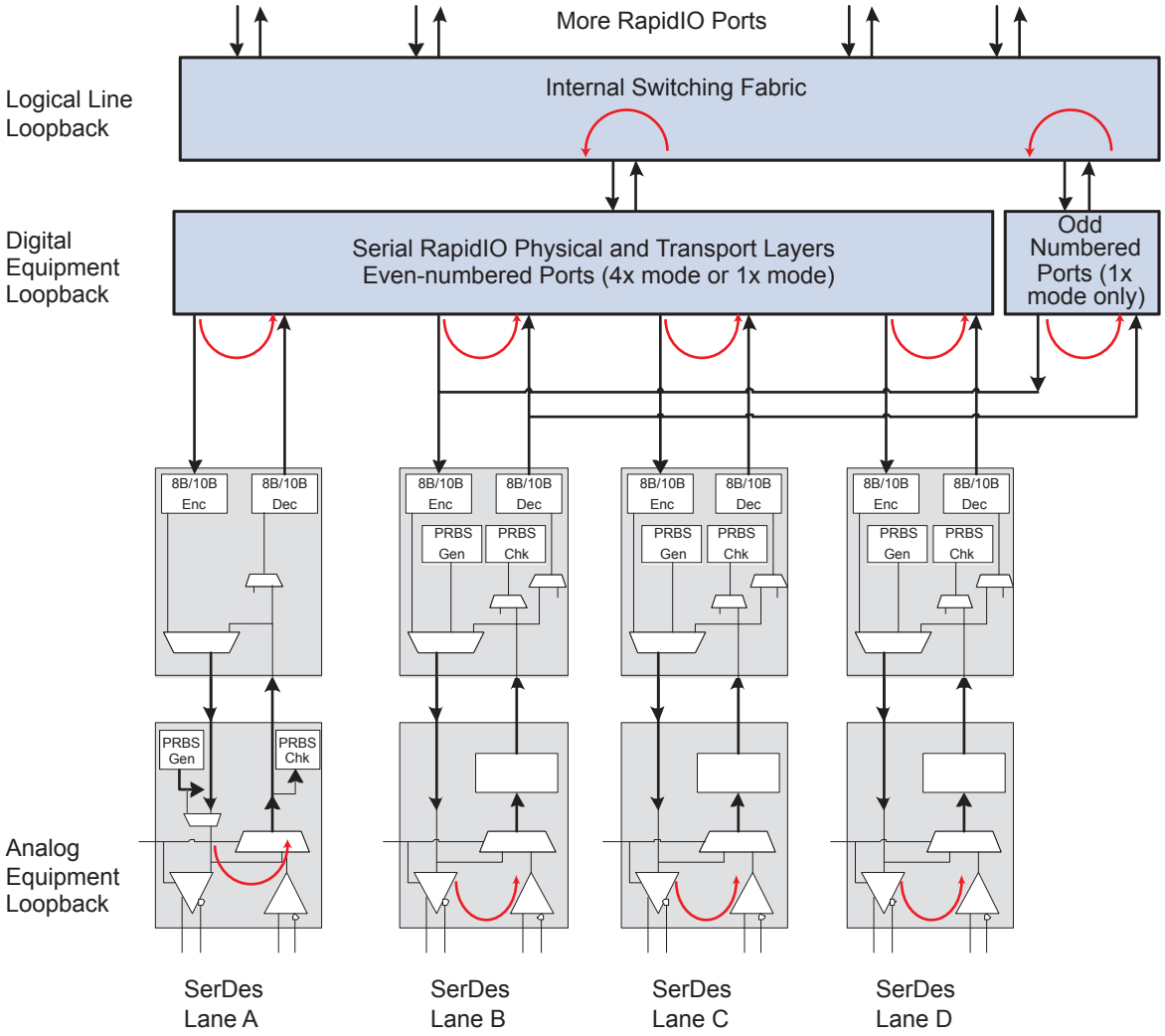
3.8 Port Loopback Testing

The Tsi564A's serial RapidIO ports support the following kinds of loopback:

- Analog equipment loopback
- Digital equipment loopback
- Logical line loopback

Figure 7 on page 59 shows where each loopback is implemented in the Tsi564A MAC.

Figure 7: Tsi564A MAC Loopbacks



3.8.1 Analog Equipment Loopback

Analog equipment loopback is enabled on a per-SerDes-lane basis through the “SRIO MAC x SERDES Configuration — Channel 0” on page 196.

When this form of loopback is enabled, the SerDes transmitter for the selected lane is looped back to the receiver. All incoming data on the external link is ignored.

Analog equipment loopback can only be enabled or disabled when the SerDes is reset. To change the operational state:

- Put the SerDes into reset with the RST_4X_N] bit in the "SRIO MAC x Clock Selection Register" on page 211
- Write the "SRIO MAC x SERDES Configuration — Channel 0" on page 196 register to enable or disable analog equipment loopback
- Wait for 500 ns
- Take the SerDes out of reset with the RST_4X_N bit in the "SRIO MAC x Clock Selection Register" on page 211

Because this loopback MUX is placed just before electrical transmission of the data on the transmit side, and just before the input sampler on the receive side, all normal mode data streams and test mode patterns can be looped back

3.8.2 Digital Equipment Loopback

Digital equipment loopback is enabled on a per-port basis through the "SRIO MAC x Digital Loopback Register" on page 210.

When this form of loopback is enabled, the serial port transmit logic is connected to the receive logic just before where the 8B/10B encoding and transmission would occur. Digital equipment loopback requires the use of packets and a correctly configured look up table. The Bit Error Rate Tester patterns cannot be used when in digital equipment loopback mode. The SerDes does not have to be trained for this loopback to function since the SerDes 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.

3.8.2.1 Disable SerDes Framing

Depending on the type of testing required in the system, the SerDes framing function might need to be disabled in the Tsi564A. For example, loopback framing must be disabled if a BERT test is performed.

To disable the framer, write the FRM_DIS bit in the "SRIO MAC x SERDES Configuration — Channel 0" on page 196. Disabling this feature makes sure that data passes through the loopback path without being re-aligned to 10 bit codeword boundaries.

3.8.3 Logical Line Loopback

Logical line loopback causes a packet sent into the Tsi564A's internal switching fabric to be directed back to the originating port. To cause packets to loop back in this fashion, configure the destination ID lookup tables (LUTs) for the port with line loopback is enabled so that all destination IDs are destined for the incoming port.

For more information on LUT programming, see "Lookup Tables" on page 28.

3.9 Bit Error Rate Testing (BERT)

The RapidIO ports on the Tsi564A 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).

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_PRBS_CTRL register to select either normal operation, PRBS-based BERT, or fixed-pattern-based BERT (see "RIO Serial MAC x PRBS Control Register" on page 213).

When testing a link on the Tsi564A with the BERT feature, the link partner device 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 Tsi564A. Consult the appropriate documentation for other devices to determine if they support these features, and to determine how to configure them.

The Tsi564A PRBS sequence is DC balanced and therefore is suitable for use in an AC coupled system. Caution must be exercised when other sources of PRBS are used. Some PRBS streams may introduce baseline wander and cause an unrealistically high bit error rate.

3.9.1 Fixed Pattern-based BERT

Fixed pattern-based BERT uses data in software-configurable registers to send an alternating pattern of two 10-bit 8B10B code groups. Fixed pattern-based BERT does not produce an automatic pass/fail result.

Fixed patterns are programmed in the RIO Serial MAC x BERT Data register - Channel 0 through Channel 3 (see "RIO Serial MAC x BERT Data Register for Channel 0" on page 224).

The following three patterns are particularly useful for BERT testing:

- 1010101010 1010101010 creates a high-frequency pattern
- 0011111000 1100000111creates a low-frequency pattern
- 0011111010 1100000101creates a mixed-frequency pattern

3.9.1.1 Fixed Pattern-based BERT — Transmitter Configuration

To configure a Tsi564A transmitter for fixed-pattern BERT operation:

- Write the bit stream to be transmitted into the SMACx_CHy_BERT_DATA register. Each bit lane has its own DATA register, which contains a 20 bit pattern that is repeatedly transmitted on the lane.
- Set the PATTERN_SEL field in the "RIO Serial MAC x PRBS Control Register" on page 213 to 0b100 for the desired bit lane(s). Setting this field causes the software defined pattern to transmit.

3.9.1.2 Fixed Pattern-based BERT — Receiver Configuration

The Tsi564A does not automatically generate pass or fail results for fixed-pattern BERT. Because of this, the Tsi564A receiver is usually not part of the BERT. In order to validate that the received fixed-pattern is correct a high-speed oscilloscope or SerDes tester should be used.

3.9.2 PRBS BERT

PRBS BERT uses the polynomial $x^7 - 1$ to generate 10-bit 8B/10B code groups. PRBS BERT produces an automatic pass or fail result.

PRBS testing can be used to test the quality of a link joining two devices made up of Tsi564A devices. Testing is composed of one link partner generating the PRBS pattern, and the other link partner synchronizing its PRBS receiver to the pattern. Therefore there are two scripts provided in the following sections are aids to accomplishing the task of writing all of the necessary registers.

The script can be executed as a file using the JTAG tool available from IDT.

In the context of this example, the port number used on both ends of the link is port 14. The address offsets may be changed to accommodate any other ports that may require testing.

The code example “PRBS BERT — Transmitter Configuration” on page 64 sets up the PRBS generator. The script is composed of only two writes. The first write to 0x13EC0 enables the digital loopback feature of the SerDes design. However, the write may be omitted if the ports on the link have successfully completed the training sequence and are both transmitting idle characters. If the link is not established, in a 4x modes configuration, lanes one and three are idle. Enabling the digital loopback mode will activate the lanes, allowing all four lanes to participate in the PRBS test. Enabling digital loopback if all lanes have achieved training will be transparent to the PRBS testing. The second write turns on the PRBS polynomial sequence generator in all 4 lanes.

The code example titled “PRBS BERT — Receiver Configuration” on page 65, sets up the PRBS receiver and disables the framers on all four lanes of the port because the PRBS stream is not packet-ized. The script then causes the PRBS receivers to synchronize to the PRBS streams for error detection, clears and starts the counters so that they begin from zero. Synchronization is only possible between devices that use the same polynomial for the generation of the PRBS stream.

The string of reads shows the counter registers immediately after the counters are started so the values displayed may not be meaningful.

Due to an errata with the code group counter, the counter actually counts recovered clock cycles.

Using the MON command, the JTAG tool shows on the screen a continuous reading of the registers listed. The display appears as follows:

- 13ED0 xxxxxxxx 13ECC xxxxyyyy

The *x* values are the clock counts were 0x13ED0 contains bits [31:0] of the count, and 0x13ECC contains [41:33].

The *y* values are the code errors detected by the PRBS receiver. This count should stay at zero for a link that is reliable. Should the error count increase, a review of the signal quality on the link is required. Adjustments to the drive strength and de-emphasis of the signal may be all that is necessary to make the link reliable. For more information, see "Programmable Driver Current and Equalization" on page 57.

The following sections discuss only how to configure the Tsi564A's 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.

3.9.2.1 PRBS BERT — Transmitter Configuration

To configure a Tsi564A transmitter for PRBS operation, the SerDes must be powered up and out of reset.The following actions must be performed:

- Put the port into Digital Loopback in order to turn on all 4 lanes of the port.
- Write 0x00000003 to the "SRIO MAC x Digital Loopback Register" on page 210
- Write the SMACx_PRBS_CTRL 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.
- Write 0x19191919 to the "RIO Serial MAC x PRBS Control Register" on page 213

To stop the test:

- Write the SMACx_PRBS_CTRL register, setting the PATTERN_SEL field and the START_PRBS field to 0 for the desired bit lane(s).

Code example

```
////////////////////////////////////

//  PRBS set-up (generation side)

////////////////////////////////////

//

i 0

r 0

//////////////////////////

// the address offsets apply to port 14 of the Tsi564A

// PORT 14 -

//////////////////////////

w 13ec0 00000003 // Force the port into Digital loopback mode

                 // in order to turn on all 4 lanes active

 // if the lanes have not achieved training

                 // resulting in only lanes A and C being active

w 13ec8 19191919 // enable PRBS patter generation (and clear counters)

////////////////////////////////////////////////////////////////////////////////////////////////////

// Now run Receiver Configuration Example on the other side to sync and monitor for errors

////////////////////////////////////////////////////////////////////////////////////////////////////
```

3.9.2.2 PRBS BERT — 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 Tsi564A receiver to validate an incoming PRBS pattern:

- Write the SMACx_CFG_CH{0..3}.FRM_DIS register bits to disable the framer for each SerDes receiver SerDes lane involved in the test
- Write the “RIO Serial MAC x PRBS Control Register” on page 213 register, setting PATTERN_SEL to 1, SYNC_PRBS to 1, and CLR_PRBS_CNT to 1 for the desired bit lane(s). This is accomplished by writing 0x1D1D1D1D to the RIO Serial MAC x PRBS Channel {0..3} Counter registers.

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

- Write the SMACx_PRBS_CTRL 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.
- Tell the transmitter to stop sending PRBS pattern. If the transmitter is a Tsi564A, see "PRBS BERT — Transmitter Configuration" on page 64 for instructions.
- Re-enable the receiver's framer by writing 0 to the SMACx_CFG_CH{0..3}.FRM_DIS register bits.

See "Interpreting the Test Results" on page 68 for instructions on interpreting the results of the test.

Code Example

```
////////////////////////////////////

//  PRBS error checking (Rx side)

////////////////////////////////////

//

i 0

r 0


//////////////////////////

// the address offsets apply to port 14 of the Tsi564A

// PORT 14 -

//////////////////////////


w 13eb0 0005ffff // reset and disable framer ch0

w 13eb4 00050000 // reset and disable framer ch1

w 13eb8 00050000 // reset and disable framer ch2

w 13ebC 00050000 // reset and disable framer ch3
```

w 13eb0 0001ffff // release reset ch0

w 13eb4 00010000 // release reset ch1

w 13eb8 00010000 // release reset ch2

w 13ebC 00010000 // release reset ch3

w 13ec8 1d1d1d1d // sync PRBS receiver and clear counters

// Read counters

r 13ecc //CH0 counter 0

r 13eD0 //CH0 counter 1

r 13eD4 //CH1 counter 0

r 13eD8 //CH1 counter 1

r 13eDc //CH2 counter 0

r 13eE0 //CH2 counter 1

r 13eE4 //CH3 counter 0

r 13eE8 //CH3 counter 1

// Now you can monitor the errors and counters:

// you should see the first counter incrementing as it is

// the “code group counter” (see errata, really clocks..)

// second counter has the higher bits of the code group counter

// lower bits are the errors (should be 0's unless pull cable)

// mon 13eD0 13eCC // channel 0

3.9.2.3 Interpreting the Test Results

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

- Read the "RIO Serial MAC x PRBS Control Register" on page 213 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

- 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 is saturated, a reliable bit error rate cannot be computed. The test should be re-run with a shorter duration.

4. Internal Switching Fabric

This chapter describes the main features and functions of the Tsi564A's Internal Switching Fabric (ISF). It includes the following information:

4.1 Overview

The Internal Switching Fabric (ISF) is the crossbar-switching matrix at the core of the Tsi564A. It transfers packets from ingress ports to egress ports and prioritizes traffic based on the RapidIO priority associated with packet and port congestion.

The ISF has the following features:

- Full-duplex, 8-port, line rate, non-blocking, crossbar-based switching fabric
- 10 Gbit/s fabric ports allow up to 10x internal speedup

The maximum bandwidth stated above is based on the maximum allowable ISF clock speed of 156.25 MHz, and summing the bandwidth for all ingress and egress ports.

- Manages head-of-line blocking on each port
- Buffers hold eight packets per ingress RapidIO port
- Buffers hold eight packets per egress RapidIO port
- Cut-through and store-and-forward switching of variable-length packets

4.2 Functional Behaviour

The ISF is responsible for transporting packets from an ingress port to an egress port. When RapidIO packets arrive at the ingress ports, the Tsi564A 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 ISF transfers entire packets without interruption.

For more information how RapidIO packets are tested as valid, see "RapidIO Interface" on page 27.

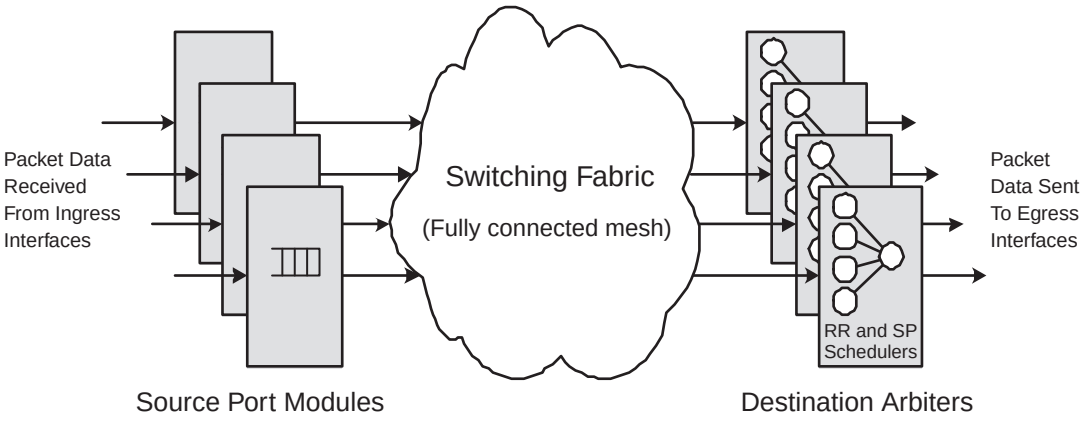
The ISF is a crossbar switch, which means that an ingress port can only send one packet at a time to the ISF, and an egress port can only receive one packet at a time from the ISF. However, the ISF can simultaneously transport packets from multiple disjoint ingress port, egress port pairs. This architecture 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 fabric 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.

Figure 8 illustrates a conceptual block diagram, showing the relationship of the components within the ISF.

Figure 8: ISF Block Diagram



4.2.1 Transfer Modes

The 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 cut-through mode. To change the configuration, write the TRANS_MODE field in the "RIO Port x Control Independent Register" on page 188when traffic is not flowing through the port.

4.2.1.1 Store and Forward Mode

When a port is configured for store-and-forward mode, the port must receive the entire packet before the 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.

In store and forward mode, the incoming packet is not sent to the ISF until the whole packet is received.

4.2.1.2 Cut-through Mode

When a port is configured for cut-through mode, the port is permitted to start sending the packet before the packet has fully arrived at the Tsi564A. This is possible because the RapidIO destination identifier (routing information) appears near the front of a RapidIO packet.

In cut-through mode, the incoming packet is forwarded through the switch as soon as the routing information is received.

Congestion

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

Cut-through mode, generally, 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.

4.3 Arbitration on the 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 a fair 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 avoid head-of-line (HOL) blocking and maximize throughput.

4.3.1 Primary Strict Priority Arbitration

The ISF considers packet priority with a strict priority (SP) service algorithm. The output 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 1.2)*.

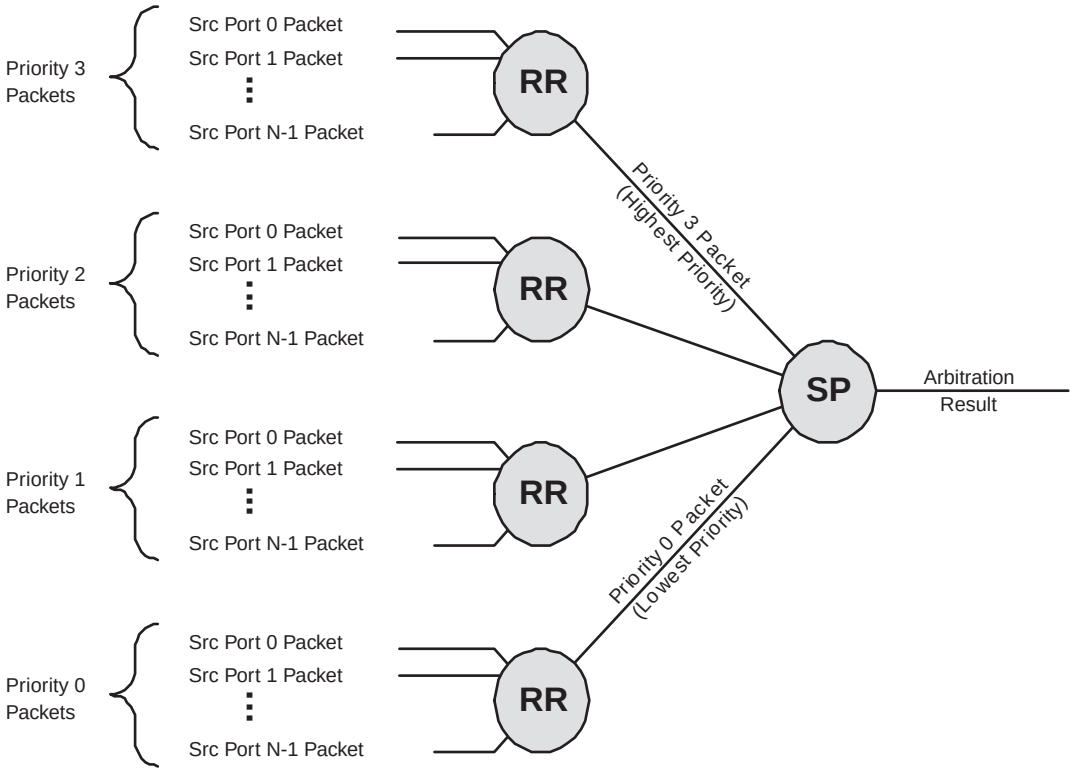
4.3.2 Secondary Round Robin Arbitration

Multiple ingress ports can have packets of the same priority destined to the same egress port. A secondary level of arbitration hierarchy selects among the eligible ports with a round robin (RR) service algorithm.

The arbitration hierarchy for the output arbiters is shown in Figure 9. There is a round robin scheduler for each priority, for each port. 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.

Round robin behaviour based on priority levels can result in temporary unfair bandwidth allocation between ports, because higher priority packets stall the transmission of lower priority packets.

Figure 9: ISF Output Arbitration Hierarchy



4.4 Packet Queuing

The Tsi564A has a queuing system on both the ingress and egress ports.

4.4.1 Output Queuing Model on the Egress Port

Each egress port has a queue that holds up to eight packets. This buffer is required because packets can need retransmitting, and because a egress port with a slow bit rate receives packets from the ISF much faster than the egress port transmits them to their link partner.

The ISF generates switch fabric flow control based on the current depth of the queue. This flow control prevents an egress port from accepting packets of a certain priority. This architecture can result in link-level flow control on the RapidIO ports.

In the event that the output queue is full, the ingress port must begin queuing packets. In the event that the ingress port runs out of buffers for packets, RapidIO link level flow control (packet retries) is activated on the ingress RapidIO port.

For more information on packet retries, see the *RapidIO Interconnect Specification (Revision 1.2)*.

ISF flow control is generated according to the following rules:

- If there are no free packet buffers on the egress port, the egress port does not accept any packets
- If there is at least one free packet buffer on the egress port, the egress port accepts a packet with RapidIO priority 3 only from the next port selected in the priority 3 round-robin arbitration that offers a priority 3 packet.

- If there are at least two free packet buffers on the egress port, the egress port accepts a packet with RapidIO priority 3 from the next port selected in the priority 3 round-robin arbitration that offers a priority 3 packet.

  If no such priority 3 packet is offered, the egress port accepts a packet with RapidIO priority 2 from the next port selected in the priority 2 round-robin arbitration that offers a priority 2 packet.

- If there are at least three free packet buffers, the egress port accepts a packet with RapidIO priority 3 from the next port selected in the priority 3 round-robin arbitration that offers a priority 3 packet.

  If no such priority 3 packet is offered, the egress port accepts a packet with RapidIO priority 2 from the next port selected in the priority 2 round-robin arbitration that offers a priority 2 packet

  If no such priority 2 packet is offered, the egress port accepts a packet with RapidIO priority 1 from the next port selected in the priority 1 round-robin arbitration that offers a priority 1 packet.

- If there are at least four free packet buffers, the egress port accepts a packet with RapidIO priority 3 from the next port selected in the priority 3 round-robin arbitration that offers a priority 3 packet.

  If no such priority 3 packet is offered, the egress port accepts a packet with RapidIO priority 2 from the next port selected in the priority 2 round-robin arbitration that offers a priority 2 packet.

  If no such priority 2 packet is offered, the egress port accepts a packet with RapidIO priority 1 from the next port selected in the priority 1 round-robin arbitration that offers a priority 1 packet.

  If no such priority 1 packet is offered, the egress port accepts a packet with RapidIO priority 0 from the next port selected in the priority 0 round-robin arbitration that offers a priority 0 packet.

The packet offered for selection by the output port is subject to the input queuing arbitration.

4.4.1.1 Transmitting Packets from the Egress Port to the Link Partner

Packets in the output queue are transmitted on the RapidIO link in first-come, first-served (FCFS) order (except during retransmission). Retransmission represents an opportunity for reordering operations as described in input arbitration (see "Input Queuing Buffering Model on the Ingress Port" on page 76).

When a port cannot transmit packets, its output queue becomes full. Since the depth of this queue is used to generate flow control across the internal fabric a mechanism is required to ensure that the fabric does not eventually suffer performance degradation when a port is unable to retire its packets. This is accomplished in the following ways:

- When a port is powered down, it flushes its buffer and continues to accept packets. 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 rest of the system is limited. For more information on detection and recovery from non-operative links, see "Dead Link Timer" on page 41.

4.4.2 Input Queuing Buffering Model on the Ingress Port

Each ingress port has a queue that holds up to eight packets. The buffering is required to deal with congestion in the ISF. Since the ISF is a crossbar switch, each egress port can receive one packet from the 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.

Packets are admitted to the ingress buffer based on the following criteria:

- Packets with RapidIO priority 3 are admitted if there is at least one free packet buffer
- Packets with RapidIO priority 2 are admitted if there are at least two free packet buffers
- Packets with RapidIO priority 1 are admitted if there are at least three free packet buffers
- Packets with RapidIO priority 0 are admitted if there are at least four free packet buffers

The Tsi564A provides registers that system software can use to determine the extent of input congestion on the switch (see "RIO Port x Control Independent Register" on page 188).

Table 12: 8 Packet Buffers and Four Priority Levels

Packet Buffers Available	Packet Priority that can be Accepted
8	0, 1, 2, 3
7	0, 1, 2, 3
6	0, 1, 2, 3
5	0, 1, 2, 3

Table 12: 8 Packet Buffers and Four Priority Levels

Packet Buffers Available	Packet Priority that can be Accepted
4	0, 1, 2, 3
3	1, 2, 3
2	2, 3
1	3
0	none

4.4.3 Input Arbitration

Tsi564A implements a first come, first serve arbitration mode on the ingress port.

If packets are placed in a single input queue, head-of-line (HOL) blocking can result. HOL 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 ISF, even if all the packets, save the first, have an uncongested path to their respective destinations.

The ISF manages HOL blocking by reordering packets in a manner compliant with the *RapidIO Interconnect Specification (Revision 1.2)*. 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.

4.4.3.1 First Come, First Serve Mode

In this mode, packets flow through the ingress queues in order unless reordering is required to manage head-of-line blocking.

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 is no other packets in the buffer.

The packet closest to the head of the queue that can make progress is selected to make progress regardless of its priority.

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

5. I²C Interface

This chapter describes the main features and functions of the I²C interface. It includes the following information:

5.1 Overview

The Tsi564A has a single, master-only, I²C compatible interface that supports up to eight I²C slave devices. The interface is used to load the post-reset register configuration from external EEPROMs.

The primary function of the Tsi564A's I²C Interface is to provide the default configuration after a system boot. The Tsi564A retrieves configuration data from an I²C-compatible serial EEPROM and then writes this to its own system configuration registers.

The block has the following features:

- I²C controller is master only with respect to the I²C bus
- Master register port for the system boot procedure
- General purpose I²C interface operated through four 32-bit internal control/data registers
- Standard mode I²C interface, up to 100 KHz operation
- Supports up to 2K byte of address space in 8 EEPROMs and up to 255 address/data pairs for register configuration during system boot.

5.2 Clock Generation

The I^2C clock is generated by dividing the P_CLK input reference clock by a factor of 1000. For example, with the nominal P_CLK input frequency of 100 MHz, the I^2C clock frequency is 100 kHz.

5.3 Slave Addressing

The slave address for an attached I^2C device is generated from a 4-bit device ID and a 3-bit page number. The 4-bit device ID is the upper 4 bits of the slave address, and is reset to 0xA (the industry standard code for an EEPROM).

The page number is taken from the I^2C control registers during a software-initiated read/write operation, or is derived from the three MSBs of an 11-bit (2K) memory address during the boot load procedure. If a device supports more that 256 bytes, the device must support using the LSB three bits of the device address as a page selector, or multiple slave devices on the bus must respond to unique device addresses based on the page number.

The Tsi564A reads the I^2C address when loading registers it uses the standard EEPROM addressing method. The EEPROM addressing method means that the three least significant bits of the device address can be used to select between different devices, which form the most significant three bits of the I^2C memory address, and the byte transmitted after the device address is the EEPROM memory address.

The Tsi564A also uses the standard device ID for EEPROMs. This means the most significant four bits of the device address are 0b1010. Data is read 8 bytes at a time.

For more information on the format of the EEPROM and device operations for initializing register values from I^2C, see “Boot Description” on page 82.

5.4 Read Sequence

The read sequence placed on the I^2C bus is as follows (italics are slave device responses):

Start, DevCode[3:0], Page[2:0], Write(0), *Ack,* BytAddr[7:0], *Ack*,

...Restart, DevCode[3:0], Stop, Start, DevCode[3:0], Page[2:0], Read(1), *Ack*,

...*DataByte[7:0],* Ack, *DataByte[7:0],* Ack, ..., *DataByte[7:0],* Nack, Stop

The number of data bytes is 8 during the boot load sequence, and 1 to 4 during a software initiated read, based on the Lane and Size fields.

5.5 Write Sequence

The write sequence placed on the I^{2}C bus is as follows (italics are slave device responses):

Start, DevCode[3:0], Page[2:0], Write(0), *Ack*, BytAddr[7:0], *Ack*,

...DataByte[7:0], *Ack*, Stop

...Start, DevCode[3:0], Page[2:0], Write(0), *Ack*/*Nack*, (repeat until *Ack* or timeout), Stop

One data byte is written at a time. The device is then repeatedly addressed until either it asserts *Ack*, indicating the write is complete, or until a 10ms timer expires, in which case an error is signalled and an interrupt generated. If multiple byte writes are needed, based on the Lane and Size fields, the write sequence is repeated with an adjusted BytAddr.

5.5.1 Write Example

This example show how to write to the EEPROM using maintenance transactions or through the JTAG interface.

5.5.1.1 To write into the EEPROM

w 0x1AB00 [0x01000000 /set the RW to write (I2C ctrl reg 1)

| 0x00[dest address in eeprom]0000 /address is an 8 bit field

| 0x0000000[page_sel] /page sel is a three bit field

| 0x0000000A] /DEV_CODE for EEPROM

w 0x1AB0C [data to be written] /I2C data write register

r 0x1AB04 /returns reg_contents (I2C ctrl reg 2)

w 0x1AB04 [reg_contents | 0x00000100] /set the start bit to go

5.6 I^{2}C Controller Read/Write Description

The I^{2}C controller can perform read and write accesses of 1, 2 or 4 bytes in size. For a write access, the data must be written to the “I2C Write Data Register” on page 236. For both reads and writes, the“I2C Control Register 1” on page 232 must then be written to set up the parameters of the access. The “I2C Control Register 2” on page 233 is the last register written. This register write initiates the access.

Software can poll the “I2C Control Register 2” on page 233 for completion of the access. Alternatively, an interrupt will be generated. It is not possible to receive notification for an I^{2}C event through a port write mechanism.

If the access performed was a read, and it completes successfully, the data can be read from the "I2C Read Data Register" on page 235.

The I^2C supports two methods for performing multiple byte writes. The first method is to write each byte as a separate transaction. The second method is to write all bytes as a single transaction. The method used is controlled by the MULTI_BYTE bit in the "I2C Control Register 2" on page 233.

5.6.1 Read Example

This example shows how to read from the EEPROM using maintenance transactions or through the JTAG interface.

5.6.1.1 To read from the EEPROM

w 0x1AB00 [0x00000000 /clear the RW to read (I2C ctrl reg 1)

| 0x00[dest address in eeprom]0000 /Address is an 8 bit field

| 0x0000000[page_sel] /page sel is a three bit field

| 0x0000000A] /DEV_CODE for EEPROM

r 0x1AB04 /returns reg_contents (I2C ctrl reg 2)

w 0x1AB04 [reg_contents | 0x00000100] /set the start bit to go

r 0x1AB08 / returns eeprom_data (I2C read data reg)

5.7 Boot Description

The I^2C Interface always assumes an EEPROM device is attached to the I^2C bus and automatically attempts to read from this device for the purposes of configuring a set of the Tsi564A registers.

The following sections describe this boot procedure.

5.7.1 Serial EEPROM Reset

The I^2C Interface issues an EEPROM reset immediately following the chip hardware reset (HARD_RST_b). The purpose of the EEPROM reset is to re-synchronize the serial EEPROM with the I^2C Interface, after the interface has been reset.

The I^2C Interface generates an EEPROM reset by pulling the SDA signal (I^2C serial data line) high while putting out a sequence of nine pulses on SCLK signal (I^2C serial clock). The nine SCLK pulses clear the I^2C transaction that was in progress prior to the reset and provides one or more NACK signals. Following the ninth pulse, both the SDA signal and the SCLK signal are left high.

If a reset occurs during the write of a data byte to the serial EEPROM, the write is not completed and the data at the target EEPROM address may be corrupted.

5.7.2 Serial EEPROM Detection

Following reset, the I^2C Interface attempts to read the first 8 bytes of the EEPROM, using a slave I^2C address of 0x50. If a serial EEPROM addressed during the boot-up sequence fails to respond with an acknowledge (ACK) signal, the I^2C Interface re-issues the EEPROM address. The I^2C Interface retries the EEPROM address up to six times if there is no response.

If no acknowledge signal is received after the sixth attempt, I^2C Interface assumes that a serial EEPROM is not present. When there is no serial EEPROM present the I^2C Time-out bit is set, which raises the interrupt signal (INT).

5.7.2.1 Slave Address

This section explains the origin of the 0x50 value used in the I2C slave address.

The value for discovering an EEPROM device is 0xA (or 0b1010). Since the A2, A1, A0 bits need to be tied to VSS (since only one device is supported by the interface), and the address field is made up of seven bits plus a R/W bit, the following is true:

- A6-A5-A4-A3-A2-A1-A0-RW
- 1 0 1 0 0 0 0 RW

By taking the address portion (the left/upper 7 bits) 1010000 and converting the nibbles to hex, 0x101 and 0x0000, the final value of 0x50 is derived.

5.7.3 EEPROM Data Format

The first 8 bytes of the EEPROM contain the number of registers to be loaded during the boot procedure. This count is a 16-bit value in EEPROM location 0 (MSB) and location 1 (LSB). The I^2C Interface is limited to 255 register loads, so byte location 0 must always be set to zero and byte location 1 must contain an 8-bit count. The remaining 6 bytes (memory locations two through seven) should be set to 0xFF and are ignored by the I^2C Interface.

Any 16-bit value greater than 255 (0x00FF) aborts the boot load from the EEPROM.

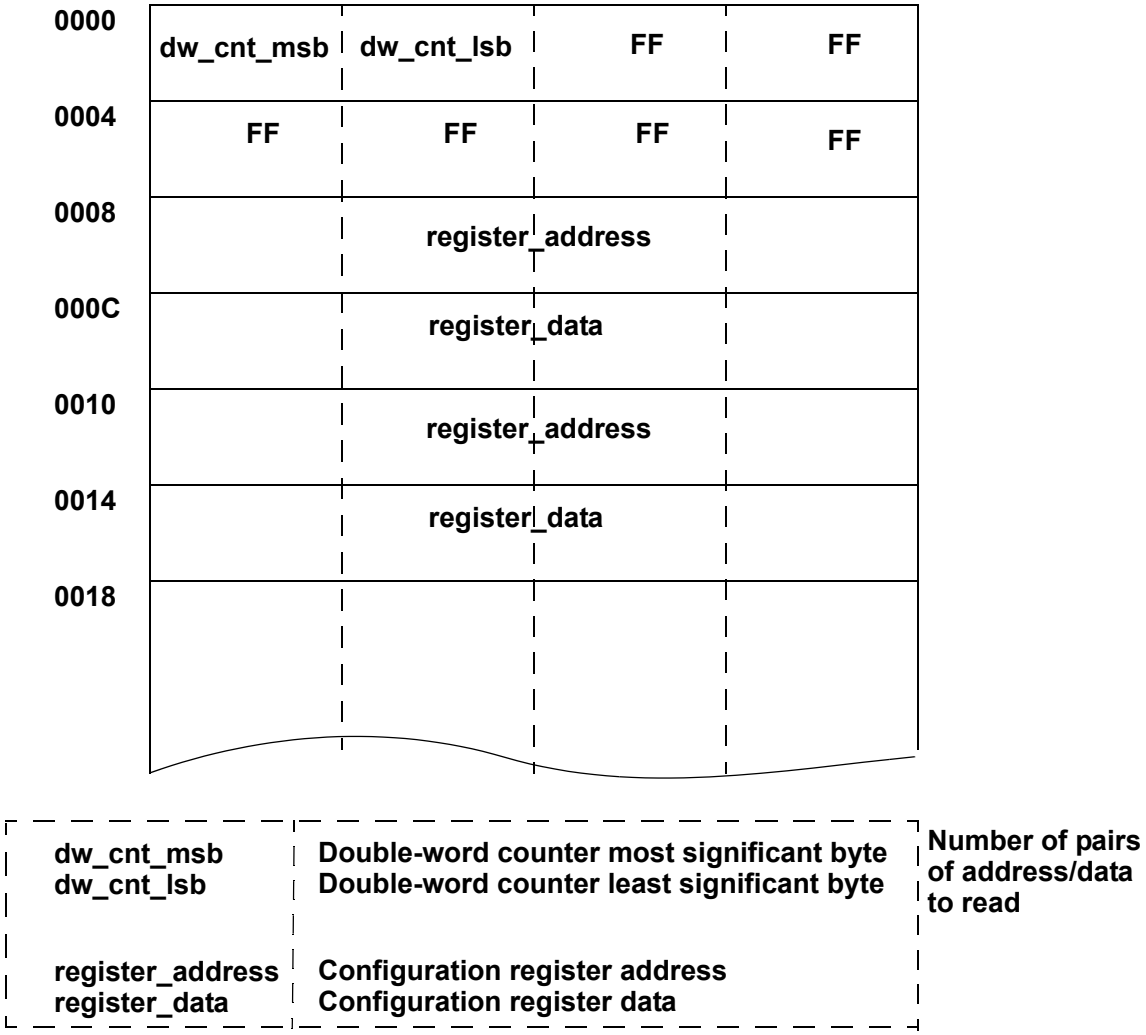
The I^2C Interface can load initial values to any configuration register of the device. During I^2C boot of the system all normally writable register fields are writable and some read-only register fields are also writable. For details on which read-only fields can be written during the I^2C boot tim, see “Registers” on page 125.

After reading the register count, the I^2C Interface reads blocks of data 8 bytes at a time from the EEPROM. The first 4 bytes are used as the register address, and the next 4 bytes are the data to load into the register. The address and data are placed on the internal register bus, causing the data to be written to the addressed register. The data from the EEPROM is written to the internal register as is; no checking is performed to ensure that data values fall within valid ranges for each register being written.

Internally, an 11-bit address is maintained and incremented by 8 following each register load. This 11-bit address is then partitioned into a 3-bit page number (bits 10:8) and an 8-bit address for the purposes of positioning and reading the next 8 consecutive bytes, as described in the I^2C Read Sequence section (Section 5.4 on page 80).

Figure 10 shows the expected EEPROM format.

Figure 10: EEPROM Data Format



5.7.4 I²C Boot Time

The time required to perform I²C boot varies depending on the number of registers that require configuration during the boot time.

Each register read from the EEPROM requires 105 clock cycles on the I²C Interface. The time required for bootup can be calculated using the following equation:

Boot_Time = I2C_Clock_Period * (Num_Registers + 1) * 105 cycles/register

Where:

I2C_Clock_Period = 10 us (100 kHz clock)
Num_Registers = number of registers that are being read from the EEPROM

In the worst case, 255 registers would be read, giving a boot time of:

Boot_Time = 10 us/cycle * 256 registers * 105 cycles/register
Boot_Time = 268800 us (0.27 seconds)

6. Event Notification

This chapter describes the system of error and event notification in the Tsi564A. It includes the following information:

6.1 Overview

The Tsi564A has the following ways to notify external devices about events occurring within the switch:

1. Generate a RapidIO port-write maintenance message when enabled (as described in the *RapidIO Interconnect Specification (Revision 1.2)*).
2. 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 Tsi564A.

6.2 Event Summary

Table 13 describes all the events that can be raised within the Tsi564A and whether these events generate a interrupt, a port-write, or both.

Table 13: Tsi564A Events

Event Name	Type	Description	Can Generate Interrupt?	Can Generate Port-write?
Max 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 RIO Port x Error Detect CSR (see Section 10.6.3 on page 172) and the MAX_RETRY bit in the RIO Port x Interrupt Status Register (see Section 10.7.11 on page 192).	Yes	Yes
Illegal Transaction	Error	This event occurs when an inbound port receives a transaction with one of the following errors: • Maintenance transaction type is not a read or write • Unmapped entry in LUT • Reserved TT field value The status of this event is contained in the IMP_SPEC_ERR bit in the RIO Port x Error Detect CSR (see Section 10.6.3 on page 172).	Yes	No
LUT Parity Error	Error	This event occurs when a parity error is detected when a port is performing a destination ID lookup. The status of this event is contained in the IMP_SPEC_ERR bit in the RIO Port x Error Detect CSR (see Section 10.6.3 on page 172) and the LUT_PAR_ERR bit in the RIO Port x Interrupt Status Register (see Section 10.7.11 on page 192).	Yes	Yes
Reset Request Received	Status	This event occurs when four consecutive reset requests are received by an inbound port. This event is described in detail in Section 8.2.1.3 on page 109. The status of this event is contained in the RCS bit in the RIO Port x Reset Control Symbol Interrupt CSR (see Section 10.7.6 on page 185), as well as the RCS bit in the Global Interrupt Status Register (see Section 10.11.1 on page 237).	Yes	No

Table 13: Tsi564A Events *(Continued)*

Event Name	Type	Description	Can Generate Interrupt?	Can Generate Port-write?
TEA in Fabric	Error	This event is raised when a fabric transmission request times out and a packet is dropped. The status of this event is contained in the TEA bit of the RIO Port x Error Detect CSR (see Section 10.6.3 on page 172). Global TEA status is found in the TEA bit of the Global Interrupt Status Register (see Section 10.11.1 on page 237), and the bits of the Fabric Interrupt Status Register (see Section 10.9.2 on page 230).	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 with link-response, but no outstanding ackID • Four link-request with time-out error for link-response The status of this event is contained in the PORT_ERR bit of the RIO Port x Error and Status CSR (see Section 10.6.3 on page 172).	Yes	Yes
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 typically used to detect hot-swap events. The status of this event is contained in the LINK_INIT_NOTIFICATION bit of the RIO Port x Interrupt Status Register (see Section 10.7.11 on page 192).	Yes	Yes

6.3 Physical Layer Errors

Many errors that are detectable by the RapidIO physical layer are recoverable. These errors are listed in Table 14. These errors do not cause an interrupt or port write to occur.

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

Table 14: Recoverable Error Events

RapidIO Error	Description	Capture Registers
Implementation Specific (Logical Error)	The Tsi564A Switch uses the implementation specific error to combine three other error events, so that they can be included within the Error Rate reporting mechanism. These events are: • Illegal Transaction Error (Reserved TT type, unsupported maintenance transaction) • Max Retry Occurred Error • Parity Error in Lookup Table • Unmapped destination ID for packet • ISF TEA Error When any of these events occur, the Implementation Specific event is considered to have occurred. The status of this error is contained in the IMP_SPEC_ERR bit in the RIO Port x Error Detect CSR (see Section 10.6.3 on page 172).	Yes
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 in the RIO Port x Error Detect CSR (see Section 10.6.3 on page 172).	Yes
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_ID bit in the RIO Port x Error Detect CSR (see Section 10.6.3 on page 172).	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 in the RIO Port x Error Detect CSR (see Section 10.6.3 on page 172).	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 in the RIO Port x Error Detect CSR (see Section 10.6.3 on page 172).	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 in the RIO Port x Error Detect CSR (see Section 10.6.3 on page 172).	Yes

Table 14: Recoverable Error Events *(Continued)*

RapidIO Error	Description	Capture Registers
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 in the RIO Port x Error Detect CSR (see Section 10.6.3 on page 172).	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 in the RIO Port x Error Detect CSR (see Section 10.6.3 on page 172).	No
Protocol error	An unexpected packet or control symbol was received. The status of this error is contained in the PROT_ERR bit in the RIO Port x Error Detect CSR (see Section 10.6.3 on page 172).	Yes
Delineation error	Received unaligned /SC/ or /PD/ or undefined code-group. The status of this error is contained in the DELIN_ERR bit in the RIO Port x Error Detect CSR (see Section 10.6.3 on page 172).	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 in the RIO Port x Error Detect CSR (see Section 10.6.3 on page 172).	Yes
Link time-out	An acknowledge-response or Link-response is not received within the specified time-out interval, see the RIO_SP_LT_CTL register (Section 10.5.2 on page 153).	No

6.4 Port-Write Notifications

In the Tsi564A all RapidIO ports can generate port-write messages based on interrupt events. Because of this architecture, the RapidIO interrupt enables also control whether a port-write message is issued for each interrupt. In each port the register bit IRQ_EN in the "RIO Port x Control Independent Register" on page 188 controls whether any enabled interrupts are propagated to the "Global Interrupt Status Register" on page 237 and therefore creates a port-write. If the IRQ_EN bit is disabled, no interrupt propagates to the GLOB_INT_STAT register from this port.

Each port has its own copy of the "RIO Port-Write Target Device ID CSR" on page 171. When a port is powered down and powered up, or reset, the RIO Port-Write Target Device ID CSR must be reprogrammed.

The system is notified of most events that occur in the Tsi564A RapidIO interfaces through the RapidIO port-write message. The port-write mechanism is enabled by default, but can be disabled through the PW_DIS field within the "RIO Port x Mode CSR" on page 183. Table 13 on page 88 indicates which events cause RapidIO Port-write messages. Note that, by default, all events that can trigger a port-write to be sent are disabled.

The port-write packet does not have a guaranteed delivery and does not have an associated response (see *RapidIO Interconnect Specification (Revision 1.2)*). RapidIO systems are allowed to drop port-write packets.

When the Port-write mechanism 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 "RIO Port-Write Target Device ID CSR" on page 171. 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 has been sent, no port-write is sent for the new event.

System designers must ensure that the number of port-writes sent under error conditions do not impair intended system operation.

Depending on system design, a port write can be sent repeatedly until cleared. The programmable time out counter is defined in the PW_TIMER field of the "RIO Port x Discovery Timer" on page 181.

6.4.1 Source and Destination ID

There is only one port-write destination ID programmed for the entire device; 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. For more information, see "RIO Port-Write Target Device ID CSR" on page 171.

Each port has its own copy of the RIO Port-Write Target Device ID CSR. When a port is powered down and powered up, or reset, the RIO Port-Write Target Device ID CSR must be reprogrammed.

The port-write logic also writes the source ID and hop-count fields for the packet. The source ID is filled with 0xFF or 0xFFFF (depending on the transaction type value). The hop count is set to 0xFF to ensure that the packet is routed by the switches in the architecture.

6.4.2 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 15 on page 93.

The payload of the maintenance port-write packet is defined by the *RapidIO Interconnect Specification (Revision 1.2) RapidIO Error Management Extensions*.

Table 15 shows the port write packet data payload for error reporting. Note that port-writes are sent at priority 0 (lowest priority).

Table 15: 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: • bit 13 - LINK_INIT_NOTIFICATION bit in the RIO Port x Interrupt Status Register (see Section 10.7.11 on page 192)	Port ID (8 bits)	0s

6.4.3 Port-writes and Hot Insertion/Hot Extraction Notification

Port-Write requests are used to support hot insertion/extraction notification. For more information, see "Hot Insertion and Hot Extraction" on page 37.

The sending device sets the PORT_W_PEND status bit in the "RIO Port x Error and Status CSR" on page 161. Software indicates that it has seen the port-write operation by clearing the PORT_W_PEND bit. In order to clear the PORT_W_PEND bit, the PW_TIMER in the "RIO Port x Discovery Timer" on page 181 must be 0.

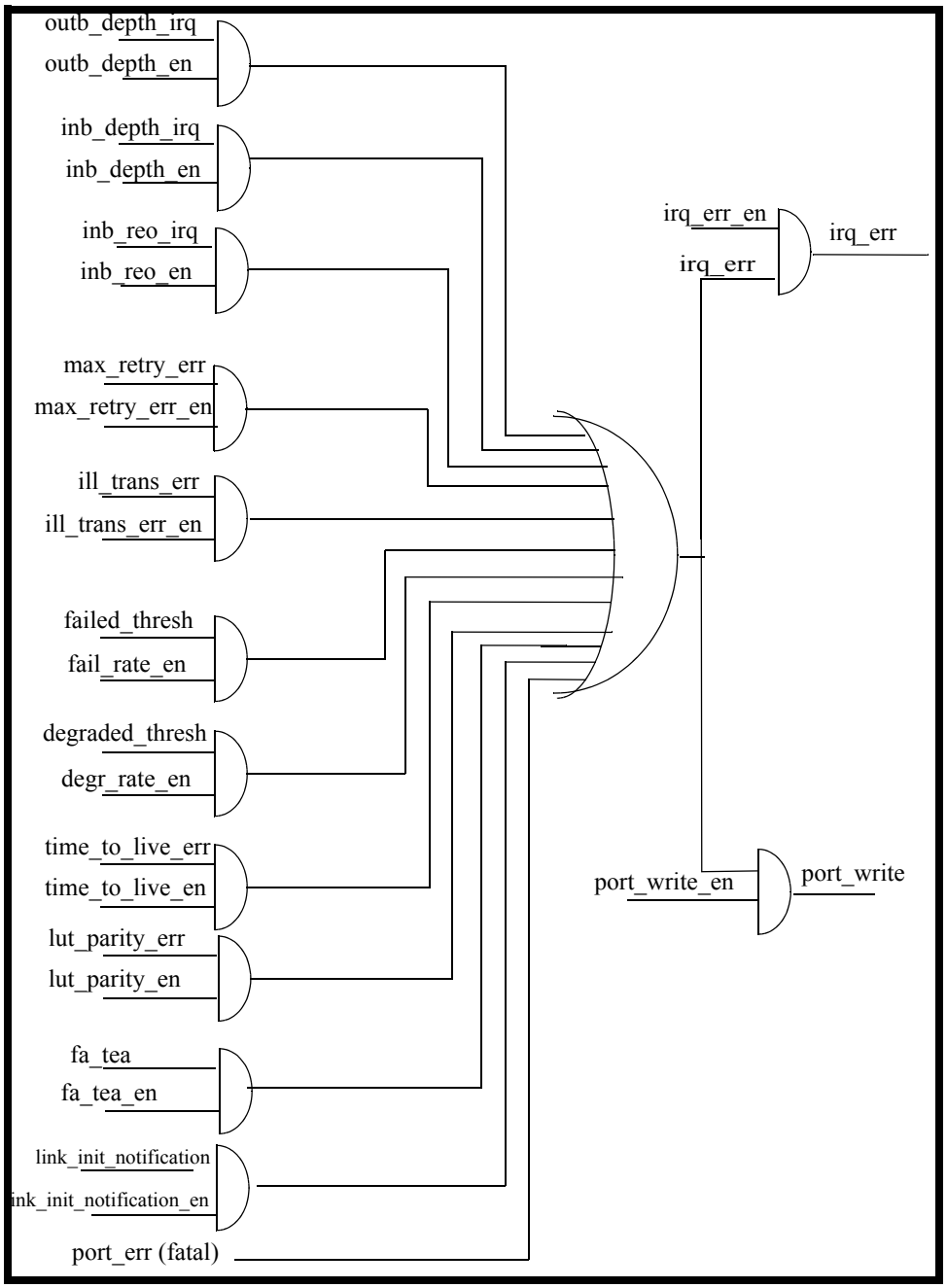
6.5 Interrupt Notifications

In the Tsi564A interrupts are hierarchical, which allows software to determine the cause of the interrupt with minimum register access.

System designers must decide upon a maximum rate of interrupt notifications, and set the error thresholds appropriately.

Figure 11 illustrates the interrupt hierarchy within the Tsi564A RapidIO ports.

Figure 11: RapidIO Interrupt and Port Write hierarchy



6.5.1 INT_b Signal

At the top level of the interrupt hierarchy is the external interrupt signal INT_b. 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 device.

The INT_b signal is driven by the Global Interrupt Status register (see Section 10.11.1 on page 237). The INT_b signal is asserted when any bit within the Global Interrupt Status register is set and its corresponding enable bit in the Global Interrupt Enable register is also set (see Section 10.11.2 on page 239). If an interrupt in the Global Interrupt Enable register is not enabled, the bit in Global Interrupt 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 propagated up the interrupt hierarchy.

6.5.2 Global Interrupt Status Register and Interrupt Handling

The Global Interrupt Status register must be read to determine why the interrupt was raised. Interrupt causes in the Global Interrupt Status register shows the cause of the interrupt and enables the interrupt service routine to decide which port raised an interrupt (see Section 10.11.1 on page 237).

After the software has read the Global Interrupt Status register and determined which port has an interrupt pending, the port's interrupt status registers must be accessed to determine the exact cause. Each port contains an interrupt status register (see Section 10.7.11 on page 192) and an associated interrupt enable register (see Section 10.7.9 on page 188). 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 GLOB_INT_STAT register if that interrupt is enabled (see Section 10.11.1 on page 237).

Table 16 details the interrupt status and interrupt enable registers for each port.

Table 16: Tsi564A Per Port Interrupt Registers

Name	Interrupt Status Register	Interrupt Enable Register
Serial RapidIO	RIO Port x Interrupt Status Register (see Section 10.7.11 on page 192)	RIO Port x Control Independent Register (see Section 10.7.9 on page 188)
Internal Switching Fabric	Fabric Interrupt Status Register (see Section 10.9.2 on page 230)	Fabric Control Register (see Section 10.9.1 on page 229)

6.5.3 Interrupt Notification and Port-writes

In the Tsi564A all RapidIO ports can also generate port-write messages based on interrupt events. Because of this architecture, the RapidIO interrupt enables also control whether a port-write message is issued for each interrupt. In each port the register bit IRQ_EN in RIO Port x Control Independent Register (see Section 10.7.9 on page 188) controls whether any enabled interrupts are propagated to the GLOB_INT_STAT register and therefore create a port-write. If the IRQ_EN bit is disabled, no interrupt propagates to the GLOB_INT_STAT register from this port.

6.5.4 Reset Control Symbol and Interrupt Handling

Reception of a valid reset control symbol sequence is a port specific feature that has separate indicator bits to allow for faster interrupt handling. Reception of a reset control symbol sequence is signalled by the RCS bit in the "Global Interrupt Status Register" on page 237. Each port has a "RIO Port x Reset Control Symbol Interrupt CSR" on page 185 to allow software to determine which port received the reset control symbol. The reset control sequence interrupt can be cleared on a per port basis, or globally through the "RIO Port x Reset Control Symbol Interrupt CSR" on page 185 broadcast register.

7. JTAG Interface

This chapter describes the main features of the JTAG interface. It includes the following information:

7.1 Overview

The JTAG Interface includes dedicated user-accessible test logic that is compliant with the *IEEE1149.1 Standard Test Access Port and Boundary Scan Architecture*. The IEEE 1149.1 compliant JTAG interface is used for device testing, boundary scan, and to read and write registers.

The Joint Test Action Group (JTAG) created the boundary-scan testing standard (documented in the *IEEE 1149.1 Standard*) for testing printed circuit boards (PCBs).

The boundary-scan approach involves designing boundary-scan circuitry into the integrated circuit. PCBs populated with 1149.1 compliant devices can be tested for connectivity, correct device orientation, correct device location, and device identification. All the pins on compliant devices can be controlled and observed using (typically) five pins that are routed to the board edge connector. Board designers can develop a standard test for all 1149.1 compliant devices regardless of device manufacturer, package type, technology, or device speed.

The SerDes pins are not on the boundary scan chain.

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 Tsi564A's configuration registers
- Supports the following instruction opcodes:
  - Sample/Preload
  - Extest
  - Bypass
  - Hi-Z
  - IDCODE
  - Clamp
  - User data select

7.2 JTAG Device Identification Numbers

The JTAG device identification number for the Tsi564 is 0x00564167.

7.3 JTAG Register Access Details

The Tsi564A has the capability to read and write registers through the JTAG interface.

Prior to using the IEEE Register Access Command feature, the part must be reset by driving TRST_b low.

The JTAG Interface has the ability to access registers in order to debug issues that may affect RapidIO register accesses.

A user defined command is used to enable the read and write capabilities of the interface. The command is in the IEEE 1149.1 Instruction Register (IR) in the Tsi564A.

- IEEE Register Access Command (IRAC)

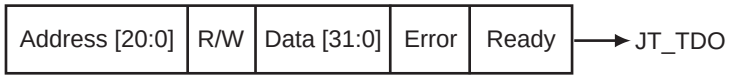
There must be IEEE 1149.1 capability on the board to use the IEEE 1149.1 register access feature.

7.3.1 Format

The format used to access the registers is shown in Figure 12 and Figure 13. The address shown in the figure is the RapidIO offset.

Figure 12: Register Access From JTAG - Serial Data In

Figure 13: Register Access From JTAG - Serial Data Out



7.3.2 Write Access to Registers from the JTAG Interface

The following steps are required in order 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 by writing into Instruction Register bits [2:0] with 0x5.
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 un-used bits, the full DR bits have to be written. This is completed by writing DR[112:56] with 0x0, DR[55:35] with addr, DR[34] with 0x1, DR[33:2] with data and DR[1:0] with 0x0.
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 113 zero bits to DR[112:0], while at the same time verify the Ready and Error bits that are being shifted-out as the first two bits shifted-out.
5. Go back to step two to perform another write.

7.3.3 Read Access to Registers from the JTAG Interface

The following steps are required in order 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.
2. This step is optional if the instruction register is already programmed during the write cycle.
3. 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 un-used bits, the full DR bits have to be written. This is completed by writing DR[112:56] with 0x0, DR[55:35] with addr, DR[34] with 0x0, DR[33:0] with 0x0.
4. Move to the “Run-test idle” state and loop in this state for a minimum of 20 TCK cycles.
5. Move to the “Shift-DR” state and shift the data[31:0] out through TDO. To prevent corruption of un-used bits, write into reserved bits DR[112:56] with 57'b0.
6. The Error and Ready bits are shifted out at the same time.
7. Verify that the Error bit is at logic low and the Ready bit is at logic high.
8. Go back to step two to perform another read.

7.4 JTAG Register Access Software

The IDT JTAG Register Access Software (JTAG access software) is a DOS-based application that allows access to the internal registers of IDT devices for debug and monitoring purposes.

The Tsi564A JTAG chain must be isolated from other device’s in the JTAG chain when using the JTAG Register Access Software.

This software is available at www.IDT.com.

8. Clocks, Resets and Power-up Options

This chapter describes the clock and reset of the Tsi564A. It includes the following information:

8.1 Clocks

The Tsi564A has three input reference clocks that are used to produce the Tsi564A's internal clock domains.

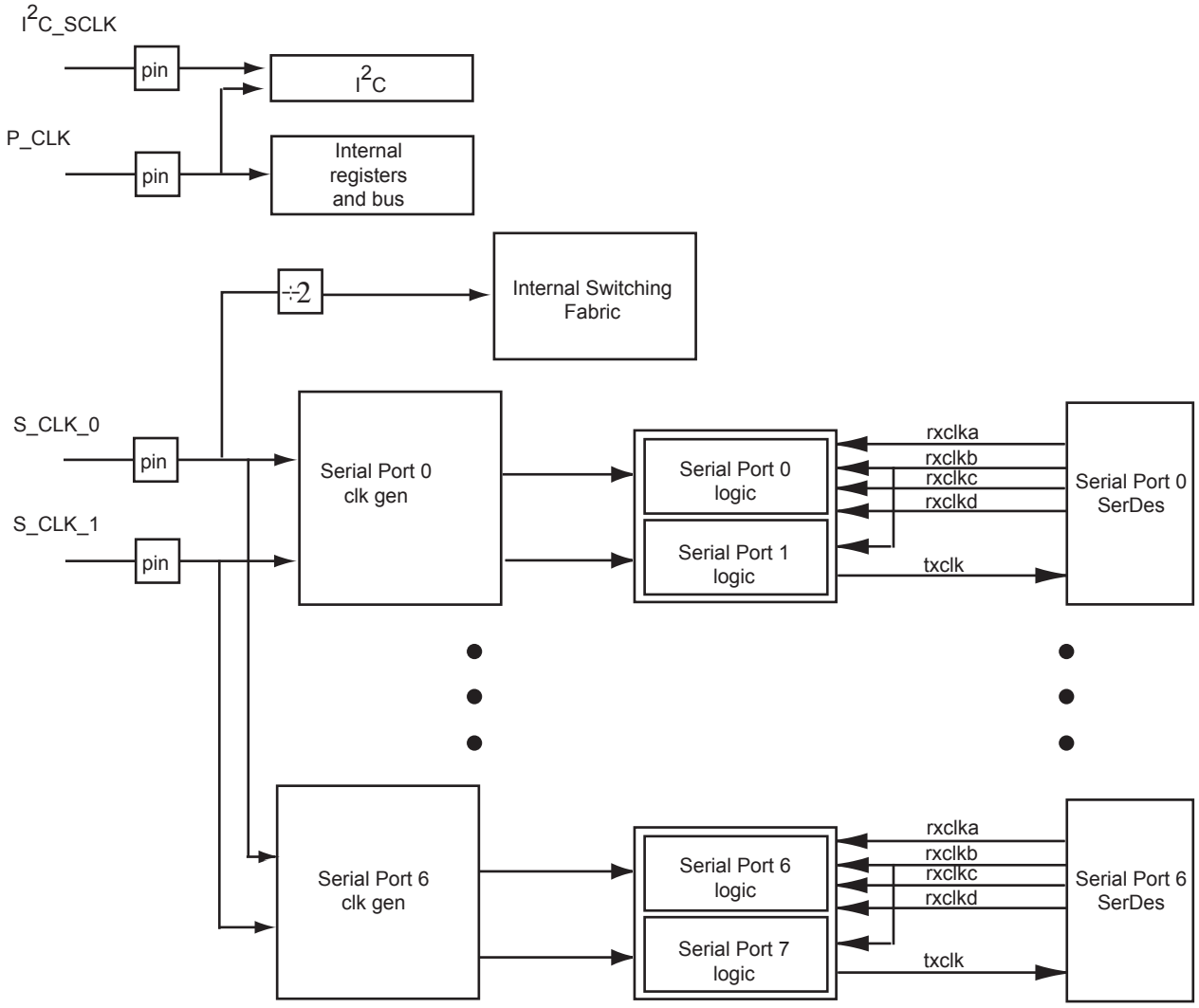
In most configurations, all three reference clocks are required, but in some configurations it is possible to use only two input clocks.

In addition to these reference clocks, each RapidIO ingress port contains an independent receive clock domain. The receive clock is extracted from the 8B/10B encoding on each lane.

8.1.1 Clocking Architecture

The following diagram illustrates the clocking architecture of the Tsi564A.

Figure 14: Tsi564A Clocking Architecture



8.1.2 PLL(s)

There are two PLLs within each MAC, one for lanes 0 and 1, the other for lanes 2 and 3. The PLLs both lock within 30 microseconds of HARD_RST_b de-assertion. The PLLs are brought up sequentially by logic internal to the Tsi564A.

8.1.3 Reference clocks

The three reference clocks are described in Table 17. Each listed clock domain is described in detail in Section 8.1.4 on page 105.

For information on configuring the clock rate of RapidIO ports, see “Clocking” on page 52.

Table 17: Tsi564A Input Reference Clocks

Clock Input Pin	Type	Maximum Frequency	Clock Domains
S_CLK_1_p/n	Differential	312.5 MHz	Serial Transmit Domain 2 (Maximum 312.5 MHz) Internal Switching Fabric (ISF) Domain
S_CLK_2_p/n	Differential	250 MHz	Serial Transmit Domain 1 (Nominally 250 MHz) Serial Transmit Domain 0 (Nominally 125 MHz) Note: If the Tsi564A never uses Serial Transmit Domains 1 and 2, then this input reference clock is not required to be driven.
P_CLK	Single-ended	100 MHz	Internal Register Domain I^2C Domain

8.1.4 Clock Domains

The Tsi564A contains a number of clock domains that are generated from the three input reference clocks. These domains are detailed in Table 18.

Table 18: Tsi564A Clock Domains

Clock Domain	Clock Source	Maximum Frequency	Description
Internal Register Domain	P_CLK	100 MHz	This clock domain includes all internal registers within each of the internal blocks, as well as the AHB bus that performs the register accesses. The domain uses the input P_CLK directly.
Internal Switching Fabric (ISF) Domain	S_CLK_1_p/n (divided by 2)	156.25 MHz	This clock domain includes the switching matrix of the ISF, and the portion of each interface that communicates with the ISF. The domain uses the S_CLK_1_p/n divided by 2.

Table 18: Tsi564A Clock Domains *(Continued)*

Clock Domain	Clock Source	Maximum Frequency	Description
I^2C Domain	P_CLK divided by 1000	97.7 kHz	This clock domain is responsible for driving the I^2C output clock pin I2C_SCLK. This clock domain is generated by dividing the P_CLK input by 1000. The majority of the I^2C logic runs in the Internal Register Domain.
Serial Transmit Domain 0	S_CLK_2_p/n (divided by 2)	125 MHz	This clock domain is used to clock all the Serial RapidIO transmit ports that have the SCLK_SEL field in the SMACx_CLK_SEL register set to 0. The S_CLK_2_p/n input is divided in half and used to clock the transmit logic. This clock is multiplied by 10 to produce the high-speed clock that is used to output the serial data on output pins SP{0..15}_T{A..D}_p/n. The maximum data rate for this domain is 1.25 Gb/s per lane.
Serial Transmit Domain 1	S_CLK_2_p/n	250 MHz	This clock domain is used to clock all the Serial RapidIO transmit ports that have the SCLK_SEL field in the SMACx_CLK_SEL register set to 1. The S_CLK_2_p/n input is used directly to clock the transmit logic. This clock is multiplied by 10 to produce the high-speed clock that is used to output the serial data on output pins SP{0..15}_T{A..D}_p/n. The maximum data rate for this domain is 2.5 Gb/s per lane.
Serial Transmit Domain 2	S_CLK_1_p/n	312.5 MHz	This clock domain is used to clock all the Serial RapidIO transmit ports that have the SCLK_SEL field in the SMACx_CLK_SEL register set to 2. The S_CLK_1_p/n input is used directly to clock the transmit logic. This clock is multiplied by 10 to produce the high-speed clock that is used to output the serial data on output pins SP{0..15}_T{A..D}_p/n. The maximum data rate for this domain is 3.125 Gb/s per lane.

8.1.5 Clocking Scenarios

This section lists a number of scenarios to decrease power by reducing or eliminating the frequency of the reference clocks.

The minimum required frequency of the Internal Switching Fabric (ISF) clock domain is:

- For a 4x serial RapidIO port: The minimum frequency is equal to the clock frequency divided by 20 (for example, 3.125 Gb/s / 20 = 156.25 MHz)
- For a 1x serial RapidIO port: The minimum frequency is equal to the clock frequency divided by 80.

The minimum possible clock frequency for the ISF clock domain is 62.5 MHz, because the slowest possible RapidIO frequency for S_CLK_1_p/n is 1.25 GHz.

Table 19 lists possible scenarios and indicates the minimum frequency the ISF must maintain to provide enough data to fill the output port bandwidth.

Table 19: Clocking Examples

Example Description	Minimum Required ISF Frequency	S_CLK_1_p/n Input	S_CLK_2_p/n Input
Requirement: Serial RapidIO ports to operate at 3.125, 2.5, and 1.25 Gbit/s. The 3.125 Gbit/s serial RapidIO frequency is the limiting factor for the ISF frequency. The ISF frequency must be 156.25 MHz.	156.25 MHz	312.5 MHz	250 MHz
Requirement: Serial RapidIO ports to operate at 3.125 GHz rate for 4x ports. The 3.125 Gbit/s serial RapidIO frequency is the limiting factor for the ISF frequency. The ISF frequency must be 156.25 MHz.	156.25 MHz	312.5 MHz	Not required
Requirement: Serial RapidIO ports to operate at 2.5 Gb/s rate for 4x ports. The 2.5 Gbit/s serial RapidIO frequency is the limiting factor for the ISF frequency. The ISF frequency must be at least 125 MHz. Since the ISF clock is half of the S_CLK_1 input, S_CLK_1 must be 250 MHz.	125 MHz	250 MHz	Not required
Requirement: Serial RapidIO ports to operate at 1.25 Gb/s rate for 4x ports. The 1.25 Gbit/s serial RapidIO frequency is the limiting factor for the ISF frequency. The ISF frequency must be at least 62.5 MHz. Since the ISF clock is half of the S_CLK_2 input, S_CLK_2 must be 125 MHz.	62.5 MHz	125 MHz	Not required

Table 19: Clocking Examples

Example Description	Minimum Required ISF Frequency	S_CLK_1_p/n Input	S_CLK_2_p/n Input
Requirement: One Serial RapidIO port to operate at 1.25 Gbit/s rate in 4x mode and the other ports to operate in 1x mode at 2.5 Gbit/s. For this scenario the 1.25 Gb/s serial RapidIO frequency is the limiting factor for the ISF frequency. The ISF frequency must be at least 62.5 MHz.	62.5 MHz	125 MHz	250 MHz

8.2 Resets

Internal logic is responsible for automatically sequencing the removal of reset; no additional software programming is required.

8.2.1 Device Reset

The Tsi564A can be rest the following ways:

1. Assertion of the HARD_RST_b input pin
2. Issuing a RapidIO reset request to any of the RapidIO ports.
   - The SELF_RST bit in the RIO Port x Mode CSR (see Section 10.7.5 on page 183) must be set to 1

In both cases, when the Tsi564A is reset it goes through its full reset and power-up sequence. All state machines and the configuration registers are reset to the original power on states.

Lookup tables are left in an undefined state after reset. It is recommended that lookup tables be completely initialized after a reset to ensure deterministic operation.

8.2.1.1 I^2C Boot

When all interfaces have been taken out of reset, the I^2C Interface is responsible for performing automatic reads from an externally attached EEPROM device in order to load the initial configuration of the device. For more details, see "I2C Interface" on page 79.

The I^2C device is not reset by the Tsi564A, so the I^2C bus could be left in an undefined state if the Tsi564A is reset during initial configuration. It is recommended that resets of the Tsi564A occur at a rate that ensures that register loading from I^2C device has completed before another reset is issued.

8.2.1.2 HARD_RST_b Reset

The HARD_RST_b signal is a external system reset input signal and causes a general reset of the Tsi564A; all blocks are reset within the device. HARD_RST_b is an active low signal with asynchronous assertion and de-assertion. The internal reset synchronizers are responsible for assuring that reset is de-asserted internally at the correct time for each of the clock domains.

HARD_RST_b must be asserted for a minimum of 1 millisecond. The power rails must be stable before the 1 millisecond countdown begins. Also, P_CLK must be operating and stable before the countdown begins.

When HARD_RST_b is asserted, SW_RST_b is de-asserted. SW_RST_b remains de-asserted after HARD_RST_b is released.

8.2.1.3 RapidIO Reset Requests

The Tsi564A responds to four link-request/reset control symbols (as defined by the *RapidIO Interconnect Specification (Revision 1.2)*). The receiving port triggers a reset if it receives four link-request/reset control symbols with only idle control symbols between them. The requirement to receive four link-request/reset control symbols prevents spurious resets due to transmission errors.

Self Reset

When a reset request occurs, the Tsi564A's response is controlled by the RIO Port x Mode CSR (see Section 10.7.5 on page 183). By default, the Tsi564A resets itself. A self-reset occurs when a reset request is received and the SELF_RST field in the RIO Port x Mode CSR is set. When a self-reset is performed, it is not necessary to drive the HARD_RST_b input signal. The SW_RST_b signal remains asserted for the duration of the self reset, which is at least four P_CLK clock cycles.

If the SELF_RST field is not set an interrupt signal is asserted (if RCS_INT_EN is also asserted, see Section 10.7.5 on page 183).

System Control of Resets

Self-reset of the Tsi564A may not be sufficient in systems which require other components to be reset at the same time as the Tsi564A. The Tsi564A supports system control of resets in two ways. First, the Tsi564A can generate an interrupt so that a local processor can trigger a reset through the Tsi564A's HARD_RST_b pin. For this design to work, reset interrupts must be enabled in the RIO Port x Mode CSR and in the Global Interrupt Enable Register (see Section 10.11.2 on page 239).

Secondly, if interrupts are not suitable for reset functionality in a system, it is possible to use the SW_RST_b output pin. When the Tsi564A has received a reset request, the SW_RST_b output pin is asserted until the reset request status is cleared in the port that received it. The SW_RST_b output pin can be used as an input to a reset controller to trigger the start of a system reset. If self-reset is not enabled, SW_RST_b remains asserted until the device is reset through the input reset pin HARD_RST_b or until the interrupt bit is cleared for the port that received the reset message. If self-reset is enabled, SW_RST_b is asserted for the duration of the reset, which is at least four P_CLK clock cycles.

When the Tsi564A is in reset, the INT_b pin is not asserted.

8.2.2 Per-Port Resets

Each RapidIO port within the Tsi564A can be independently reset. To perform the reset it is necessary to power down the port and then power it up again. The port power down feature is described in Section 3.5 on page 53.

After a port has been reset, the contents of the ports configuration registers revert to reset default values and the LUT for the reset port must be re-configured; register values are not loaded from I^2C on a port reset.

After a port has been reset, all IDT-Specific RapidIO Registers must be reprogrammed. This includes those registers which support broadcast addresses in the address range 0x10000 through 0x100FF.

Register values are maintained when a port is disabled. To disable a port, set PORT_DIS to 1 in the RIO Serial Port x Control CSR (see Section 10.5.8 on page 164).

8.2.3 Generating a RapidIO Reset Request to a Peer Device

The following steps show how software can reset a peer device:

1. Determine which RapidIO port is connected to the peer to be reset.
2. Alter the LUT contents to ensure that no packets are being routed to the link partner that is to be reset.

3. Lockout the port using the PORT_LOCKOUT field in the RIO Serial Port x Control CSR (see Section 10.5.8 on page 164). This ensures that any traffic received from the peer device will be dropped, and any traffic still in flight to the peer device will be 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 RIO Serial Port x Link Maintenance Request CSR (see Section 10.5.4 on page 155) register to transmit four reset control symbols in a row.
5. Write '0' to the OUTBOUND field of the RIO Serial Port x Local ackID Status CSR (see Section 10.5.6 on page 159).
6. Unlock the port using the PORT_LOCKOUT field within the RIO Serial Port x Control CSR (see Section 10.5.8 on page 164).
7. The Tsi564A 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 will therefore not be acknowledged immediately by the Tsi564A. The link partner will time out waiting for a packet acknowledge control symbol, and will enter output-error stopped state. To recover, as per the RapidIO specification the link partner will send a link-request/input-status control symbol to the Tsi564A port. This will clear the input-error stopped state on the Tsi564A. The Tsi564A will respond with a link-response/status control symbol, which the link partner will accept and exit the output-error stopped state. The maintenance response packet is then retransmitted and accepted by the Tsi564A.

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 Tsi564A recovery from input-error stopped state will be delayed, and the step 6 maintenance write will potentially time-out at the originating entity.

8. Perform a maintenance write to the OUTBOUND field of the RapidIO Serial Port x Local ackID Status CSR of the peer device.

8.2.4 JTAG Reset

The JTAG TAP controller's reset is independent of the Tsi564A functional resets. The TAP controller can be reset with either the external pin TRST_b or by holding the pin TMS asserted for more than five TCK cycles.

To ensure predictable operation of the Tsi564A, for power-up reset, HARD_RST_b and TRST_b must be asserted prior to operation. After power-up, the TAP controller can be reset at any time and this will not affect the Tsi564A operation.

Normal functional reset is still required to reset the device's internal registers. Reset of the Tsi564A does not reset the TAP.

The TAP controller must be reset on power-up, whether or not it is going to be used, to ensure predictable operation of the Tsi564A.

8.3 Power-up Options

The Tsi564A 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_MODESEL).

The power-up option pins must be stable for 10 P_CLK cycles after HARD_RST_b is de-asserted.

8.3.1 Power-up Option Signals

Power up options are latched at reset for initializing the Tsi564A. The power-up option pins are listed in Table 20.

The power-up signals do have internal PU or PD, however external resistors are recommended on these signals.

Table 20: Power-up Option Signals

Pin Name	Pin Type	Description
SP{n}_MODESEL	I/O, CMOS PD	Selects the serial port operating mode for ports n and n+1 0 - Port n operating in 4x mode (Port n+1 not available) 1 - Ports n and n+1 operating in 1x mode **Note:** This signal is not latched, but must remain at the required value from the time the port(s) are powered up to the time the ports(s) are powered down. For more information on changing the port mode selected, see "Changing Port Mode" on page 51.
SP_IO_SPEED[1]	I/O, CMOS PD	Serial Port Transmit and Receive operating frequency select, bit 1. When combined with SP_IO_SPEED[0], this pin selects the default serial port frequency for all ports. 00 - S_CLK_2 reference divided by 2 01 - S_CLK_2 reference (default) 10 - S_CLK_1 reference 11 - reserved The output data rate per lane is 10 times the selected input clock. Selects the speed at which the ports operates when reset is removed.
SP_IO_SPEED[0]	I/O, CMOS, PU	See SP_IO_SPEED[1]
SP{n}_PWRDN	I/O, CMOS PU	Port n Transmit and Receive Power Down control This signal controls the state of Port n *and Port n+1* The PWRDN controls the state of all four lanes (A/B/C/D) of SerDes Macro. 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_x1 field in the "SRIO MAC x Clock Selection Register" on page 211.
SP{n+1}_PWRDN	I/O, CMOS PU	Port n+1 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 Serial Rapid IO MAC, 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_x4 field in the "SRIO MAC x Clock Selection Register" on page 211.

8.3.2 Default Port Speed

The speed is overridden by the SCLK_SEL field in the SRIO MAC x Clock Selection Register (see Section 10.8.6 on page 211).

8.3.3 Port Power-up and Power-down

The power-up and power-down is overridden by the PWDN_X1 and PWDN_X4 fields in the SRIO MAC x Clock Selection Register (see Section 10.8.6 on page 211).

8.3.4 Mode Selection

The operating width of a 4x port can be overridden by the OVER_PWIDTH field in the RIO Serial Port x Control CSR (see Section 10.5.8 on page 164).

When a port operating in 4x/1x mode must change to enable another port mode, the procedure in Section 3.3.3 on page 51 must be followed.

9. Signals and Pinlist

This chapter describes the signals and pinout of the Tsi564A. It includes the following information:

9.1 Overview

The following conventions are used in the pin description table:

- 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.
- DDR signals are Double Data Rate, data is transferred on both edges of the associated clock.

Signals are classified according to the types defined in Table 21 on page 115.

Table 21: Signal Types

Pin Type	Definition
I	Input
O	Output
I/O	Input/Output
OD	Open Drain
SRIO	CML driver/receiver defined by *RapidIO Interconnect Specification (Revision 1.2)*
PU	Pulled Up internal to the Tsi564A
PD	Pulled Down internal to the Tsi564A
Hyst	Hysteresis

Table 21: Signal Types *(Continued)*

Pin Type	Definition
Core Power	Core supply
Core Ground	Ground for core logic
I/O Power	I/O supply

9.2 Endian Ordering

This document follows the bit-numbering convention adopted by *RapidIO Interconnect Specification (Revision 1.2)*, where [0:7] is used to represent an 8 bit bus with bit 0 as the most-significant bit.

9.3 Signal Groupings

Figure 15 summarizes the Tsi564A signals.

Figure 15: Tsi564A Pinout List

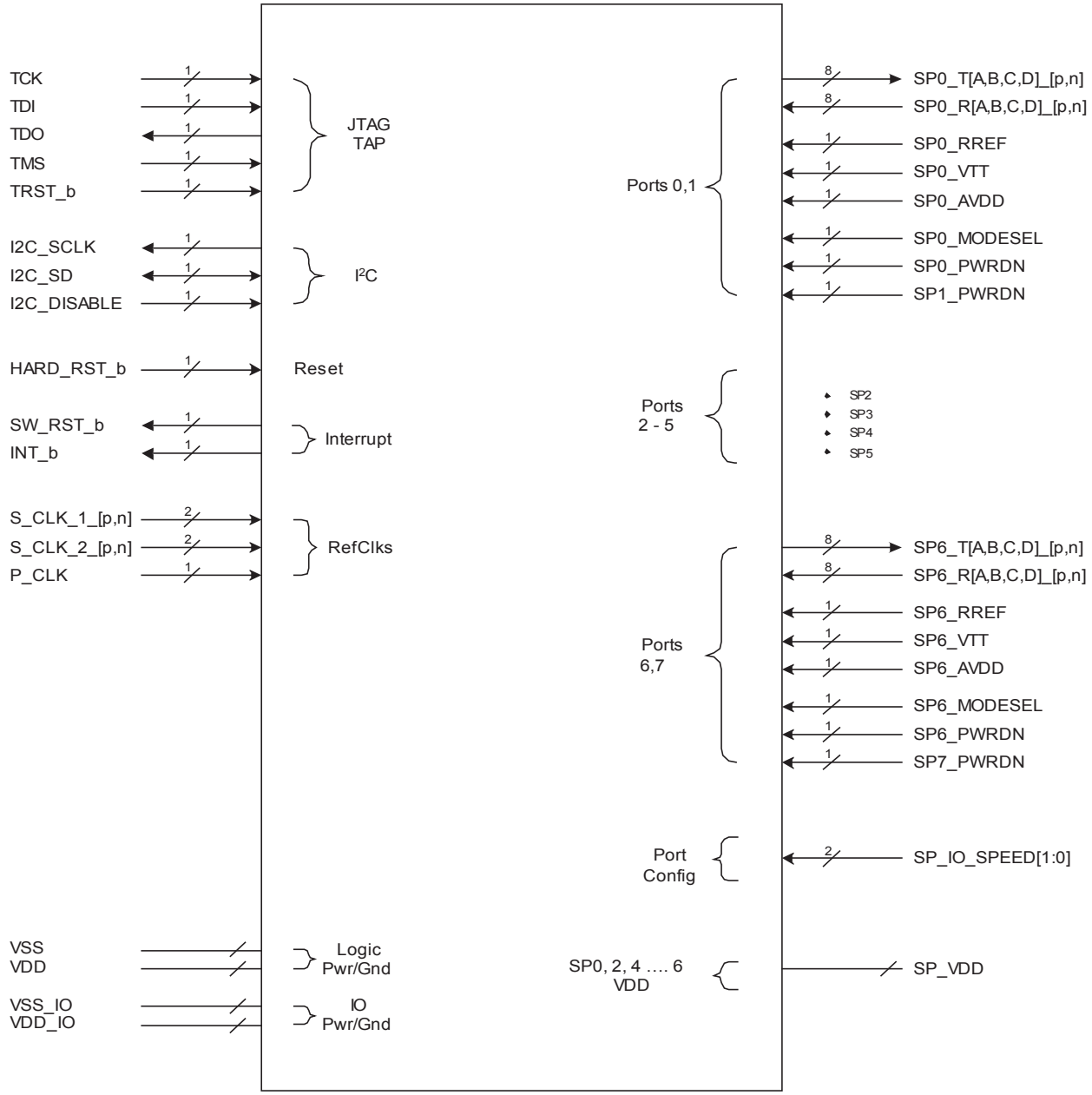


Table 22 describes the Tsi564A signals.

Table 22: Tsi564A Signal Description

Pin Name	Pin Count	Type	Description
PORT n - 1x/4x Mode Serial RapidIO PORT (n+1) - 1x Mode Serial RapidIO where n = 0, 2, 4, 6			
SERIAL PORT n/n+1 TRANSMIT where n = 0, 2, 4, 6, 8			
SP{n}_TA_p	1	O, SRIO	Port n Lane A Differential Non-inverting Transmit Data output (4x mode) Port n Differential Non-inverting Transmit Data output (1x mode)
SP{n}_TA_n	1	O, SRIO	Port n Lane A Differential Inverting Transmit Data output (4x mode) Port n Differential Inverting Transmit Data output (1x mode)
SP{n}_TB_p	1	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)
SP{n}_TB_n	1	O, SRIO	Port n Lane B Differential Inverting Transmit Data output (4x mode) Port n+1 Differential Inverting Transmit Data output (1x mode)
SP{n}_TC_p	1	O, SRIO	Port n Lane C Differential Non-inverting Transmit Data output (4x mode)
SP{n}_TC_n	1	O, SRIO	Port n Lane C Differential Inverting Transmit Data output (4x mode)
SP{n}_TD_p	1	O, SRIO	Port n Lane D Differential Non-inverting Transmit Data output (4x mode)
SP{n}_TD_n	1	O, SRIO	Port n Lane D Differential Inverting Transmit Data output (4x mode)
SERIAL PORT n/n+1 RECEIVE where n = 0, 2, 4, 6			
SP{n}_RA_p	1	I, SRIO	Port n Lane A Differential Non-inverting Receive Data input (4x node) Port n Differential Non-inverting Receive Data input (1x mode)
SP{n}_RA_n	1	I, SRIO	Port n Lane A Differential Inverting Receive Data input (4x node) Port n Differential Inverting Receive Data input (1x mode)
SP{n}_RB_p	1	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)
SP{n}_RB_n	1	I, SRIO	Port n Lane B Differential Inverting Receive Data input (4x mode) Port n+1 Differential Inverting Receive Data input (1x mode)
SP{n}_RC_p	1	I, SRIO	Port n Lane C Differential Non-inverting Receive Data input (4x mode)
SP{n}_RC_n	1	I, SRIO	Port n Lane C Differential Inverting Receive Data input (4x mode)
SP{n}_RD_p	1	I, SRIO	Port n Lane D Differential Non-inverting Receive Data input (4x mode)
SP{n}_RD_n	1	I, SRIO	Port n Lane D Differential Inverting Receive Data input (4x mode)

Table 22: Tsi564A Signal Description *(Continued)*

Pin Name	Pin Count	Type	Description
SERIAL PORT n/n+1 CONFIGURATION where n = 0, 2, 4, 6			
SP{n}_RREF	1		Used to connect a 1K +/-1% resistor to SP{n}_AVDD to provide a reference current for the driver and equalization circuits.
SP{n}_MODESEL	1	I/O, CMOS PD	Selects the serial port operating mode for ports n and n+1 0 - Port n operating in 4x mode (Port n+1 not available) 1 - Ports n and n+1 operating in 1x mode Must remain stable for 10 P_CLK cycles after HW_RST_b is de-asserted in order to be sampled correctly. Ignored after reset.
SP{n}_PWRDN	1	I/O, CMOS PU	Port n Transmit and Receive Power Down control This signal controls the state of Port n *and Port n+1* The PWRDN controls the state of all four lanes (A/B/C/D) of SerDes Macro. 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_x1 field in SRIO MAC x Clock Selection Register (see Section 10.8.6 on page 211) **Note:** Output capability of this pin is only used in test mode. Must remain stable for 10 P_CLK cycles after HW_RST_B is de-asserted in order to be sampled correctly. Ignored after reset.
SP{n+1}_PWRDN	1	I/O, CMOS PU	Port n+1 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 Serial Rapid IO MAC, 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_x4 field in SRIO MAC x Clock Selection Register (see Section 10.8.6 on page 211). **Note:** Output capability of this pin is only used in test mode. Must remain stable for 10 P_CLK cycles after HW_RST_B is de-asserted in order to be sampled correctly. Ignored after reset.
SERIAL PORT SPEED SELECT			

Table 22: Tsi564A Signal Description *(Continued)*

Pin Name	Pin Count	Type	Description
SP_IO_SPEED[1]	1	I/O, CMOS, PD	Serial Port Transmit and Receive operating frequency select, bit 1. When combined with SP_IO_SPEED[0], this pin selects the default serial port frequency for all ports. 00 - S_CLK_2 reference divided by 2 01 - S_CLK_2 reference (default) 10 - S_CLK_1 reference 11 - reserved The output data rate per lane is 10 times the selected input clock. Selects the speed at which the ports operates when reset is removed. This could be either due to HARD_RST_b being de-asserted or by the completion of a self-reset. This signal must remain stable for 10 P_CLK cycles after HW_RST_B is de-asserted in order to be sampled correctly. The signal is ignored after reset. **Note:** The SP_IO_SPEED[1:0] setting is equal to the SCLK_SEL field in the SRIO MAC x Clock Selection Register (see Section 10.8.6 on page 211). **Note:** Output capability of this pin is only used in test mode.
SP_IO_SPEED[0]	1	I/O, CMOS, PU	See SP_IO_SPEED[1]
CLOCK and RESET			
P_CLK	1	I CMOS	This clock is used for the register bus clock. The maximum frequency of this input clock is 100 MHz.
S_CLK_1_p	1	I CML	Differential non-inverting reference clock. The clock is used for following purposes: SerDes reference clock, serial port system clock, ISF clock (equal to half of this clock) and test clock. The clock frequency is defined in the Minimum Clock Frequency Requirements section. The maximum frequency of this input clock is 312.5 MHz. If this clock input is not used, pull this signal up.
S_CLK_1_n	1	I CML	Differential inverting reference clock. The clock is used for following purposes: SerDes reference clock, serial port system clock, ISF clock (equal to half of this clock) and test clock. The clock frequency is defined in the Minimum Clock Frequency Requirements section.The maximum frequency of this input clock is 312.5 MHz. If this clock input is not used, pull this signal down.

Table 22: Tsi564A Signal Description *(Continued)*

Pin Name	Pin Count	Type	Description
S_CLK_2_p	1	I CML	Differential non-inverting reference clock. The clock is used for following purposes: SerDes reference clock, serial port system clock, 125MHz clock (equal to half of this clock) and test clock. The clock frequency is defined in the Minimum Clock Frequency Requirements section. The maximum frequency of this input clock is 250 MHz. If this clock input is not used, pull this signal up.
S_CLK_2_n	1	I CML	Differential inverting reference clock. The clock is used for following purposes: SerDes reference clock, serial port system clock, 125MHz clock (equal to half of this clock) and test clock. The clock frequency is defined in the Minimum Clock Frequency Requirements section.The maximum frequency of this input clock is 250 MHz. If this clock input is not used, pull this signal down.
HARD_RST_b	1	I CMOS, Hyst, PU	Schmidt-triggered hard reset. Asynchronous active low reset for the entire device.
INTERRUPTS			
INT_b	1	O, OD, CMOS	Interrupt signal (open drain output)
SW_RST_b	1	O, OD, CMOS	Software reset (open drain output): This signal is asserted when a RapidIO port receives a valid reset request on a RapidIO link. If self-reset is not selected, this pin remains asserted until the reset request is cleared from the status registers. If self-reset is selected, this pin remains asserted until the self reset is complete. If the Tsi564A is reset from the HARD_RST_b pin, this pin is de-asserted and remains de-asserted after HARD_RST_b is released. For more information, see “Resets” on page 108.
MISCELLANEOUS			
I^2C			
I2C_SCLK	1	O, OD, CMOS, PU	I^2C clock, up to 100 kHz. Although this clock is open drain, the I2C controller does not support multiple bus masters. This clock signal must be connected to the clock of the serial EEPROM on the I2C bus.
I2C_SD	1	I/O, OD, CMOS, PU	I^2C input and output data bus (bidirectional open drain)

Table 22: Tsi564A Signal Description *(Continued)*

Pin Name	Pin Count	Type	Description
I2C_DISABLE	-	I, CMOS, PD	Disable I^2C register loading after reset. When asserted, the Tsi564A will not attempt to load register values from I^2C. 0 - Enable I^2C register loading 1- Disable I^2C register loading
JTAG / TAP CONTROLLER			
TCK	1	I, CMOS, PD	IEEE 1149.1 Test Access Port Clock input
TDI	1	I, CMOS, PU	IEEE 1149.1 Test Access Port Serial Data Input
TDO	1	O, CMOS	IEEE 1149.1 Test Access Port Serial Data Output
TMS	1	I, CMOS, PU	IEEE 1149.1 Test Access Port Test Mode Select
TRST_b	1	I, CMOS, PU	IEEE 1149.1 Test Access Port TAP Reset Input This input should asserted during a power-up reset.
POWER SUPPLIES			
PORT n/n+1 n = 0, 2, 4, 6			
SP{n}_AVDD	1	-	Port n and n+1: 1.2V supply for bias generator circuitry. This is required to be a low-noise supply.
SP{n}_VTT	1	-	Port n and n+1: Driver termination voltage - common to all lanes
COMMON SUPPLY - TIS564			
VDD_IO	12	-	Common 3.3V supply for CMOS I/O
VSS_IO	12	-	Common ground supply for I/Os
VSS	188	-	Common ground supply for digital logic
VDD	32	-	Common 1.2V supply for digital logic
SP_VDD	30	-	1.2V supply for CDR, Tx/Rx, and digital logic for all RapidIO ports
Total Power and Ground	274		

9.4 Pinlist and Ballmap

Pinlist and ballmap information for the Tsi564A is available by visiting www.IDT.com and using the request for support form. For more information, see the following documents:

- *Tsi564A Pinlist*
- *Tsi564A Ballmap*

10. Registers

This chapter describes the Tsi564A registers. The following topics are discussed:

10.1 Overview

The following rules apply to the Tsi564A registers:

- All read/write (R/W) registers are read/write during the I^2C boot
- All read only (R) registers are not read/write during the I^2C boot unless it specified in the register description
- Application defined registers receive initial values during power on initialization through the I^2C interface and external serial EEPROM

10.1.1 Reserved Register Addresses and Fields

Reserved register addresses should not be read or written. Reads to reserved register addresses return unspecified data and may result in undefined behavior. Writes to reserved register addresses may lead to unpredictable results.Reserved fields should always be written as 0 (unless otherwise specified).

Table 23 shows the defined register access types.

Table 23: Register Access Types

Abbreviation	Description
R	Read Only
R/W	Read or Write
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)
RC	When read these bits are automatically cleared These fields are writable for test purposes

This register specification uses direct addressing of 32-bit registers. The RapidIO specification, *RapidIO Interconnect Specification (Revision 1.2)*, uses 64-bit addressing of registers. Table 24 shows the rules used to associate the register offsets in both specifications.

Table 24: Address rules

Tsi564A 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

The ports are mapped as shown in Table 25.

Table 25: Port Map

Port	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	1x/4x Serial port
7	1x Serial port

10.2 Conventions

Often, there are multiple instances of a register, for example, one instance per RapidIO port. The following two notations are used describe these 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 explicitly listed. For example, P{BC,0..2}_DESTID refers to registers PBC_DESTID, P0_DESTID, P1_DESTID, and P2_DESTID.

Generally, 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 when the register is read, returns a value from port number 0. For this reason, port 0 must be powered-up when broadcast registers are used.

10.3 Register Map

Table 26 gives an overview of the Tsi564A register map.

Table 26: 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	0x000FC
RapidIO Physical Layer Registers (Parts VI of the RapidIO Specification)	0x00100	0x003FC
Reserved	0x00400	0x00FFC
RapidIO Error Management Extension Registers (Part VIII of the RapidIO Specification)	0x01000	0x013BC
Reserved	0x013C0	0x0FFFF
IDT-Specific RapidIO Registers, and Serial Port Electrical Layer Registers	0x10000	0x14FFC
Reserved	0x15000	0x1ABFC
Internal Switching Fabric (ISF) Registers	0x1AA00	0x1AAFC
I^2C Registers	0x1AB00	0x1ABFC
Utility registers	0x1AC00	0x1ACFC
Reserved	0x1AD00	0xFFFFC

Table 27 shows the Tsi564A register map.

Table 27: Register Map

Offset	Register Name	See
RapidIO Logical Layer and Transport Layer Registers		
00000	RIO_DEV_ID	“RIO Device Identity CAR” on page 136
00004	RIO_DEV_INFO	“RIO Device Information CAR” on page 137
00008	RIO_ASBLY_ID	“RIO Assembly Identity CAR” on page 138
0000C	RIO_ASBLY_INFO	“RIO Assembly Information CAR” on page 139
00010	RIO_PE_FEAT	“RIO Processing Element Features CAR” on page 140
00014	RIO_SW_PORT	“RIO Switch Port Information CAR” on page 142
00018	RIO_SRC_OP	“RIO Source Operation CAR” on page 143
00020 - 00064	Reserved	
00068	RIO_HOST_BASE_ID_LOCK	“RIO Host Base Device ID Lock CSR” on page 145
0006C	RIO_COMP_TAG	“RIO Component Tag CSR” on page 146
0070	RIO_ROUTE_CFG_DESTID	“RIO Route Config DestID CSR” on page 147
0074	RIO_ROUTE_CFG_PORT	“RIO Route Config Output Port CSR” on page 148
0078	RIO_LUT_SIZE	“RIO Route LUT Size CAR” on page 149
007C	RIO_LUT_ATTR	“RIO Route LUT Attributes CSR” on page 150
00080 - 000FC	Reserved	
RapidIO Physical Layer Registers (using extended features ID = 0x0009)		
General Physical Layer Registers		
00100	RIO_SP_MB_HEAD	“RIO 1x/4x-LP Switch Port Maintenance Block Header” on page 152
00104 - 0011C	Reserved	
00120	RIO_SP_LT_CTL	“RIO Switch Port Link Time Out Control CSR” on page 153
00124 - 00138	Reserved	
0013C	RIO_SP_GEN_CTL	“RIO Switch Port General Control CSR” on page 154

Table 27: Register Map *(Continued)*

Offset	Register Name	See
Serial Port 0 Registers (Offset 0x140 - 0x15C)		
00140	SP0_LM_REQ	"RIO Serial Port x Link Maintenance Request CSR" on page 155, offset 0x00 for each Serial Port.
00144	SP0_LM_RESP	"RIO Serial Port x Link Maintenance Response CSR" on page 157, offset 0x04 for each Serial Port
00148	SP0_ACKID_STATUS	"RIO Serial Port x Local AckID Status CSR" on page 159, offset 0x08 for each Serial Port
0014C - 00154	Reserved	
00158	SP0_ERR_STATUS	"RIO Port x Error and Status CSR" on page 161, offset 0x18 for each Serial Port
0015C	SP0_CTL	"RIO Serial Port x Control CSR" on page 164, offset 0x1C for each Serial Port
00160 - 0017C	Serial Port 1	Same set of registers as Serial Port 0, offset 00140 - 0015C.
00180 - 0019C	Serial Port 2	
001A0 - 001BC	Serial Port 3	
001C0 - 001DC	Serial Port 4	
001E0 - 001FC	Serial Port 5	
00200 - 0021C	Serial Port 6	
00220 - 0023C	Serial Port 7	
00340- 00FFC	Reserved	
RapidIO Error Management Extensions (Part VIII of the RapidIO Specification)		
General Error Management Registers		
01000	RIO_ERR_RPT_BH	"RIO Error Reporting Block Header" on page 170, offset 0x00 for each Serial Port
01004 - 01024	Reserved	
01028	RIO_PW_DESTID	"RIO Port-Write Target Device ID CSR" on page 171, offset 0x28 for each Serial Port
0102C - 0103C	Reserved	
Serial Port 0 Registers (Offset 0x1040 - 0x107C)		

Table 27: Register Map *(Continued)*

Offset	Register Name	See
01040	SP0_ERR_DET	"RIO Port x Error Detect CSR" on page 172, offset 0x00 for each Serial Port.
01044	Reserved	
01048	SP0_ERR_ATTR_CAPT_DBG0	"RIO Port x Debug 0" on page 174, offset 0x08 for each Serial Port.
0104C	SP0_ERR_ATTR_CAPT_0_DBG1	"RIO Port x Packet Debug 1" on page 175, offset 0x0C for each Serial Port
01050	SP0_ERR_ATTR_CAPT_1_DBG2	"RIO Port x Packet Debug 2" on page 176, offset 0x10 for each Serial Port.
01054	SP0_ERR_ATTR_CAPT_2_DBG3	"RIO Port x Packet Debug 3" on page 177, offset 0x14 for each Serial Port
01058	SP0_ERR_ATTR_CAPT_3_DBG4	"RIO Port x Packet Debug 4" on page 178, offset 0x18 for each Serial Port.
0105C - 0107C	Reserved	
01080 - 010BC	Serial Port 1 (1x mode)	Same set of registers as Serial Port 0, offset 01040 - 0107C.
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	Serial Port 7 (1x mode)	
014401240 - 0FFFC	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	"RIO Port x Discovery Timer" on page 181
10004	SPBC_MODE	"RIO Port x Mode CSR" on page 183
10008	SPBC_CS_INT_STATUS	"RIO Port x Reset Control Symbol Interrupt CSR" on page 185

Table 27: Register Map *(Continued)*

Offset	Register Name	See
10070	SPBC_ROUTE_CFG_DESTID	"RIO Port x Route Config DestID CSR" on page 186
10074	SPBC_ROUTE_CFG_ PORT	"RIO Port x Route Config Output Port CSR" on page 187
10078 - 1033C	Reserved	
Port-specific copies of the broadcast registers (0x11000 - 0x11FFC). Reads and writes to these registers affect only one port.		
11000 - 110FC	Serial Port 0 (1x/4x mode)	Same set of registers as the 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	Serial Port 7 (1x mode)	
1120011800 - 12FFC	Reserved	
IDT-specific Non-Broadcast Registers		
13000	Reserved	
13004	SP0_CTL_INDEP	"RIO Port x Control Independent Register" on page 188, offset 0x04 for each port
13008-13010	Reserved	
13014	SP0_CS_TX	"RIO Port x Control Symbol Transmit" on page 191, offset 0x14 for each port
13018	SP0_INT_STATUS	"RIO Port x Interrupt Status Register" on page 192, offset 0x18 for each port
1301C	SP0_INT_GEN	"RIO Port x Interrupt Generate Register" on page 194, Offset 0x1C for each port
13020-130AC	Reserved	
130B0	SMAC0_CFG_CH0	"SRIO MAC x SERDES Configuration — Channel 0" on page 196, Offset 0xB0 for each even port

Table 27: Register Map *(Continued)*

Offset	Register Name	See
130B4	SMAC0_CFG_CH1	"SRIO MAC x SERDES Configuration — Channel 1" on page 200, Offset 0xB4 for each even port
130B8	SMAC0_CFG_CH2	"SRIO MAC x SERDES Configuration — Channel 2" on page 203, Offset 0xB8 for each even port
130BC	SMAC0_CFG_CH3	"SRIO MAC x SERDES Configuration — Channel 3" on page 207, Offset 0xBC for each even port
130C0	SMAC0_DLOOP	"SRIO MAC x Digital Loopback Register" on page 210, Offset 0xC0 for each even port
130C4	SMAC0_CLK_SEL	"SRIO MAC x Clock Selection Register" on page 211, Offset 0xC4 for each even port
130C8	SMAC0_PRBS_CTRL	"RIO Serial MAC x PRBS Control Register" on page 213, Offset 0xC8 for each even port
130CC	SMAC0_CH0_PRBS_CTR0	"RIO Serial MAC x PRBS Channel 0 Counter 0" on page 216, Offset 0xCC for each even port
130D0	SMAC0_CH0_PRBS_CTR1	"RIO Serial MAC x PRBS Channel 0 Counter 1" on page 217, Offset 0xD0 for each even port
130D4	SMAC0_CH1_PRBS_CTR0	"RIO Serial MAC x PRBS Channel 1 Counter 0" on page 218, Offset 0xD4 for each even port
130D8	SMAC0_CH1_PRBS_CTR1	"RIO Serial MAC x PRBS Channel 1 Counter 1" on page 219, Offset 0xD8 for each even port
130DC	SMAC0_CH2_PRBS_CTR0	"RIO Serial MAC x PRBS Channel 2 Counter 0" on page 220, Offset 0xDC for each even port
130E0	SMAC0_CH2_PRBS_CTR1	"RIO Serial MAC x PRBS Channel 2 Counter 1" on page 221, Offset 0xE0 for each even port
130E4	SMAC0_CH3_PRBS_CTR0	"RIO Serial MAC x PRBS Channel 3 Counter 0" on page 222, Offset 0xE4 for each even port
130E8	SMAC0_CH3_PRBS_CTR1	"RIO Serial MAC x PRBS Channel 3 Counter 1" on page 223, Offset 0xE8 for each even port
130EC-13EEC	Reserved	

Table 27: Register Map *(Continued)*

Offset	Register Name	See
13100 - 131FC	Serial Port 1(1x Mode)	Same set of registers as Port 0, offset 13000 - 130FC. SMAC registers only present on serial ports 0, 2, 4, 6, 8, 10, 12, and 14
13200 - 132FC	Serial Port 2 (1x/4x mode)	
13300 - 133FC	Serial Port 3 (1x Mode)	
13400 - 134FC	Serial Port 4 (1x/4x Mode)	
13500 - 135FC	Serial Port 5 (1x Mode)	
13600 - 136FC	Serial Port 6 (1x/4x Mode)	
13700 - 137FC	Serial Port 7 (1x Mode)	
1400013800 - 14FFC	Reserved	
Fabric Registers		
1AA00	FAB_TEA_CTL	“Fabric Control Register” on page 229
1AA04	FAB_INT_STAT	“Fabric Interrupt Status Register” on page 230
I^2C Registers		
1AB00	I2C_CTL_1	“I2C Control Register 1” on page 232
1AB04	I2C_CTL_2	“I2C Control Register 2” on page 233
1AB08	I2C_RD_DAT	“I2C Read Data Register” on page 235
1AB0C	I2C_WT_DAT	“I2C Write Data Register” on page 236
Utility Unit Registers		
1AC00	GLOB_INT_STAT	“Global Interrupt Status Register” on page 237
1AC04	GLOB_INT_EN	“Global Interrupt Enable Register” on page 239
1AB10 - 3FFFC	Reserved	

10.4 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 1.2)*, or in RapidIO-approved showings.

When a individual port is powered down, the RapidIO Logical Layer and Transport Layer Registers for that port are read only and return 0.

Register values are maintained when a port is disabled. To disable a port, set PORT_DIS to 1 in the "RIO Serial Port x Control CSR" on page 164.

These registers are reset by the HARD_RST_b reset input signal, as well as when the Tsi564A 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 Tsi564A reset implementation and behavior, see "Clocks, Resets and Power-up Options" on page 103.

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 controller register loading capability, see "I2C Interface" on page 79.

Reads to reserved register addresses return 0, writes to reserved register addresses complete without error and do not affect the operation of the Tsi564A.

10.4.1 RIO Device Identity CAR

This register identifies the device and vendor information for the Tsi564A.

Register name: RIO_DEV_ID Reset value: 0x0568_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	0x0564
16:31	DEV_VEN_ID	Device Vendor Identifier Identifies IDT as the vendor that manufactured the device. This value is assigned by the RapidIO Trade Association.	R	0x000D

10.4.2 RIO 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	Reserved	R	0
24:27	SILICON_REV	Indicates the version of silicon used in the device. 0 = A revision 1 = B revision	R	0
28:31	METAL_REV	Indicates the version of the metal layers for the given silicon version.	R	Undefined

10.4.3 RIO Assembly Identity CAR

This register identifies the following:

- The vendor that manufactured the assembly or subsystem that contains the device.
- The type of assembly that contains the device.

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

10.4.4 RIO 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 Pointer to the first entry in the extended features list.	R	0x0100

10.4.5 RIO Processing Element Features CAR

This register identifies the major functionality provided by the processing element.

Register name: RIO_PE_FEAT Reset value: 0x1000_031F	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	SBR
24:31	Reserved			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	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 (BRDG)).	R	0
3	SW	Switching Capabilities Does the device have the ability to bridge to another external RapidIO interface. 0 = Not capable 1 = Ftype 8 packets with hop count equal to 0 are routed to the register bus.	R	1
4:21	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
22	MC	Multicast 0 = Does not support the multicast extensions 1 = Supports the multicast extensions (unsupported) **Note:** Multicast is not supported in the Tsi564A. For more information on this bit, see *Tsi564A Errata and Design Notes* document.	R	1
23	SBR	System bringup register extension 0 = the system bringup register extension is not supported 1 = the system bringup register extension is supported	R	1
24:26	Reserved	N/A	R	0
27	CTLS	For the Tsi564A, 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 Tsi564A. 0 = Device supports 8 bit destination ID's 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. 001 = Supports 34-bit addresses 011 = Supports 50 and 34-bit addresses 101 = Supports 66 and 34-bit addresses 111 = Supports 66, 50, and 34-bit addresses	R	0b111

10.4.6 RIO 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 serial RapidIO ports are in 1x mode. That is, when a port is configured for 4x mode it consumes two ports from this reported number.	R	0x08
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.	R	Undefined

10.4.7 RIO Source Operation CAR

The Tsi564A can route any type of packet, but it generates only Port-Write packets. Thus, software should ignore all but the PORT_WR field when reading this register.

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

(Continued)

Bits	Name	Description	Type	Reset Value
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

10.4.8 RIO 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 is responsible for initializing 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: 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 end point.	R/W	0xFFFF

The HOST_BASE_ID set in this register does not enforce exclusive access to the device. It is intended to coordinate device identification during initialization and discovery.

10.4.9 RIO Component Tag CSR

This register is written by software. It is used for labeling and identifying the port-write transactions to the host.

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

10.4.10 RIO Route Config DestID CSR

This register and RIO_ROUTE_CFG_PORT 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 the LUT through these registers always read from port 0.

This register set is identical to SPx_ROUTE_CFG_DESTID — Offset 10070 and SPx_ROUTE_CFG_PORT — Offset 10074, except that SPx_ROUTE_CFG_PORT are per-port configuration registers and they include an auto-increment bit to increment the contents of SPx_ROUTE_CFG_DESTID after a read or write operation.

For more information on how to configure the LUTs, see “Lookup Tables” on page 28.

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 “RIO Route Config Output Port CSR” on page 148 is read or written.	R/W	0x00
24:31	CFG_DEST_ID	Configuration Destination ID Specifies the destination ID used to select an entry in the LUT when the “RIO Route Config Output Port CSR” on page 148 r is read or written.	R/W	0x00

10.4.11 RIO Route Config Output Port CSR

This register and RIO_ROUTE_CFG_DESTID 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 the LUT through these registers always read from port 0.

This register set is identical to SPx_ROUTE_CFG_DESTID — Offset 10070 and SPx_ROUTE_CFG_PORT — Offset 10074, except that SPx_ROUTE_CFG_PORT are per-port configuration registers and they include an auto-increment bit to increment the contents of SPx_ROUTE_CFG_DESTID after a read or write operation.

For more information on how to configure the LUTs, see “Lookup Tables” on page 28.

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 messages intended for CONFIG_DESTID in the “RIO Route Config Output Port CSR” on page 148 are sent. Writing a value greater than the PORT_TOTAL field in the “RIO Switch Port Information CAR” on page 142 to this field sets the LUT entry to an unmapped state. For future compatibility, write the value 0xFF to indicate an unmapped destID. When set to unmapped the value in DEFAULT_PORT field in the “RIO Route LUT Attributes CSR” on page 150 is used if a packet is sent to the Tsi564A with the given destination ID.	R/W	Undefined

10.4.12 RIO Route LUT Size CAR

This register tells host software that the Tsi564A supports 512 destination IDs in its Destination ID Lookup Table (LUT).

Register name: RIO_LUT_SIZE **Reset value: 0x0000_01FF**	**Register offset: 0078**

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 be used to route packets through the switch. Destination IDs 0x0000 to LUT_SIZE are valid.	R	0x01FF

10.4.13 RIO Route LUT Attributes 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.

Register name: RIO_LUT_ATTR **Reset value: 0x0000_00FF**	**Register offset: 007C**

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 unmapped destination IDs in the LUT are routed to this pre-defined default output port. DEFAULT_OUT_PORT may be set to “unmapped” with a value greater than the PORT_TOTAL field in the“RIO Switch Port Information CAR” on page 142. For compatibility with future devices, it is recommended that the value 0xFF be used to indicated “unmapped”. If a packet needs to consult the default route and the default route is “unmapped”, the packet is discarded. See “RapidIO Port Numbering” on page 48 for a mapping of port numbers to physical ports.	R/W	0xFF

10.5 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 *Part I: Input/Output Logical Specification*.

When a individual port is powered down, the RapidIO Physical Layer Registers for that port are read only and return 0. Any registers that are exceptions to this are identified in the register tables.

These registers are reset by the HARD_RST_b reset input signal, as well as when the Tsi564A 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 Tsi564A reset implementation and behavior, see “Clocks, Resets and Power-up Options” on page 103. 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 controller register loading capability, see “I2C Interface” on page 79.

Reads to reserved register addresses will return 0, writes to reserved register addresses will complete without error and will not affect the operation of theTsi564A

Table 28: Physical Interface Register Offsets

RIO 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

10.5.1 RIO 1x/4x-LP 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

10.5.2 RIO Switch Port Link Time Out Control CSR

This register contains the time-out timer value for all ports on a device. This time-out 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 time-out interval, and represents between three and six seconds.

When the Link Time Out has expired, the port enters the Output-Error state, as outlined in the *RapidIO Interconnect Specification (Revision 1.2)*.

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	Time-out Interval Value Time-out = (32/F) * TVAL, where F is the PCLK frequency, 100 MHz. When F = 100 MHz, the default value of this register gives a time out of 5.4 seconds. When TVAL is 0, the timer is disabled.	R/W	0xFFFFFF
24:31	Reserved	N/A	R	0

10.5.3 RIO 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 implemented on a device.

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 bit This device has been located by the processing element responsible for system configuration. 1 = Device discovered by system host 0 = Device not discovered	R/W	0
3:31	Reserved	N/A	R	0

10.5.4 RIO Serial Port x Link Maintenance Request CSR

According to the *RapidIO Interconnect Specification (Revision 1.2)* only one link maintenance request can be outstanding at a time. However, the Tsi564A can produce four consecutive link maintenance requests in order to quickly re-establish a link.

Multiple link maintenance request symbols are generated by the CMD field in the RIO Serial Port x Link Maintenance Request CSR. 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.

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.

When the Tsi564A sends its own link maintenance request, and if that Request is outstanding and the CMD field is written to, the register write is ignored. If this register is written twice in rapid succession, it could cause a protocol violation.

If the RIO Serial Port x Link Maintenance Response CSR 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 RIO Switch Port Link Time Out Control CSR, has passed before attempting another link maintenance request to avoid protocol violations.

Register name: SP{0..7}_LM_REQ **Reset value: 0x0000_0000**	**Register offset: 140, 0160, 180, 1A0, 1C0, 0x1E0, 200, 220**

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

(Continued)

Bits	Name	Description	Type	Reset Value
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 will cause the device to send four consecutive reset control symbols. For more information on reset control symbols, see "Reset Control Symbol Processing" on page 36. 100 = Request for status Other values are reserved, but the device will send a control symbol with the requested value.	R/W	0

10.5.5 RIO 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 RIO Serial Port x Link Maintenance Request CSR.

Register name: SP{0..7}_LM_RESP **Reset value: 0x0000_0000**	**Register offset: 144, 164, 184, 1A4, 1C4, 1E4, 204, 224**

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 has been 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 has been 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. This field reflects the OUTSTANDING field in the “RIO Serial Port x Local AckID Status CSR” on page 159. **Note:** The OUTSTANDING field reflects the next expected ackID of the Tsi564A’s link partner; the Tsi564A is the receiver.	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
27:31	LINK_STAT	Link Status Link status field from the link-response control symbol. Other values are reserved. 00010 = Error 00100 = Retry-stopped 00101 = Error-stopped 10000 = OK	R	0

10.5.6 RIO 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.

Register name: SP{0..7}_ACKID_STAT Reset value: 0x0000_0000	Register offset: 148, 168, 0188, 1A8, 1C8, 1E8, 208, 228

Bits	0	1	2	3	4	5	6	7
00:07	Reserved			INBOUND				
08:15	Reserved							
16:23	Reserved			OUTSTANDING				
24:31	Reserved			OUTBOUND				

Bits	Name	Description	Type	Reset Value
0:2	Reserved	N/A	R	0
3:7	INBOUND	Inbound Acknowledge ID Next expected ackID value for the receive side of the port. This bit can be reset to 0 by resetting the port. **Note:** The INBOUND value can be initialized through the I^2C Interface. However, this functionality is required for test purposes only. Unless the INBOUND value is initialized to 0, the device state is not be 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.	R	0
8:18	Reserved	N/A	R	0
19:23	OUTSTANDING	Outstanding Acknowledge IDs The first unacknowledged ackID. **Note:** The Outstanding field on one side of the link should match the Inbound field on the other side of the link.	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. Writing to the OUTBOUND field, is reflected in the OUTSTANDING field. If an error or retry occurs on the inbound while the write to this register is being processed, the OUTBOUND value will not be used for the next packet transmitted. The new OUTBOUND value will always be used when the port is operating normally, and when the port is in an error state. **Note:** For more information on this field, see the *Tsi564A Errata and Design Notes* document.	R/W	0

Changing the OUTBOUND field when there are packets outstanding in the transmit queue results in non-deterministic ackID values being transmitted with the outstanding packets. It is likely that a fatal error due to ackID mismatch will result.

The OUTBOUND field may only be written when there are no packets outstanding in the transmit queue.

10.5.7 RIO 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.

Register name: SP{0..7}_ERR_STATUS Reset value: 0x0000_0001	Register offset: 158, 0178, 198, 1B8, 01D8, 1F8, 218, 238

Bits	0	1	2	3	4	5	6	7
00:07	Reserved					OUTPUT_DROP	Reserved	
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 a TEA error has occurred. Write 1 to clear this bit.	R/W1C	0
6:10	Reserved	N/A	R	0
11	OUTPUT_RE	Output Retry-encountered Outbound port has encountered a retry condition. Set when bit 13, Output Retry-stopped, is set. Write 1 to clear this bit.	R/W1C	0
12	OUTPUT_R	Output Retried Outbound port has 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.	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
13	OUTPUT_RS	Output Retry-stopped Outbound port has received a packet-retry control symbol and is in the output retry-stopped state.	R	0
14	OUTPUT_ERR	Output Error-encountered Outbound port has 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.	R	0
16:20	Reserved	N/A	R	0
21	INPUT_RS	Input Retry-stopped Inbound port is in the input retry-stopped state.	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.	R	0
24:26	Reserved	N/A	R	0
27	PORT_W_PEND	Port-Write Pending Port has encountered a condition which required it to issue an I/O logical port-write maintenance request.This bit is only required if the device can issue a maintenance port-write transaction.	R/W1C	0
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 time-out error for link-response Write 1 to clear this bit.	R/W1C	0

(Continued)

Bits	Name	Description	Type	Reset Value
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.	R	0
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.	R	1

10.5.8 RIO 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 the port is powered down.

Register name: SP{0..7}_CTL Reset value: undefined	Register offset: 15C, 17C, 19C, 1BC, 1DC, 1FC, 0x21C, 23C

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		
16:23	Reserved							
24:31	Reserved						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_MODESEL pin (where x= 0,2,4,6) as follows: • If the SPx_MODESEL pin is high, PORT_WIDTH for SP*x* is 0 and PORT_WIDTH for SP*x+1* is 0. • If the SPx_MODESEL 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.	R	undefined

(Continued)

Bits	Name	Description	Type	Reset Value
2:4	INIT_PWIDTH	Initialization Port Width This field shows the post initialization width for a port that was pre-defined as a 4x port. 000 = 1x port, lane 0. For a port that has downgraded from a 4x to a 1x and lane 0 is the active lane. This is also the indication for a port where PORT_WIDTH is 0x00 when the Spx_MODESEL pin is high, which indicates it was pre-defined as a 1x port. This is the only valid setting for a1x port. 001 = 1x port, lane 2.This value is read when the port has downgraded from a pre-definition 4x port to a post-initialization configuration a 1x port and lane 2 is the active lane. 010 = 4x port. This value is read when the port was pre-defined as a 4x port, and the port-initialized configuration is a 4x port. **Note:** This field is not updated for a port that has OVER_PWIDTH active.	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 (that is, on even numbered ports). May be configured as a power-up option (I^2C) or forced during normal mode of operation (forced re-initialization). 000 = No override 001 = Reserved 010 = Force single lane, lane 0 011 = Force single lane, lane 2 Other values are reserved. Re-initialization can be forced by the FORCE_REINIT field of the SP{0..15}_CTL_INDEP register.	R/W	0
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 outbound queue.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
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.	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.	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, the CRC is not verified and packets are forwarded through the Tsi564A. **Note:** If error checking is disabled, then corrupt maintenance packets may be accepted by the Tsi564A. However, if error checking is disabled any writes to the Tsi564A registers are ignored. If error checking is enabled, corrupt maintenance packets are not accepted.	R/W	0
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 (unsupported) **Note:** Multicast is not supported in the Tsi564A.	R	0
13:29	Reserved	N/A	R	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. All received packets return packet-not-accepted control symbols to force the sending device to signal an error condition.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
31	PORT_TYPE	Port Type Indicates the port type 1 = Serial port	R	1

Table 29 shows the width and re-initialization of the override port.

Table 29: Override Port Width and Re-initialization

Mode	From	To	Action
1x/4x, SPx_MODESEL is low. Valid for serial ports 0, 2, 4, 6 Serial ports 1, 3, 5, 7 are not used in this mode and can be powered down.	000	000	Ignored
	xxx	001 100 - 111	Ignored, reserved value
	000	010	Re-initialize and force lane 0
	000	011	Re-initialize and force lane 2
	010	011	Re-initialize and force lane 2
	011	010	Re-initialize and force lane 0
	010	000	Re-initialize as a 4x port
	011	000	Re-initialize as a 4x port
1x, SPx_MODESEL is high Valid for all serial ports (0..7).	000	000	Ignored
	xxx	001 100 - 111	Ignored, reserved value
	000	010	Re-initialize and force lane 0
	000	011	Re-initialize and force lane 2
	010	011	Re-initialize and force lane 2
	011	010	Re-initialize and force lane 0
	010	000	Re-initialize as a 1x port.
	011	000	Re-initialize as a 1x port.

10.6 RapidIO Error Management Extension Registers

This section describes the registers in the Extended Features (EF_ID = 0x0007), which is defined in *Part VIII of the RapidIO Specification*. This functionality allows an external processing element to manage the error status and reporting for a processing element.

These registers are reset by the HARD_RST_b reset input signal, as well as when the Tsi564A 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 Tsi564A reset implementation and behavior., see "Clocks, Resets and Power-up Options" on page 103. It is possible to override reset values of writable fields using the I^2C register loading capability on boot. For more information on the use of I^2C controller register loading capability, see "I2C Interface" on page 79.

The Logical/Transport Error detect registers are not required for a switch. However, a switch's register bus access errors and transport errors are reported per port in bit 0 of the RIO 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.

Register values are maintained when a port is disabled. To disable a port, set PORT_DIS to 1 in the "RIO Serial Port x Control CSR" on page 164.

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 Tsi564A.

Not all Error Management Extension registers are supported in the Tsi564A.

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 30: Error Management Registers

Port	Offset	Description
All	0x1000	General Error Management capability registers
SP0	0x1040	1x/4x Serial port
SP1	0x1080	1x Serial port
SP2	0x10C0	1x/4x Serial port

Table 30: Error Management Registers *(Continued)*

Port	Offset	Description
SP3	0x1100	1x Serial port
SP4	0x1140	1x/4x LSerial port
SP5	0x1180	1x Serial port
SP6	0x11C0	1x/4x Serial port
SP7	0x1200	1x Serial port

10.6.1 RIO Error Reporting Block Header

The error reporting block header indicates the start of the Error Management Extensions registers in the Tsi564A.

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

10.6.2 RIO 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 switch.

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

10.6.3 RIO Port x Error Detect CSR

This register indicates transmission errors that are detected by the hardware.

Register name: SP{0..7}_ERR_DET Reset value: 0x0000_0000	Register offset: 1040, 1080, 10C0, 1100, 1140, 1180, 11C0, 1200

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_ ACKID	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 the following illegal field errors: • Reserved transport type (TT) detected • Ftype 8 packet (maintenance packet) has a ttype other than read or write • LUT parity error • LUT unmapped destID • TEA Error • Fatal error (PORT_ERR in RIO Port x Error and Status CSR) • Maximum retry threshold exceeded **Note:** Clearing this bit will also clear the following interrupt status bits in the “RIO Port x Interrupt Status Register” on page 192: • MAX_RETRY • TVAL_EXPIRED • TEA	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

(Continued)

Bits	Name	Description	Type	Reset Value
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.	R/W0C	0
11	CS_NOT_ACC	Received packet-not-accepted control symbol.	R/W0C	0
12	PKT_ILL_ACKID	Received packet with unexpected ackID.	R/W0C	0
13	PKT_CRC_ERR	Received a packet with a CRC error.	R/W0C	0
14	PKT_ILL_SIZE	Received packet exceeds 276 bytes.	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.	R/W0C	0
27	PROT_ERR	Protocol Error Received control symbol is unexpected.	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.	R/W0C	0
30	CS_ACK_ILL	Received an unexpected acknowledge control symbol	R/W0C	0
31	LINK_TO	An acknowledge or Link-response is not received within the specified time-out interval (see the “RIO Switch Port Link Time Out Control CSR” on page 153). The Capture register does not capture information for this error.	R/W0C	0

10.6.4 RIO Port x Debug 0

In debug mode this register is unlocked and all its fields are used for writing the content of the debug packet.

Register name: SP{0..7}_ERR_ATTR_CAPT_DBG0 Reset value: 0x0000_0000	Register offset: 1048, 1088, 10C8, 1108, 1148, 1188, 11C8, 1208

Bits	0	1	2	3	4	5	6	7
00:07	DBG0[0:7]							
08:15	DBG0[8:15]							
16:23	DBG0[16:23]							
24:31	DBG0[24:31]							

Bits	Name	Description	Type	Reset Value
0:31	DBG0	Debug mode This field is the content of debug packet	R/W	0

10.6.5 RIO Port x Packet Debug 1

This register is unlocked in debug mode.

The "RIO Port x Error Detect CSR" on page 172 is writable by software to allow software debug of the system error recovery and threshold mechanism.

Register name: SP{0..7}_ERR_CAPT_0_DBG1 **Reset value: 0x0000_0000**	**Register offset: 104C, 108C, 010CC, 110C, 114C, 118C, 11CC, 120C**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:7	DBG1[0:7]							
8:15	DBG1[8:15]							
16:23	DBG1[16:23]							
24:31	DBG1[24:31]							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:31	DBG1	Character and control symbol or bytes 0 to 3 of packet header.	R/W	0

10.6.6 RIO Port x Packet Debug 2

See “Debug Packet and Control Symbol Generator” on page 42 for more information on debug packet generation.

Register name: SP{0..7}_ERR_CAPT_1_DBG2 Reset value: 0x0000_0000	Register offset: 1050, 1090, 10D0, 1110, 1150, 1190, 11D0, 1210

Bits	0	1	2	3	4	5	6	7
00:7	DBG2[0:7]							
8:15	DBG2[8:15]							
16:23	DBG2[16:23]							
24:31	DBG2[24:31]							

Bits	Name	Description	Type	Reset Value
0:31	DBG2	Byte 4 to 7 of the packet header	R/W	0

10.6.7 RIO Port x Packet Debug 3

See "Debug Packet and Control Symbol Generator" on page 42 for more information on debug packet generation.

Register name: SP{0..7}_ERR_CAPT_2_DBG3 Reset value: 0x0000_0000	Register offset: 1054, 1094, 10D4, 1114, 1154, 1194, 11D4, 1214

Bits	0	1	2	3	4	5	6	7
0:7	DBG3[0:7]							
8:15	DBG3[8:15]							
16:23	DBG3[16:23]							
24:31	DBG3[24:31]							

Bits	Name	Description	Type	Reset Value
0:31	DBG3	Byte 8 to 11 of the packet header	R/W	0

10.6.8 RIO Port x Packet Debug 4

See “Debug Packet and Control Symbol Generator” on page 42 for more information on debug packet generation.

Register name: SP{0..7}_ERR_CAPT_3_DBG4 Reset value: 0x0000_0000	Register offset: 1058, 1098, 10D8, 1118, 1158, 1198, 11D8, 1218

Bits	0	1	2	3	4	5	6	7
0:7	DBG4[0:7]							
8:15	DBG4[8:15]							
16:23	DBG4[16:23]							
24:31	DBG4[24:31]							

Bits	Name	Description	Type	Reset Value
0:31	DBG4	Byte 12 to 15 of the packet header.	R/W	0

10.7 IDT-Specific RapidIO Registers

The registers in this section are specific to IDT's switching products.Table 31 shows IDT-specific RapidIO Registers not defined in the *RapidIO Interconnect Specification (Revision 1.2)*.

10.7.1 Powering Ports Up and Down

Individual ports can be powered up or down through specific registers.

10.7.1.1 Powering Down

When a individual port is powered down, the IDT-Specific RapidIO Registers are read only and return 0 with the exception of "RIO Port x Error and Status CSR" on page 161 and "RIO Serial Port x Control CSR" on page 164, both of which return 0x00000001 when read.

10.7.1.2 Powering Up

When an individual port is powered down and powered up, all IDT-Specific RapidIO Registers must be reprogrammed. This includes those registers which support broadcast addresses in the address range 0x10000 through 0x100FF.

10.7.2 Disabling Ports

Register values are maintained when a port is disabled. To disable a port, set PORT_DIS to 1 in the "RIO Serial Port x Control CSR" on page 164.

10.7.3 Register Reset

These registers are reset by the HARD_RST_b reset input signal, as well as when the Tsi564A 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 Tsi564A reset implementation and behavior, see "Clocks, Resets and Power-up Options" on page 103.

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 controller register loading capability, see "I2C Interface" on page 79.

Table 31: 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

Table 31: IDT-Specific RapidIO Registers *(Continued)*

Port	Register Offset	Description
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

10.7.4 RIO Port x Discovery Timer

Where x refers to a broadcast register (BC) and ports 0 through 7.

This register defines discovery-timer value for the serial ports in 4x mode. It also defines the port-write timer for all ports.

- Offset 10000: (BC) Broadcast to all ports
- Offset 11000-11F00: Port 0-7 specific

Register name: SP{BC,0..7}_DISCOVERY_TIMER **Reset value: 0x9000_0000**	**Register offset: 10000, 11000, 11100, 11200, 11300, 11400, 11500, 11600, 11700**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	DISCOVERY_TIMER				Reserved			
08:15	PW_TIMER				Reserved			
16:23	Reserved							
24:31	Reserved							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:3	DISCOVERY_ TIMER	Discovery Timer This field is used by serial 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 discovery timer has a value of 12 +/- 4 ms. Register bus frequency - 100 MHz • 0: 32 cycles of sys_clk (debug only) • 1: 1.31 ms • 2: 1.31ms * 2 = 2.62 ms • .. • 9: 1.31ms * 9= 11.8 ms (default) • ... • 15: 1.31ms *15= 19.65 ms	R/W	9
4:7	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
8:11	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. Register bus frequency = 100 MHz • 0000 = Disabled. Port-Write is sent once only per event. • 0001 = 167ms - 334ms • 0010 = 334ms - 501ms • 0100 = 668ms - 835ms • 1000 = 1.33s - 1.5s • 1111 = 1.28-2.56us (Debug only) Other values are reserved.	R/W	0
12:31	Reserved	N/A	R	0

10.7.5 RIO Port x Mode CSR

Where x refers to a broadcast register (BC) and ports 0 through 7.

This register defines the mode of operation for the ports, and contains the interrupt enables for the reset control symbol.

- Offset 10004: (BC) Broadcast to all ports
- Offset 11004-11F04: Port 0-7 specific

Register name: SP{BC,0..7}_MODE **Reset value: 0x0200_0000**	**Register offset: 10004, 011004, 11104,11204, 11304, 11404, 11504, 11604, 11704**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved		IDLE_ERR_DIS	Reserved	PW_DIS	Reserved	SELF_RST	Reserved
08:15	Reserved							
16:23	Reserved							
24:31	Reserved							RCS_CS_EN

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:1	Reserved	N/A	R	0
2	IDLE_ERR_DIS	Idle Error Checking Disable 0 = Error checking is enabled (default). If one or more data characters are sent (Dx.y characters not delimited with start of packet/end of packet control symbols) in an idle sequence, the device enters the Input Error stopped state. 1 = Ignore all not idle or invalid characters in the idle sequence.	R/W	0
3	Reserved	N/A	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	N/A	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: Interrupt signal is asserted (if RCS_INT_EN is also asserted) 1 = Enable: Device is reset	R/W	1
7:30	Reserved	N/A	R	0
31	RCS_ CS_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. For more information on global interrupts, see “Global Interrupt Enable Register” on page 239.	R/W	0

10.7.6 RIO Port x Reset Control Symbol Interrupt CSR

This register contains the interrupt status for reset control symbols.

Where x refers to a broadcast register (BC) and ports 0 through 7.

- Offset 10008: (BC) Broadcast to all ports
- Offset 11008-11708: Specific Port

Register name: SP{BC,0..7}_CS_INT_STATUS **Reset value: 0x0000_0000**	**Register offset: 10008, 11008, 11108, 11208, 11308, 11408, 11508, 11608, 11708**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							
24:31	Reserved							RCS

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:30	Reserved	N/A	R	0
31	RCS	Reset Control Symbol Received Interrupt Status Indicates that four consecutive Reset control symbols have been received on the port (or on any port when the SPBC_MCR_INT register is read). All RST_CS_IRQs from all ports are ORed together. The GLOB_INT_STAT register shows the status of the combined RST_IRQs from all ports. Write 1 to clear this bit. Writing 1 to this bit in the SPBC_MCR_INT register clears the interrupt on all the ports.	R/W1C	0

10.7.7 RIO Port x Route Config DestID CSR

This register and SPx_ROUTE_CFG_PORT operate together to provide indirect read and write access to the LUTs.

They are identical to RIO_ROUTE_CFG_DESTID and RIO_ROUTE_CFG_PORT, except that SPx_ROUTE_CFG_PORT 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..7}_ROUTE_CFG_DESTID **Reset value: 0x0000_0000**	**Register offset: 10070, 11070, 11170, 11270, 11370, 11470, 11570, 11670, 11770**

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 destination ID when the destination ID is used to perform either a read or a write, through the “RIO Port x Route Config Output Port CSR” on page 187.	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 is used to route 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 “RIO Route Config Output Port CSR” on page 148 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 “RIO Route Config Output Port CSR” on page 148 is read or written. This value increments by one for every write to the RIO Port x Route Config Output Port CSR when the AUTO_INC bit is set	R/W	0x00

10.7.8 RIO Port x Route Config Output Port CSR

This register and "RIO Port x Route Config DestID CSR" on page 186 operate together to provide indirect read and write access to the LUTs.

They are identical to RIO_ROUTE_CFG_DESTID and RIO_ROUTE_CFG_PORT, except that SPx_ROUTE_CFG_PORT 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..7}_ROUTE_CFG_PORT **Reset value: Undefined**	**Register offset: 10074, 11074, 11174, 11274, 11374, 11474, 11574, 11674, 11774**

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 intended for CONFIG_DESTID are sent. Writing a value greater or equal to PORT_TOTAL sets the LUT entry to an unmapped state. For future compatibility, 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

10.7.9 RIO Port x Control Independent Register

This register is used for error recovery.

Register name: SP{0..7}_CTL_INDEP Reset value: 0x0100_0000	Register offset: 13004, 13104, 13204, 13304, 13404, 13504, 13604, 13704

Bits	0	1	2	3	4	5	6	7
00:07	Reserved					FORCE_ REINIT	Reserved	TRANS_ MODE
08:15	DEBUG_ MODE	SEND_ DBG_PKT	Reserved				LINK_INIT _NOTIFIC ATION_EN	LUT_PAR_ ERR_EN
16:23	RETRY_THR							
24:31	Reserved	IRQ_EN	MAX_ RETRY_ EN	Reserved				TEA_EN

Bits	Name	Description	Type	Reset Value
0:4	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	Reserved	N/A	R	0
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 switch as soon as the routing information is received. In store and forward mode, the incoming packet is not sent to the switch fabric until the whole packet is received. **Note:** Cut-through mode may have detrimental effect on overall switch performance if ports are running at different speeds. This is because in cut-through mode, a slower port can use the internal switching fabric for a long time relative to a faster port, incurring additional latency and potentially throughput loss on the faster port.	R/W	1

(Continued)

Bits	Name	Description	Type	Reset Value
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.	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.	R/W	0
10:13	Reserved	N/A	R	0
14	LINK_INIT_NOTI FICATION_EN	Enables interrupts and port writes for LINK_INIT_NOTIFICATION events. 0 = Interrupt and port write disabled. 1 = Interrupt and port write enabled. See the “RIO Port x Interrupt Status Register” on page 192.	R/W	0
15	LUT_PAR_ERR_ EN	Enables interrupts for parity errors in the lookup table. 0 = Interrupt disabled 1 = Interrupt enabled See the “RIO Port x Interrupt Status Register” on page 192.	R/W	0
16:23	RETRY_ THR	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 switch generates a port-write and sends the LOG_ERR bit in the RIO Port x Error Detect CSR. 00 = Disable the RETRY_ERROR reporting 01 = Set the RETRY_THRESHOLD to 1 02 = Set the RETRY_THRESHOLD to 2 ... FF = Set the RETRY_THRESHOLD to 255	R/W	0x00
24	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
25	IRQ_EN	Interrupt Error Report Enable If enabled, the interrupt signal is high when the IRQ_ERR bit is set to 1, see the "RIO Port x Interrupt Status Register" on page 192	R/W	0
26	MAX_RETRY_EN	RETRY_THR Report Enable If enabled, the port-write and interrupt report an error when a packet is retried RETRY_THR times in a row and the MAX_RETRY bit is set in the "RIO Port x Interrupt Status Register" on page 192.	R/W	0
27:30	Reserved	N/A	R	0
31	TEA_EN	Transfer Error Acknowledge Enable An interrupt is generated if the internal switching fabric times out trying to send the packet to its egress port.	R/W	0

10.7.10 RIO Port x Control Symbol Transmit

Where x refers to ports 0 through 7.

Writing to this register transmits a single control symbol to RapidIO. This register is used for debug purposes only.

All control symbol fields are defined according to the *RapidIO Interconnect Specification (Revision 1.2)*. The control symbol's CRC field is generated by hardware.

Register name: SP{0..7}_CS_TX **Reset value: 0x0000_0000**	**Register offset: 13014, 13114, 13214, 13314, 13414, 13514, 13614, 13714**

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

10.7.11 RIO Port x Interrupt Status Register

Where x refers to ports 0 through 7.

Register name: SP{0..7}_INT_STATUS Reset value: 0x0000_0000	Register offset: 13018, 13118, 13218, 13318, 13418, 13518, 13618, 13718

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved						LINK_INIT_NOTIFICATION	LUT_PAR_ERR
16:23	Reserved							
24:31	Reserved	IRQ_ERR	MAX_RETRY	Reserved				TEA

Bits	Name	Description	Type	Reset Value
0:13	Reserved	N/A	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 "RIO Serial Port x Control CSR" on page 164, and a link has initialized according to the PORT_OK bit in "RIO Port x Error and Status CSR" on page 161, the LINK_INIT_NOTIFICATION is set to 1. To stop the LINK_INIT_NOTIFICATION bit from being set, PORT_LOCKOUT must be set to 0 and/or the link must no longer be in an initialized state.	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 "RIO Port x Error Detect CSR" on page 172.	R/W1C	0
16:24	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
25	IRQ_ERR	Interrupt Error Status Set to one if an error occurs and there is a port-write condition. Once set, remains set until written with logic 1 to clear. Setting this bit generates an interrupt if the IRQ_EN bit in the"RIO Port x Control Independent Register" on page 188 is set.	R/W1C	0
26	MAX_ RETRY	Maximum retry threshold exceeded. MAX_RETRY is set when a packet is retried RETRY_THR times in a row. The Port-Write request and interrupt are generated if enabled. The setting of this bit generates a Port-Write and interrupt if a bit MAX_RETRY_EN bit is set bit in the "RIO Port x Control Independent Register" on page 188 is set. This bit is will not be set if RETRY_THR is 00. This bit is cleared by writing a 1 to it, or by clearing all bits in the "RIO Port x Error Detect CSR" on page 172.	R/W1C	0
27:30	Reserved	N/A	R	0
31	TEA	This interrupt is raised when the internal switching fabric has timed out before it could deliver a packet to an egress port.	R/W1C	0

10.7.12 RIO Port x Interrupt Generate Register

This register can be used to generate the corresponding error in the "RIO Port x Interrupt Status Register" on page 192.

Register name: SP{0..7}_INT_GEN Reset value: 0x0000_0000	Register offset: 1301C, 1311C, 1321C, 1331C, 1341C, 1351C, 1361C, 1371C

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved						LINK_INIT_NOTIFICATION	LUT_PAR_ERR
16:23	Reserved							
24:31	Reserved	IRQ_ERR	MAX_RETRY	Reserved				TEA

Bits	Name	Description	Type	Reset Value
0:13	Reserved	N/A	R	0
14	LINK_INIT_NOTIFICATION	Forces the LINK_INIT_NOTIFICATION bit in the "RIO Port x Interrupt Status Register" on page 192 to be set. Bit always reads as zero.	R/W1S	0
15	LUT_PAR_ERR	Forces the LUT_PAR_ERR bit in the "RIO Port x Interrupt Status Register" on page 192 to be set. Bit always reads as zero.	R/W1S	0
16:24	Reserved	N/A	R	0
25	IRQ_ERR	Forces the IRQ_ERR bit in the "RIO Port x Interrupt Status Register" on page 192 to be set. Bit always reads as zero.	R/W1S	0
26	MAX_RETRY	Forces the MAX_RETRY bit of in the "RIO Port x Interrupt Status Register" on page 192. Bit always reads as zero.	R/W1S	0
27:30	Reserved	N/A	R	0
31	TEA	Forces the TEA bit of the corresponding interrupt status register to be set to 1. Bit always reads as zero.	R/W1S	0

10.8 Serial Port Electrical Layer Registers

The Serial Port Electrical Layer Registers are not defined in the *RapidIO Interconnect Specification (Revision 1.2)*. They are specific to IDT's switching products.

These registers are reset by the HARD_RST_b reset input signal, as well as when the Tsi564A 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 Tsi564A reset implementation and behavior, see "Clocks, Resets and Power-up Options" on page 103.

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 controller register loading capability, see "I2C Interface" on page 79.

Software must not access reserved addresses or bits; the reserved registers may affect device operation in non-deterministic ways.

The registers in Table 32 are accessible even when the serial RapidIO ports are in reset or powered down.

Table 32: 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

10.8.1 SRIO MAC x SERDES Configuration — Channel 0

This register is used to control serial port SerDes channel 0. For more information on port configuration, see “Port Power Down” on page 53.

Register name: SMAC{0,2,4,6}_CFG_CH0 **Reset value: 0x0000_FFFF**	**Register offset: 130B0, 132B0, 134B0, 136B0**

Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
00:07	LOWSIG	ALB_EN	DRV_STRENGTH		DEQ			
08:15	DTX				PWDN	RESET	RESET_ PLL	FRM_DIS
16:23	DLT_EN	DLT_THRESH						
24:31	DLT_THRESH							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0	LOWSIG	Low Signal 0 = Normal operation 1= SerDes is receiving a weak signal from the link (<VLOS - min 85 mv). The received data is not forced to any particular state when this happens. This is a status bit only. No actions are taken by the design during this condition.	R	0
1	ALB_EN	Analog Equipment Loopback Enable. 0 = Normal operation 1 = SerDes Tx is looped back to Rx for this lane. Changes to this register field must be completed when the SerDes is in reset. Wait 500 ns after changing this register field before de-asserting reset. For more information, see “Port Loopback Testing” on page 58.	R/W	0
2:3	DRV_ STRENGTH	Drive Strength 00 = Nominal Driver Current = 20 mA 01 = Nominal Driver Current = 10 mA 10 = Nominal Driver Current = 28 mA 11 = Reserved For more information, see “Programmable Driver Current and Equalization” on page 57.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
4:7	DEQ	Tx driver pre-emphasis settings: 0000 = equalization current/driver current = 0.00 0001 = equalization current/driver current = 0.04 0010 = equalization current/driver current = 0.08 0011 = equalization current/driver current = 0.12 0100 = equalization current/driver current = 0.16 0101 = equalization current/driver current = 0.20 0110 = equalization current/driver current = 0.24 0111 = equalization current/driver current = 0.28 1000 = equalization current/driver current = 0.32 1001 = equalization current/driver current = 0.36 1010 = equalization current/driver current = 0.40 1011 = equalization current/driver current = 0.44 1100 = equalization current/driver current = 0.48 1101 = equalization current/driver current = 0.52 1110 = equalization current/driver current = 0.60 1111 = equalization current/driver current = 0.65 For more information, see “Programmable Driver Current and Equalization” on page 57.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
8:11	DTX	Tx driver current setting: 0000 = actual current/ nominal current = 1.00 0001 = actual current/ nominal current = 1.05 0010 = actual current/ nominal current = 1.10 0011 = actual current/ nominal current = 1.15 0100 = actual current/ nominal current = 1.20 0101 = actual current/ nominal current = 1.25 0110 = actual current/ nominal current = 1.30 0111 = actual current/ nominal current = 1.35 1000 = actual current/ nominal current = 0.60 1001 = actual current/ nominal current = 0.65 1010 = actual current/ nominal current = 0.70 1011 = actual current/ nominal current = 0.75 1100 = actual current/ nominal current = 0.80 1101 = actual current/ nominal current = 0.85 1110 = actual current/ nominal current = 0.90 1111 = actual current/ nominal current = 0.95 For more information, see “Programmable Driver Current and Equalization” on page 57.	R/W	0
12	PWDN	Power Down Mode Asynchronous power-down signal for channel 0 1 = Power-down 0 = Normal mode of operation This bit powers down the channel but does not power down or disable the entire port.	R/W	0
13	RESET	Asynchronous reset signal for channel 0 digital logic 1 = Reset mode 0 = Normal mode of operation	R/W	0
14	RESET_ PLL	Asynchronous reset signal for PLL of channel 0 and 1 When asserted high, RESET_PLL initializes the PLL circuitry. 1 = Reset mode 0 = Normal mode of operation This bit is only present for SerDes channels 0 and 2.	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
15	FRM_DIS	Framer Disable 0 = Normal operation 1 = Disable the framer. This should only be completed when performing a BERT test.	R/W	0
16	DLT_EN	Dead Link Timer Enable 0 = Disabled 1 = When enabled, this timer is used to determine when a link is powered up and enabled, but dead (that is, there is no link partner responding). When a link is declared dead, the transmitting port on the Tsi564A will remove 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 MAC -- this feature is not limited to lane 0 of the SerDes despite the register it happens to be located in.	R/W	1
17:31	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.	R/W	0x7FFF

10.8.2 SRIO MAC x SERDES Configuration — Channel 1

This register is used to control serial port SerDes channel 1. For more information on port configuration, see “Port Power Down” on page 53.

Register name: SMAC{0,2,4,6}_CFG_CH1 Reset value: 0x0000_0000	Register offset: 130B4, 132B4, 134B4, 136B4

Bits	0	1	2	3	4	5	6	7
00:07	LOWSIG	ALB_EN	DRV_STRENGTH		DEQ			
08:15	DTX				PWDN	RESET	Reserved	FRM_DIS
16:23	Reserved							
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0	LOWSIG	When 1 indicates SerDes is receiving a weak signal from the link (<VLOS - min 85 mv). The received data is not forced to any particular state when this happens. This is a status bit only. No actions are taken by the design during this condition.	R	0
1	ALB_EN	Analog Equipment Loopback Enable. 0 = Normal operation 1 = SerDes Tx is looped back to Rx for this lane. Changes to this register field must be completed when the SerDes is in reset. Wait 500ns after changing this register field before de-asserting reset. For more information, see “Port Loopback Testing” on page 58.	R/W	0
2:3	DRV_ STRENGTH	00 = Nominal Driver Current = 20 mA 01 = Nominal Driver Current = 10 mA 10 = Nominal Driver Current = 28 mA 11 = Reserved For more information, see “Programmable Driver Current and Equalization” on page 57.	R/W	00

(Continued)

Bits	Name	Description	Type	Reset Value
4:7	DEQ	Tx driver pre-emphasis settings: 0000 = equalization current/driver current = 0.00 0001 = equalization current/driver current = 0.04 0010 = equalization current/driver current = 0.08 0011 = equalization current/driver current = 0.12 0100 = equalization current/driver current = 0.16 0101 = equalization current/driver current = 0.20 0110 = equalization current/driver current = 0.24 0111 = equalization current/driver current = 0.28 1000 = equalization current/driver current = 0.32 1001 = equalization current/driver current = 0.36 1010 = equalization current/driver current = 0.40 1011 = equalization current/driver current = 0.44 1100 = equalization current/driver current = 0.48 1101 = equalization current/driver current = 0.52 1110 = equalization current/driver current = 0.60 1111 = equalization current/driver current = 0.65 For more information, see “Programmable Driver Current and Equalization” on page 57.	R/W	0000

(Continued)

Bits	Name	Description	Type	Reset Value
8:11	DTX	Tx driver current settings: 0000 = actual current/ nominal current = 1.00 0001 = actual current/ nominal current = 1.05 0010 = actual current/ nominal current = 1.10 0011 = actual current/ nominal current = 1.15 0100 = actual current/ nominal current = 1.20 0101 = actual current/ nominal current = 1.25 0110 = actual current/ nominal current = 1.30 0111 = actual current/ nominal current = 1.35 1000 = actual current/ nominal current = 0.60 1001 = actual current/ nominal current = 0.65 1010 = actual current/ nominal current = 0.70 1011 = actual current/ nominal current = 0.75 1100 = actual current/ nominal current = 0.80 1101 = actual current/ nominal current = 0.85 1110 = actual current/ nominal current = 0.90 1111 = actual current/ nominal current = 0.95 For more information, see “Programmable Driver Current and Equalization” on page 57.	R/W	0000
12	PWDN	Power Down Mode Asynchronous power-down signal for channel 1. 1 = Power-down 0 = Normal mode of operation **Note:** This bit does not affect whether a port is disabled, or powered down.	R/W	0
13	RESET	Asynchronous reset signal for channel 1 digital logic. 1 = Reset mode 0 = Normal mode of operation	R/W	0
14	Reserved	N/A	R	0
15	FRM_DIS	Framer Disable 0 = Normal operation 1 = Disable the framer. This should only be completed when performing a BERT test.	R/W	0
16:31	Reserved	N/A	R	0

10.8.3 SRIO MAC x SERDES Configuration — Channel 2

This register is used to control serial port SerDes channel 2.For more information on port configuration, see “Port Power Down” on page 41.

Register name: SMAC{0,2,4,6}_CFG_CH2 **Reset value: 0x0000_0000**	**Register offset: 130B8, 132B8, 134B8, 136B8**

Bits	0	1	2	3	4	5	6	7
00:07	LOWSIG	ALB_EN	DRV_STRENGTH		DEQ			
08:15	DTX				PWDN	RESET	RESET_ PLL	FRM_DIS
16:23	Reserved							
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0	LOWSIG	When 1 indicates SerDes is receiving a weak signal from the link (<VLOS - min 85 mv). The received data is not forced to any particular state when this happens. This is a status bit only. No actions are taken by the design during this condition.	R	0
1	ALB_EN	Analog Equipment Loopback Enable. 0 = Normal operation 1 = SerDes Tx is looped back to Rx for this lane. Changes to this register field must be completed when the SerDes is in reset. Wait 500ns after changing this register field before de-asserting reset. For more information, see “Port Loopback Testing” on page 58.	R/W	0
2:3	DRV_ STRENGTH	00 = Nominal Driver Current = 20 mA 01 = Nominal Driver Current = 10 mA 10 = Nominal Driver Current = 28 mA 11 = Reserved For more information, see “Programmable Driver Current and Equalization” on page 57.	R/W	00

(Continued)

Bits	Name	Description	Type	Reset Value
4:7	DEQ	Tx driver pre-emphasis settings: 0000 = equalization current/driver current = 0.00 0001 = equalization current/driver current = 0.04 0010 = equalization current/driver current = 0.08 0011 = equalization current/driver current = 0.12 0100 = equalization current/driver current = 0.16 0101 = equalization current/driver current = 0.20 0110 = equalization current/driver current = 0.24 0111 = equalization current/driver current = 0.28 1000 = equalization current/driver current = 0.32 1001 = equalization current/driver current = 0.36 1010 = equalization current/driver current = 0.40 1011 = equalization current/driver current = 0.44 1100 = equalization current/driver current = 0.48 1101 = equalization current/driver current = 0.52 1110 = equalization current/driver current = 0.60 1111 = equalization current/driver current = 0.65 For more information, see “Programmable Driver Current and Equalization” on page 57.	R/W	0000

(Continued)

Bits	Name	Description	Type	Reset Value
8:11	DTX	Tx driver current settings: 0000 = actual current/ nominal current = 1.00 0001 = actual current/ nominal current = 1.05 0010 = actual current/ nominal current = 1.10 0011 = actual current/ nominal current = 1.15 0100 = actual current/ nominal current = 1.20 0101 = actual current/ nominal current = 1.25 0110 = actual current/ nominal current = 1.30 0111 = actual current/ nominal current = 1.35 1000 = actual current/ nominal current = 0.60 1001 = actual current/ nominal current = 0.65 1010 = actual current/ nominal current = 0.70 1011 = actual current/ nominal current = 0.75 1100 = actual current/ nominal current = 0.80 1101 = actual current/ nominal current = 0.85 1110 = actual current/ nominal current = 0.90 1111 = actual current/ nominal current = 0.95 For more information, see “Programmable Driver Current and Equalization” on page 57.	R/W	0000
12	PWDN	Power Down Mode Asynchronous power-down signal for channel 0 1 = Power-down 0 = Normal mode of operation This bit does not affect whether a port is disabled, or powered down.	R/W	0
13	RESET	Asynchronous reset signal for channel digital logic 0 1 = Reset mode 0 = Normal mode of operation	R/W	0
14	RESET_ PLL	Asynchronous reset signal for PLL of channel 2 and 3. When asserted high, RESET_PLL initializes the PLL circuitry. 1 = Reset mode 0 = Normal mode of operation (This bit is present for channels 0 and 2 only.)	R/W	0

(Continued)

Bits	Name	Description	Type	Reset Value
15	FRM_DIS	Framer Disable 0 = Normal operation 1 = Disable the framer. This should only be completed when performing a BERT test.	R/W	0
16:31	Reserved	N/A	R	0

10.8.4 SRIO MAC x SERDES Configuration — Channel 3

This register is used to control serial port SerDes channel 3. For more information on port configuration, see "Port Power Down" on page 53.

Register name: SMAC{0,2,4,6}_CFG_CH3 Reset value: 0x0000_0000	Register offset: 130BC, 132BC, 134BC, 136BC

Bits	0	1	2	3	4	5	6	7
00:07	LOWSIG	ALB_EN	DRV_STRENGTH		DEQ			
08:15	DTX				PWDN	RESET	Reserved	FRM_DIS
16:23	Reserved							
24:31	Reserved							

Bits	Name	Description	Type	Reset Value
0	LOWSIG	Low Signal 0 = Normal operation 1 = SerDes is receiving a weak signal from the link (<VLOS - min 85 mv). The received data is not forced to any particular state when this happens. This is a status bit only. No actions are taken by the design during this condition.	R	0
1	ALB_EN	Analog Equipment Loopback Enable. 0 = Normal operation 1 = SerDes Tx is looped back to Rx for this lane. Changes to this register field must be completed when the SerDes is in reset. Wait 500ns after changing this register field before de-asserting reset. For more information, see "Port Loopback Testing" on page 58.	R/W	0
2:3	DRV_ STRENGTH	00 = Nominal Driver Current = 20 mA 01 = Nominal Driver Current = 10 mA 10 = Nominal Driver Current = 28 mA 11 = Reserved For more information, see "Programmable Driver Current and Equalization" on page 57.	R/W	00

(Continued)

Bits	Name	Description	Type	Reset Value
4:7	DEQ	Tx driver pre-emphasis settings: 0000 = equalization current/driver current = 0.00 0001 = equalization current/driver current = 0.04 0010 = equalization current/driver current = 0.08 0011 = equalization current/driver current = 0.12 0100 = equalization current/driver current = 0.16 0101 = equalization current/driver current = 0.20 0110 = equalization current/driver current = 0.24 0111 = equalization current/driver current = 0.28 1000 = equalization current/driver current = 0.32 1001 = equalization current/driver current = 0.36 1010 = equalization current/driver current = 0.40 1011 = equalization current/driver current = 0.44 1100 = equalization current/driver current = 0.48 1101 = equalization current/driver current = 0.52 1110 = equalization current/driver current = 0.60 1111 = equalization current/driver current = 0.65 For more information, see “Programmable Driver Current and Equalization” on page 57.	R/W	0000

(Continued)

Bits	Name	Description	Type	Reset Value
8:11	DTX	Tx driver current settings: 0000 = actual current/ nominal current = 1.00 0001 = actual current/ nominal current = 1.05 0010 = actual current/ nominal current = 1.10 0011 = actual current/ nominal current = 1.15 0100 = actual current/ nominal current = 1.20 0101 = actual current/ nominal current = 1.25 0110 = actual current/ nominal current = 1.30 0111 = actual current/ nominal current = 1.35 1000 = actual current/ nominal current = 0.60 1001 = actual current/ nominal current = 0.65 1010 = actual current/ nominal current = 0.70 1011 = actual current/ nominal current = 0.75 1100 = actual current/ nominal current = 0.80 1101 = actual current/ nominal current = 0.85 1110 = actual current/ nominal current = 0.90 1111 = actual current/ nominal current = 0.95 For more information, see “Programmable Driver Current and Equalization” on page 57.	R/W	0000
12	PWDN	Power Down Mode Asynchronous power-down signal for channel 3 0 = Normal mode of operation 1 = Power-down This bit does not affect whether a port is disabled, or powered down.	R/W	0
13	RESET	Asynchronous reset signal for channel 3 digital logic 0 = Normal mode of operation 1 = Reset mode	R/W	0
14	Reserved	N/A	R	0
15	FRM_DIS	Framer Disable 0 = Normal mode of operation 1 = Disable the framer. This bit should only be 1 when performing a BERT test.	R/W	0
16:31	Reserved	N/A	R	0

10.8.5 SRIO MAC x Digital Loopback Register

This register enables digital equipment loopback (TX -> RX) on a per port basis.

Register name: SMAC{0,2,4,6}_DLOOP Reset value: 0x0000_0000	Register offset: 130C0, 132C0, 134C0, 136C0

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							
24:31	Reserved						DLB_ ODD_EN	DLB_ EVEN_EN

Bits	Name	Description	Type	Reset Value
0:29	Reserved	N/A	R	0
30	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. The loopback path does not include the 8b/10B encoder/decoder. For more information, see “Port Loopback Testing” on page 58.	R/W	0
31	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. For more information, see “Port Loopback Testing” on page 58.	R/W	0

10.8.6 SRIO MAC x Clock Selection Register

This register selects the clock speed for the serial RapidIO ports. It is loaded with the value of the clock selection pins during reset. The register may then be overwritten by software.

Register name: SMAC{0,2,4,6}_CLK_SEL Reset value: Undefined	Register offset: 130C4, 132C4, 134C4, 136C4

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							
08:15	Reserved							
16:23	Reserved							
24:31	Reserved		RST_X1	RST_X4	PWDN_X1	PWDN_X4	SCLK_SEL	

Bits	Name	Description	Type	Reset Value
0:25	Reserved	N/A	R	0
26	RST_X1	Software Reset Control Odd-numbered Port Software 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.	R/W	0
27	RST_X4	Software 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 certain registers. For more information, see “Serial RapidIO Electrical Interface” on page 47.	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 reflects the sampled value of the SPx_PWDN pin. Writing to this register overrides the configuration provided by the pin. 0 = Normal mode of operation 1 = Port powered down The port will be powered down if this bit is set or if the PWDN bits are set in the "SRIO MAC x SERDES Configuration — Channel 0" on page 196. For more information, see "Port Power Down" on page 53.	R/W	Undefined
29	PWDN_X4	Power down control for both the even-numbered and odd-numbered ports using this MAC. This field initially reflects the sampled value of the SPx_PWDN pin. 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 The port will be powered down if this bit is set or if the PWDN bits are set in the "SRIO MAC x SERDES Configuration — Channel 0" on page 196. For more information, see "Port Power Down" on page 53.	R/W	Undefined
30:31	SCLK_SEL	Clock select: Reflects the value on SP_IO_SPEED after reset. 00 = S_CLK_2 reference divided by 2 (default) 01 = S_CLK_2 reference 10 =S_CLK_1 reference 11 = reserved Writing to this register overrides the power up value of SP_IO_SPEED speed selection.	R/W	Undefined

10.8.7 RIO Serial MAC x PRBS Control Register

For more information on using this register, see "Bit Error Rate Testing (BERT)" on page 48.

Register name: SMAC{0,2,4,6}_PRBS_CTRL Reset value: 0x0000_0000	Register offset: 130C8, 132C8, 134C8, 136C8

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 = Line loop back 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 hence 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/W	0
8	Reserved	N/A	R	0
9:11	PATTERN_ SEL2	Channel 2 Transmitter Pattern Select	R/W	0
12	START_ PRBS2	Channel 2 Start PRBS	R/W	0
13	SYNC_ PRBS2	Channel 2 SYNC PRBS	R/W1S	0
14	STOP_ PRBS2_CTR	Channel 2 STOP PRBS	R/W	0
15	CLR_PRBS2 _CTR	Channel 2 CLEAR PRBS	R/W	0
16	Reserved	N/A	R	0
17:19	PATTERN_ SEL1	Channel 1 Transmitter Pattern Select	R/W	0
20	START_ PRBS1	Channel 1 Start PRBS	R/W	0
21	SYNC_ PRBS1	Channel 1 SYNC PRBS	R/W1S	0
22	STOP_ PRBS1_CTR	Channel 1 STOP PRBS	R/W	0
23	CLR_PRBS1 _CTR	Channel 1 CLEAR PRBS	R/W	0
24	Reserved	N/A	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
25:27	PATTERN_ SEL0	Channel 0 Transmitter Pattern Select	R/W	0
28	START_ PRBS0	Channel 0 Start PRBS	R/W	0
29	SYNC_ PRBS0	Channel 0 SYNC PRBS	R/W1S	0
30	STOP_ PRBS0_CTR	Channel 0 STOP PRBS	R/W	0
31	CLR_PRBS0 _CTR	Channel 0 CLEAR PRBS	R/W	0

10.8.8 RIO Serial MAC x PRBS Channel 0 Counter 0

The PRBS counter 0 and 1 are used to calculate Bit Error Rate (BERT) of the link. The PRBS generator must be stopped by writing 1 to the STOP_PRBS0_CTR bit field in the "RIO Serial MAC x PRBS Control Register" on page 213 before reading this register.

Register name: SMAC{0,2,4,6}_CH0_PRBS_CTR0 **Reset value: 0x0000_0000**	**Register offset: 130CC, 132CC, 134CC, 136CC**

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 all zeros or by writing 1 to the CLR_PRBS_CTR field in the "RIO Serial MAC x PRBS Control Register" on page 213.	R	0x0000
16:31	PRBS_ERR_CTR	Error counter for PRBS checker. This counter is cleared by writing all zeros or by writing 1 to the CLR_PRBS_CTR field in the "RIO Serial MAC x PRBS Control Register" on page 213.	R	0x0000

10.8.9 RIO Serial MAC x PRBS Channel 0 Counter 1

The PRBS counter 0 and 1 are used to calculate BERT. The PRBS generator must be stopped by writing 1 to the STOP_PRBS0_CTR bit field in the "RIO Serial MAC x PRBS Control Register" on page 213 before reading this register.

Register name: SMAC{0,2,4,6}_CH0_PRBS_CTR1 **Reset value: 0x0000_0000**	**Register offset: 130D0, 132D0, 134D0, 136D0**

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 all zeros or by writing 1 to the CLR_PRBS_CTR field in the "RIO Serial MAC x PRBS Control Register" on page 213.	R	0x0000_0000

10.8.10 RIO Serial MAC x PRBS Channel 1 Counter 0

The PRBS counter 0 and 1 are used to calculate BERT. The PRBS generator must be stopped by writing 1 to the STOP_PRBS1_CTR bit field in the "RIO Serial MAC x PRBS Control Register" on page 213 before reading this register.

Register name: SMAC{0,2,4,6}_CH1_PRBS_CTR0 Reset value: 0x0000_0000	Register offset: 130D4, 132D4, 134D4, 136D4

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 all zeros or by writing 1 to the CLR_PRBS_CTR field in the "RIO Serial MAC x PRBS Control Register" on page 213.	R	0x0000
16:31	PRBS_ERR_CTR	Error counter for PRBS checker. This counter is cleared by writing all zeros or by writing 1 to the CLR_PRBS_CTR field in the "RIO Serial MAC x PRBS Control Register" on page 213.	R	0x0000

10.8.11 RIO Serial MAC x PRBS Channel 1 Counter 1

The PRBS counter 0 and 1 are used to calculate BER. The PRBS generator must be stopped by writing 1 to the STOP_PRBS1_CTR bit field in the "RIO Serial MAC x PRBS Control Register" on page 213 before reading this register.

Register name: SMAC{0,2,4,6}_CH1_PRBS_CTR1 **Reset value: 0x0000_0000**	**Register offset: 130D8, 132D8, 134D8, 136D8**

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 all zeros or by writing 1 to the CLR_PRBS_CTR field in the "RIO Serial MAC x PRBS Control Register" on page 213.	R	0x0000_0000

10.8.12 RIO Serial MAC x PRBS Channel 2 Counter 0

The PRBS counter 0 and 1 are used to calculate BERT. The PRBS generator must be stopped by writing 1 to the STOP_PRBS2_CTR bit field in the "RIO Serial MAC x PRBS Control Register" on page 213 before reading this register.

Register name: SMAC{0,2,4,6}_CH2_PRBS_CTR0 Reset value: 0x0000_0000	Register offset: 130DC, 132DC, 134DC, 136DC

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 all zeros or by writing 1 to the CLR_PRBS_CTR field in the "RIO Serial MAC x PRBS Control Register" on page 213.	R	0x0000
16:31	PRBS_ERR_CTR	Error counter for PRBS checker. This counter is cleared by writing all zeros or by writing 1 to the CLR_PRBS_CTR field in the "RIO Serial MAC x PRBS Control Register" on page 213.	R	0x0000

10.8.13 RIO Serial MAC x PRBS Channel 2 Counter 1

The PRBS counter 0 and 1 are used to calculate BERT. The PRBS generator must be stopped by writing 1 to the STOP_PRBS2_CTR bit field in the "RIO Serial MAC x PRBS Control Register" on page 213 before reading this register.

Register name: SMAC{0,2,4,6}_CH2_PRBS_CTR1 **Reset value: 0x0000_0000**	**Register offset: 130E0, 132E0, 134E0, 136E0,**

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 all zeros or by writing 1 to the CLR_PRBS_CTR field in the "RIO Serial MAC x PRBS Control Register" on page 213.	R	0x0000_0000

10.8.14 RIO Serial MAC x PRBS Channel 3 Counter 0

The PRBS counter 0 and 1 are used to calculate BERT. The PRBS generator must be stopped by writing 1 to the STOP_PRBS3_CTR bit field in the "RIO Serial MAC x PRBS Control Register" on page 213 before reading this register.

Register name: SMAC{0,2,4,6}_CH3_PRBS_CTR0 Reset value: 0x0000_0000	Register offset: 130E4, 132E4, 134E4, 136E4

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 all zeros or by writing 1 to the CLR_PRBS_CTR field in the "RIO Serial MAC x PRBS Control Register" on page 213.	R	0x0000
16:31	PRBS_ERR_CTR	Error counter for PRBS checker. This counter is cleared by writing all zeros or by writing 1 to the CLR_PRBS_CTR field in the "RIO Serial MAC x PRBS Control Register" on page 213.	R	0x0000

10.8.15 RIO Serial MAC x PRBS Channel 3 Counter 1

The PRBS counter 0 and 1 are used to calculate BERT. The PRBS generator must be stopped by writing 1 to the STOP_PRBS3_CTR bit field in the "RIO Serial MAC x PRBS Control Register" on page 213 before reading this register.

Register name: SMAC{0,2,4,6}_CH3_PRBS_CTR1 **Reset value: 0x0000_0000**	**Register offset: 130E8, 132E8, 134E8, 136E8**

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 all zeros or by writing 1 to the CLR_PRBS_CTR field in the "RIO Serial MAC x PRBS Control Register" on page 213.	R	0x0

10.8.16 RIO Serial MAC x BERT Data Register for Channel 0

This register is used during a BERT test to define a repeating fixed pattern sent by the transmitter.

Register name: SMAC{0,2,4,6}_CH0_BERT_DATA Reset value: 0x0000_0000	Register offset: 130EC, 132EC, 134EC, 136EC

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
0:5	Reserved	N/A	R	0
6:15	PAT1	Pattern 1 See text for PAT0.	R/W	0
16:21	Reserved	N/A	R	0
22:31	PAT0	Pattern 0 During a BERT test caused by setting the PATTERN_SEL field to 0b100 in the “RIO Serial MAC x PRBS Control Register” on page 213, 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 8B/10B code groups. The pattern repeats until the BERT is disabled through the “RIO Serial MAC x PRBS Control Register” on page 213. For more information on the usage model of this register, see “Bit Error Rate Testing (BERT)” on page 62.	R/W	0

10.8.17 RIO Serial MAC x BERT Data Register for Channel 1

This register is used during a BERT test to define a repeating fixed pattern sent by the transmitter.

Register name: SMAC{0,2,4,6}_CH1_BERT_DATA Reset value: 0x0000_0000	Register offset: 130F0, 132F0, 134F0, 136F0

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
0:5	Reserved	N/A	R	0
6:15	PAT1	Pattern 1 See text for PAT0.	R/W	0
16:21	Reserved	N/A	R	0
22:31	PAT0	Pattern 0 During a BERT test caused by setting the PATTERN_SEL field to 0b100 in the “RIO Serial MAC x PRBS Control Register” on page 213, 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 8B/10B code groups. The pattern repeats until the BERT is disabled through the “RIO Serial MAC x PRBS Control Register” on page 213. For more information on the usage model of this register, see Section 3.9 on page 62.	R/W	0

10.8.18 RIO Serial MAC x BERT Data Register for Channel 2

This register is used during a BERT test to define a repeating fixed pattern sent by the transmitter.

Register name: SMAC{0,2,4,6}_CH2_BERT_DATA Reset value: 0x0000_0000	Register offset: 130F4, 132F4, 134F4, 136F4

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
0:5	Reserved	N/A	R	0
6:15	PAT1	Pattern 1 See text for PAT0.	R/W	0
16:21	Reserved	N/A	R	0
22:31	PAT0	Pattern 0 During a BERT test caused by setting the PATTERN_SEL field to 0b100 in the “RIO Serial MAC x PRBS Control Register” on page 213, 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 8B/10B code groups. The pattern repeats until the BERT is disabled through the “RIO Serial MAC x PRBS Control Register” on page 213. For more information on the usage model of this register, see “Bit Error Rate Testing (BERT)” on page 62.	R/W	0

10.8.19 RIO Serial MAC x BERT Data Register for Channel 3

This register is used during a BERT test to define a repeating fixed pattern sent by the transmitter.

Register name: SMAC{0,2,4,6}_CH3_BERT_DATA Reset value: 0x0000_0000	Register offset: 130F8, 132F8, 134F8, 136F8

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
0:5	Reserved	N/A	R	0
6:15	PAT1	Pattern 1 See text for PAT0.	R/W	0
16:21	Reserved	N/A	R	0
22:31	PAT0	Pattern 0 During a BERT test caused by setting the PATTERN_SEL field to 0b100 in the “RIO Serial MAC x PRBS Control Register” on page 213, 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 8B/10B code groups. The pattern repeats until the BERT is disabled through the “RIO Serial MAC x PRBS Control Register” on page 213. For more information on the usage model of this register, see “Bit Error Rate Testing (BERT)” on page 62.	R/W	0

10.9 Internal Switching Fabric (ISF) Registers

The Internal Switching Fabric (ISF) Registers are not defined in the *RapidIO Interconnect Specification (Revision 1.2)*. They are specific to IDT's switching products.

These registers provide control and status information concerning time-out errors in data crossing the internal switching fabric.

These registers are reset by the HARD_RST_b reset input signal, as well as when the Tsi564A 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 Tsi564A reset implementation and behavior, see "Clocks, Resets and Power-up Options" on page 103. 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 controller register loading capability, "I2C Interface" on page 79.

Software must not access reserved addresses or bits; the reserved registers may affect device operation in non-deterministic ways.

10.9.1 Fabric Control Register

The TEA signal is asserted when a timeout is detected on the 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							
08:15	Reserved						TEA_INT_EN	TEA_EN
16:23	TEA_OUT[0:7]							
24:31	TEA_OUT[8:15]							

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:13	Reserved	N/A	R	0x3C0
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 Signal Out One is added to this value and it is multiplied by 2^15 to determine the number of ISF clock cycles a request waits for an acknowledge before a transaction error acknowledge (TEA) occurs. For example, assume the 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 + 1) * 2^15 * 6.4 ns = 107.6 ms. A value of 0x0000 disables the TEA timer.	R/W	0x0200

10.9.2 Fabric 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							
24:31	PORT7_ IRQ	PORT6_ IRQ	PORT5_ IRQ	PORT4_ IRQ	PORT3_ IRQ	PORT2_ IRQ	PORT1_ IRQ	PORT0_ IRQ

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:23	Reserved	N/A	R	0
24	PORT7_IRQ	Serial port 7 IRQ Indicates that a TEA has occurred on this port. Writing a 1 to this bit clears it and causes the IRQ signal to be de-asserted.	R/W1C	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

10.10 I²C Registers

The I²C Registers are not defined in the *RapidIO Interconnect Specification (Revision 1.2)*. They are specific to IDT's switching products.

This register provides the control and data capability of the general purpose I²C interface controller. This register are not accessible during operation of an EEPROM load sequence.

These registers are reset by the HARD_RST_b reset input signal, as well as when the Tsi564A performs a self-reset. For more information on Tsi564A reset implementation and behavior, see "Clocks, Resets and Power-up Options" on page 103. It is possible to override reset values of writable fields, and some read-only fields, using the I²C register loading capability on boot. For more information on the use of I²C controller register loading capability, see "I2C Interface" on page 79.

Software must not access reserved addresses or bits; the reserved registers may affect device operation in non-deterministic ways.

10.10.1 I²C Control Register 1

This register is used to control the I²C interface. The primary function of the I²C controller is to initialize the control registers of the Tsi564A at start-up.

Register name: I2C_CTL_1 Reset value: 0x0000_000A	Register offset: 1AB00

Bits	0	1	2	3	4	5	6	7
00:07	Reserved							RW
08:15	BYTE_ADDR							
16:23	Reserved					PAGE_SEL		
24:31	Reserved				DEV_CODE			

Bits	Name	Description	Type	Reset Value
0:6	Reserved	N/A	R	0
7	RW	I²C Read or Write 0 = Read from I²C memory 1 = Write to I²C memory	R/W	0
8:15	BYTE_ ADDR	I²C Address Eight address bits which select a single memory location within a page of I²C memory.	R/W	0
16:20	Reserved	N/A	R	0
21:23	PAGE_SEL	Page Select Used to select a page of memory within an I²C device type. Each page may contain up to 256 bytes.	R/W	0
24:27	Reserved	N/A	R	0
28:31	DEV_CODE	Device Code Identifies type of device on I²C bus to establish serial communications with. Device Code for an I²C serial EEPROM is binary 1010.	R/W	0xA

10.10.2 I²C Control Register 2

This register is used to control the I²C interface. The primary function of the I²C Controller is to initialize the control registers of the Tsi564A at start-up.

Register name: I2C_CTL_2 Reset value: 0x0000_0000	Register offset: 1AB04

Bits	0	1	2	3	4	5	6	7
00:07	Reserved			I2C_ DONE	CFG_ ERR	TO_ERR	Reserved	
08:15	Reserved						RD_STAT	WR_STAT
16:23	Reserved							START
24:31	Reserved			MULTI_ BYTE	LANE		SIZE	

Bits	Name	Description	Type	Reset Value
0:2	Reserved	N/A	R	0
3	I2C_ DONE	Interrupt status bit indicating whether an I2C read or write has completed successfully. 0 = No interrupt 1 = I2C operation has completed	R/W1C	0
4	CFG_ ERR	Interrupt status bit indicating that a configuration error has occurred. 0 = No error 1 = The first two bytes read from EPROM indicate that more than 255 registers are to be written.	R/W1C	0
5	TO_ERR	Interrupt status bit indicating when an I2C operation has been attempted but an acknowledge signal was not received. 0 = No error 1 = Interrupt asserted	R/W1C	0
6:13	Reserved	N/A	R	0
14	RD_ STAT	Read Status Bit Equals 1 while an I²C read operation is in progress. Clears when operation ends.	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
15	WR_ STAT	Write Status Bit Equals 1 while an I^2C write operation is in progress. Clears when operation ends.	R	0
16:22	Reserved	N/A	R	0
23	START	Start Control Set to 1 to start an I^2C operation. Clears when I^2C operation ends.	R/W	0
24:26	Reserved	N/A	R	0
27	MULTI_ BYTE	0 = Each byte of an I^2C write transaction is sent in a new address phase 1 = All bytes (up to four) of a write transaction are sent in a single address phase.	R/W	0
28:29	LANE	Data Lane Reads four bytes from adjacent I^2C memory locations, writes four bytes. Size — Lane — Location within Register: 00 — 00 — byte 0 00 — 01 — byte 1 00 — 10 — byte 2 00 — 11 — byte 3 01 — 00 — bytes 0 and 1 01 — 01 — bytes 0 and 1 01 — 10 — bytes 2 and 3 01 — 11 — bytes 2 and 3 10 — N/A — bytes 0, 1, 2 and 3 11 — N/A — bytes 0, 1, 2 and 3	R/W	0
30:31	SIZE	Number of bytes in an I^2C read or write. The Tsi564A only supports I^2C reads and writes of four bytes. 00 = 1 byte (not supported) 01 = 2 bytes (not supported) 10 = 4 bytes (3 byte access not possible) 11 = 4 bytes	R/W	0

10.10.3 I²C Read Data Register

This register contains information about the I²C interface read data.

Register name: I2C_RD_DATA Reset value: 0x0000_0000	Register offset: 1AB08

Bits	0	1	2	3	4	5	6	7
00:07	BYTE3							
08:15	BYTE2							
16:23	BYTE1							
24:31	BYTE0							

Bits	Name	Description	Type	Reset Value
0:7	BYTE3	Received I²C data, byte 3	R	0x00
8:15	BYTE2	Received I²C data, byte 2	R	0x00
16:23	BYTE1	Received I²C data, byte 1	R	0x00
24:31	BYTE0	Received I²C data, byte 0	R	0x00

10.10.4 I²C Write Data Register

This register contains information about the I²C interface write data.

Register name: I2C_WR_DATA Reset value: 0x0000_0000	Register offset: 1AB0C

Bits	0	1	2	3	4	5	6	7
00:07	BYTE3							
08:15	BYTE2							
16:23	BYTE1							
24:31	BYTE0							

Bits	Name	Description	Type	Reset Value
0:7	BYTE3	Transmitted I²C data, byte 3	R/W	0x00
8:15	BYTE2	Transmitted I²C data, byte 2	R/W	0x00
16:23	BYTE1	Transmitted I²C data, byte 1	R/W	0x00
24:31	BYTE0	Transmitted I²C data, byte 0	R/W	0x00

10.11 Utility Unit Registers

The utility unit contains global registers for interrupts and clocking.

10.11.1 Global Interrupt Status Register

This register indicates which block within the Tsi564A has generated an interrupt. The interrupt requests from a given block are “ORed” together and the value of the output is reflected 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	Reserved	I2C	TEA
08:15	Reserved							
16:23	Reserved							
24:31	PORT7	PORT6	PORT5	PORT4	PORT3	PORT2	PORT1	PORT0

Bits	**Name**	**Description**	**Type**	**Reset Value**
0:3	Reserved	N/A	R	0
4	RCS	Combined 4 Reset Control Symbols interrupt status from all ports	R	0
5	Reserved	N/A	R	0
6	I2C	I^2C Interrupt Port	R	0
7	TEA	TEA occurred in fabric Refer to the “Fabric Control Register” on page 229 to determine what port(s) experienced the TEA.	R	0
8:23	Reserved	N/A	R	0
24	PORT7	Port 7 Interrupt	R	0
25	PORT6	Port 6 Interrupt	R	0
26	PORT5	Port 5 Interrupt	R	0
27	PORT4	Port 4 Interrupt	R	0
28	PORT3	Port 3 Interrupt	R	0

(Continued)

Bits	Name	Description	Type	Reset Value
29	PORT2	Port 2 Interrupt	R	0
30	PORT1	Port 1 Interrupt	R	0
31	PORT0	Port 0 Interrupt	R	0

10.11.2 Global Interrupt Enable Register

This register allows an internal interrupt request to signal an external interrupt via the INT_b pin.

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	Reserved	I2C	TEA
08:15	Reserved							
16:23	Reserved							
24:31	PORT7	PORT6	PORT5	PORT4	PORT3	PORT2	PORT1	PORT0

Bits	Name	Description	Type	Reset Value
0:3	Reserved	N/A	R	0
4	RCS	Four Reset Control Symbols Interrupt Enable	R/W	0
5:6	Reserved	N/A	R	0
6	I2C	I^2C Interrupt Port Enable	R/W	0
7	TEA	TEA interrupt enable	R/W	0
8:23	Reserved	N/A	R	0
24	PORT7	Port 7 Interrupt Enable	R/W	0
25	PORT6	Port 6 Interrupt Enable	R/W	0
26	PORT5	Port 5 Interrupt Enable	R/W	0
27	PORT4	Port 4 Interrupt Enable	R/W	0
28	PORT3	Port 3 Interrupt Enable	R/W	0
29	PORT2	Port 2 Interrupt Enable	R/W	0
30	PORT1	Port 1 Interrupt Enable	R/W	0
31	PORT0	Port 0 Interrupt Enable	R/W	0

A. Serial RapidIO Protocol Overview

The *RapidIO Physical Layer 1x/4x LP-Serial Specification* addresses the physical layer requirements for devices utilizing an electrical serial connection medium. This specification defines a full duplex serial physical layer interface (link) between devices using unidirectional differential signals in each direction. Further, it allows ganging of four serial links for applications requiring higher link performance. It also defines a protocol for link management and packet transport over a link.

RapidIO systems are comprised of end point processing elements and switch processing elements. The RapidIO interconnect architecture is partitioned into a layered hierarchy of specifications which includes the *Logical*, *Common Transport*, and *Physical* layers. The Logical layer specifications define the operations and associated transactions by which end point processing elements communicate with each other. The Common Transport layer defines how transactions are routed from one end point processing element to another through switch processing elements. The Physical Layer defines how adjacent processing elements electrically connect to each other. RapidIO packets are formed through the combination of bit fields defined in the Logical, Common Transport, and Physical Layer specifications. The Tsi564A fully manages the end to end link on each port.

A.1 Protocol

The RapidIO Physical Layer 1x/4x LP-Serial specification defines the protocol for packet delivery between serial RapidIO devices including packet and control symbol transmission, flow control, error management, and other device to device functions. A particular device may not implement all of the mode selectable features presented in the RapidIO specification.

The 1x/4x serial physical layer specification has the following properties:

- Embeds the transmission clock with data using an 8B/10B encoding scheme.
- Supports one serial differential pair, referred to as one lane, or four ganged serial differential pairs, referred to as four lanes, in each direction.
- Allows switching packets between RapidIO 1x/4x LP-Serial Ports and RapidIO Physical Layer 8/16 LP-LVDS ports without requiring packet manipulation.
- Uses similar retry and error recovery protocols as the RapidIO Physical Layer 8/16 LP-LVDS specification.
- Supports transmission rates of 1.25, 2.5, and 3.125 Gbaud/s (data rates of 1.0, 2.0, and 2.5 Gbit/s) per lane.

A.2 Packets

A RapidIO 1x/4x serial packet is formed by prefixing a 10-bit physical layer header to the combined RapidIO transport and logical layer bit fields followed by an appended 16-bit CRC field. The sum of all of the bit fields adds 20 bytes to the encapsulated data packet size. The maximum data field size is 256 bytes resulting in a maximum packet size of 276 bytes.

A.2.1 Control Symbols

Two classes of control symbols are defined (stype0 and stype1) and are used for packet acknowledgment, link utility functions, link maintenance, and packet delineation. A control symbol is a 24-bit entity (including a 5-bit CRC code). The control symbol is used for packet delineation by placement at the beginning of a packet. The control symbol may also be embedded within a packet for message passing and link status notification as well as sent when the link is idle.

Acknowledgment control symbols are used by processing elements to indicate packet transmission status. Utility control symbols are used to communicate buffer status and link recovery synchronization. Link maintenance control symbols are used by link partner devices to communicate physical layer status, synchronization requests, and device reset.

A.3 Physical Layer

The physical layer is broken into two sub-layers, the PCS and PMA Layers describes the Physical Coding Sub-layer (PCS) functionality as well as the Physical Media Attachment (PMA) functionality.

A.3.1 PCS Layer

The PCS layer functionality includes 8B/10B encoding scheme for embedding the clock with the data. It also manages the transmission rules for the 1x and 4x interfaces and defines the link initialization sequence for clock synchronization. The PCS function is also responsible for idle sequence generation, encoding for transmission and lane striping, and decoding, lane alignment and de-striping on reception.

The PCS layer also provides mechanisms for determining the operational mode of the port as 4-lane or 1-lane operation, and means to detect link states. It provides for clock difference tolerance between the sender and receiver without requiring flow control.

A.3.2 PMA Layer

The PMA (Physical Medium Attachment) function is responsible for serializing 10-bit parallel code-groups to/from a serial bit stream on a lane-by-lane basis. Upon receiving data, the PMA function provides alignment of the received bit stream to 10-bit code-group boundaries, completed independently on a lane-by-lane basis. It then provides a continuous stream of 10-bit code-groups to the PCS, one stream for each lane. The 10-bit code-groups are not observable by layers higher than the PCS.

A.3.3 Physical Protocol

The physical connection of a RapidIO link is managed by a series of control symbols transmitted on a transmit/response basis. These control symbols are made up of 10-bit encoded special characters and 3 byte control symbols. Encoded 8-bit characters are given encoding values and names that allow easy and unique detection of the character. Detailed explanation of the encoding values and names can be found in *"PCS and PMA Layers" of "Part VI Physical Layer 1x/4x LP-Serial RapidIO Specification"*.

Table 33 contains the list of control characters that pertain to link management.

The Table 33 illustrates the Special Characters and their function as it applies to the serial protocol.

Table 33: Special Characters and Encoding

Code Group	Use	Number of Groups	Encoding	8-bit Value
/PD/	packet delimiter	1	/K28.3/	0x7C
/SC/	start of Control Symbol	1	/K28.0/	0x1C
/I/	1x Idle	/K/ or /R/ or /A/	see below	0xBC or 0xFD or 0xFB
/K/	1x Sync	1	/K28.5/	0xBC
/R/	1x Skip	1	/K29.7/	0xFD
/A/	1x Align	1	/K27.7/	0xFB
//I//	Idle Column	//K// or //R// or //A//	see below	0xBC or 0xFD or 0xFB
//K//	4x Idle	4	/K28.5/K28.5/K28.5/K28.5/	0xBC on each lane
//R//	4x Sync	4	/K29.7/K29.7/K29.7/K29.7/	0xFD on each lane
//A//	4x Skip	4	/K27.7/K27.7/K27.7/K27.7/	0xFB on each lane

Table 34 illustrates the control symbol construction in 8-bit values. Further detail on the usage of the control symbols may be found in *Part VI Physical Layer 1x/4x LP-Serial RapidIO Specification"*.

Table 34: Control Symbol Construction

3-bits	---	5-bits	5-bits	3-bits	---	3-bits	5-bits	--
stype0 [0-2]	definition	P-0	P-1	stype1 [0-2]		cmd	CRC	
000	pkt-accepted	pkt-ackID	buf_status					
001	pkt-rtry	pkt-ackID	buf_status					
010	pkt-not-accepted	pkt-ackID	cause (see below)					
011	reserved	---	---					
100	status	ackID_status	buf_status					
101	reserved	---	---					
110	link-response	ackID_status	port_status (see below)					
111	reserved	---	---					
		pkt-not-accepted cause [0-4]		definition				
			00000	reserved				
			00001	recvd unexpected ackID on pkt				
			00010	recvd a ctrl symbol w/ bad CRC				
			00011	non-maint pkt reception is stopped				
			00100	recvd pkt w/ bad CRC				
			00101	recvd invalid char or valid but illegal char				
			000110-11110	reserved				
			11111	general error				

Table 34: Control Symbol Construction

3-bits	---	5-bits	5-bits	3-bits	---	3-bits	5-bits	--
		link response port_status [0-4]		definition				
			00000	reserved				
			00001	reserved				
			00010	unrecoverable error unable to accept pkts				
			00011	reserved				
			00100	retry-stopped state				
			00101	error-stopped state				
			00110-01111	reserved				
			10000	OK				
			10001-11111	reserved				
				stype1				
				stype1[0-2]	definition	cmd[0-2]	cmd function	pkt delimiter
				000	start of pkt	000	reserved	yes
				001	stomp	000	reserved	yes
				010	end of pkt	000	reserved	yes
				011	restart from rtry	000	reserved	N/A
				100	link req	000-010	reserved	N/A
						011	reset device	N/A
						100	input status	N/A
						101-111	reserved	N/A
				101	multicast event	000	reserved	no
				110	reserved	000	reserved	no
				111	NOP	000	reserved	no

B. Ordering Information

This appendix discusses ordering information for the Tsi564A.

B.1 Ordering Information

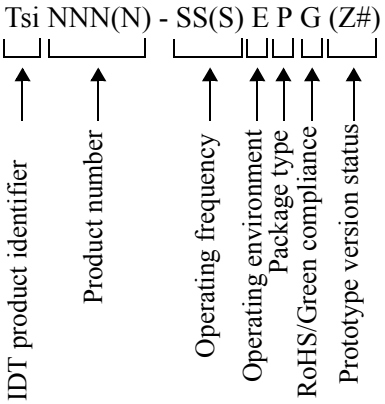
When ordering the Tsi564A please refer to the device by its full part number, as displayed in Table 35.

Table 35: Ordering Information

Part Number	Frequency	Temperature	Package	Pin Count
TSI564A-10GCL	1.25 - 3.125 Gbit/s	Commercial	FCBGA	399
TSI564A-10GCLY	1.25 - 3.125 Gbit/s	Commercial	FCBGA (RoHS)	399
TSI564A-10GIL	1.25 - 3.125 Gbit/s	Industrial	FCBGA	399
TSI564A-10GILY	1.25 - 3.125 Gbit/s	Industrial	FCBGA (RoHS)	399

B.2 Part Numbering Information

The part numbering system is explained as follows.



- ( ) – Indicates optional characters.
- Tsi – IDT system interconnect product identifier.
- NNNN – Product number (may be three or four digits).
- SS(S) – Maximum operating frequency or data transfer rate of the fastest interface. For operating frequency numbers, M and G represent MHz and GHz. For transfer rate numbers, M and G represent Mbps and Gbps.
- E – Operating environment in which the product is guaranteed. This code may be one of the following characters:
  - C - Commercial temperature range (0 to +70°C)
  - I - Industrial temperature range (-40 to +85°C)
  - E - Extended temperature range (-55 to +125°C)
- P – The Package type of the product:
  - B - Ceramic ball grid array (CBGA)
  - E, L, J, and K - Plastic ball grid array (PBGA)
  - G - Ceramic pin grid array (CPGA)
  - M - Small outline integrated circuit (SOIC)
  - Q - Plastic quad flatpack (QFP)

- G – IDT products fit into three RoHS-compliance categories:
  - Y - RoHS Compliant (6of6) – These products contain none of the six restricted substances above the limits set in the EU Directive 2002/95/EC.
  - Y - RoHS Compliant (Flip Chip) – These products contain only one of the six restricted substances: Lead (Pb). These flip-chip products are RoHS compliant through the Lead exemption for Flip Chip technology, Commission Decision 2005/747/EC, which allows Lead in solders to complete a viable electrical connection between semiconductor die and carrier within integrated circuit Flip Chip packages.
  - V - RoHS Compliant/Green - These products follow the above definitions for RoHS Compliance and meet JIG (Joint Industry Guide) Level B requirements for Brominated Flame Retardants (other than PBBs and PBDEs).
- Z# – Prototype version status (optional). If a product is released as a prototype then a “Z” is added to the end of the part number. Further revisions to the prototype prior to production release would add a sequential numeric digit. For example, the first prototype version of device would have a “Z,” a second version would have “Z1,” and so on. The prototype version code is dropped once the product reaches production status.

Index

Numerics

A

B

C

D

E

F

G

H

I

J

L

M

P

R

S

T

U

W

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01 Jan 2024)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.